Computers and Electronics in Agriculture 197 (2022) 106933

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Insect counting through deep learning-based density maps estimation

Arantza Bereciartua-Pérez [a,*], Laura Gómez [a], Artzai Picón [a], Ramón Navarra-Mestre [b], Christian Klukas [b], Till Eggers [b]

[a] Computer Vision, TECNALIA, Basque Research Technology Alliance (BRTA), 48160 Derio, Bizkaia, Spain
[b] BASF SE, Speyererstrasse 2, 67117 Limburgerhof, Germany

ARTICLE INFO

Keywords:
Convolutional neural network
Deep learning
Density map estimation
Insect counting
Image processing
Precision agriculture

ABSTRACT

Digitalization and automation of assessments in field trials are established practice for farming product development. The use of image-based methods has provided good results in different applications. Although these models can leverage some problems, they still perform poorly under real field conditions using mobile devices on complex applications.

Among these applications, insect counting and detection is necessary for integrated pest management strategies in order to apply specific treatments at early infection stages to reduce economic losses and minimize chemical usage. Currently the counting task for the assessment of the degree of infestation is done manually by the farmer.

Current state of the art object counting methods do not provide accurate counting in crowded images with overlapped or touching objects which is the case for insect counting images. This makes necessary to define novel approaches for insect counting.

In this work, we propose a novel solution based on deep learning density map estimation to tackle insects counting in wild conditions. To this end, a fully convolutional regression network has been designed to accurately estimate a probabilistic density map for the counting regression problem. The estimated density map is then used for counting whiteflies in eggplant leaves. The proposed method was compared with a baseline based on candidate object selection and classification approach. The results for alive adult whitefly counting by means of density map estimation provided $R^2 = 0.97$ for the counted insects in the main leaf of the image, that outperforms by far the baseline algorithm ($R^2 = 0.85$) based on image processing methods for feature extraction and candidate selection and deep learning-based classifier.

This solution was embedded to be used in mobile devices, and it has been gone for exhaustive validation tests, with diverse illumination conditions and background variability, over leaves taken at different heights, with different perspectives and even unfocused images, for the analyzed pest under real conditions.

1. Introduction

Digitalization and automatization are common practice in diverse applications fields, such as industry, health or entertainment. Agricultural sector is not far from this trend and the new coined term that gathers this philosophy is smart farming. Smart farming is strategic for tackling the big challenges of the sector regarding productivity, environmental impact, food security and sustainability. Agricultural sector is susceptible of automatizing big part of its processes.

Different researches have addressed diverse topics on automated actions for plants' analysis in the recent years, such as disease identification by computerized visual diagnostic methods (Fuentes et al., 2017) (Picon et al., 2019a)(Argüeso et al., 2020) or pest identification (Johannes et al., 2017)(Karar et al., 2021). A survey of Deep Learning applications in agriculture can be found in (Kamilaris and Prenafeta-Boldú, 2018) (Patrício and Rieder, 2018) (Zhang et al., 2020)(Tian et al., 2020).

Field researchers need to invest long hours in fields assessment. These are usually laborious and time-consuming tasks. Pests control is key question for the field researcher. The inaccurate evaluation of the pests may sometimes imply the survival of the crop. The faster and more accurate the detection is, the more effective the treatment will be. The dose of the insecticides needs to be regulated and it deeply depends on the degree of infestation. At this moment, visual inspection is used for

* Corresponding author.
E-mail address: aranzazu.bereciartua@tecnalia.com (A. Bereciartua-Pérez).

https://doi.org/10.1016/j.compag.2022.106933
Received 28 July 2021; Received in revised form 30 March 2022; Accepted 1 April 2022
Available online 12 April 2022
0168-1699/

the evaluation of the damage in the leaf and the insect counting. Ability to count and the underlying knowledge is required for the correct execution of this task. This is a time-consuming and inaccurate process because of the difficulty in tracking the bugs in the leaves, the small size they are, and often, the difficulty in distinguishing among different types of insects or the different stadia of the same species. Automatic insect counting could help the field researcher to decrease the time invested in leaves infestation evaluation. As it is stated in (Høye et al., 2021), deep learning and computer vision will transformed entomology.

In this work, a new method is proposed to count insects in leaves by means of density map estimation approach based on deep learning and image processing techniques. The method has been developed and validated for whitefly counting (single class) in eggplant crop. Whitefly is an insect which is affecting a large diversity of crops: from vegetables as eggplant, to cotton. The selection of eggplant has been done due to the impact to R&D. Eggplant is easy to grow, representative of greenhouse crops and vegetables. The population of white fly is growing good and it is easy to manage. Currently, the counting of insects in this culture in done manually which is highly imprecise and time consuming.

The paper is organized as follows. Section 1 introduces the problem. In Section 2 the related work is described. Section 3 explains the materials and methods used. The thorough description of the algorithm is detailed in Section 4. Section 5 gathers the results provided by the models. Discussion of the results is held in Section 6, before ending with the conclusions in Section 7.

2. Related work

Generally, counting and detecting objects in crowded images or videos is an extremely tedious, time-consuming and inaccurate task encountered in many real-world applications, performed manually most of times. The automated detection and counting of the objects were tackled initially with classical image processing techniques which presented lots of limitations and drawbacks, and a lack of generalization. The identification and extraction of the key features that allow to characterize properly an object and differentiate it from another was often a hard problem. Deep learning-based techniques have allowed to give a step forward in object detection and counting. In the recent years, many deep learning-based works that face up to this counting problem have emerged. These applications include cell or animal counting in biology (Xie et al., 2016) (Arteta et al., 2016)(Cohen et al., 2016) (Marsden et al., 2017a)(Hamwood et al., 2019)(Akintayo et al., 2018), crowd counting (Liu et al., 2018) (Marsden et al., 2017b), vehicle counting (Oñoro-Rubio and López-Sastre, 2016) (Tayara et al., 2017) (Song et al., 2019), or even whale counting in satellite images (Guirado et al., 2018), among others. Interesting reviews in the topic can be found in (Segui et al., 2015) (Sindagi and Patel, 2017) (Chattopadhyay et al., 2017) and in recent (Heinrich et al., 2019) (Ilyas et al., 2020).

Small object detection is a research line itself, and is being addressed in different works (Li et al., 2017)(Bai et al., 2018) (Kim et al., 2018)(Du et al., 2019). For instance, (Bai et al., 2018) state that although impressive results have been achieved on large/medium sized objects on large-scale detection benchmarks (e.g. the COCO dataset), the performance on small objects is far from satisfaction. The reason is that small objects lack sufficient detailed appearance information, which can distinguish them from the background or similar objects. In (Nguyen et al., 2020) they conclude that there is still a huge gap in accuracy between normal objects and small objects. Recent work is (Benjumea et al., 2021) where a new YOLO version is proposed to solve low performance of well-known network for small object detection such as YOLO, SSD and Faster R-CNN.

The core of modern approaches for counting, as density map is, was proposed in (Lempitsky and Zisserman, 2010). This approach consists on the fact that given labels with point annotations of each object, a density map of the image can be built. In this density map, each object takes up a density of 1, since it is represented by a normalized Gaussian, so the sum of the values of all the pixels of the density map image provides the total number of objects in the image. If a model manages to predict correctly this density map, the final object counting is the sum of all the image pixels values. This method considers overlapped objects. This is why density map estimation has been adopted as a good solution for counting due to many reasons. First, it simplifies the annotation process. Deep learning based semantic segmentation or instance segmentation techniques imply accurate annotation of the contour for the segmentation of the objects in the images of the dataset. However, for density map annotations, only a dot is needed on top of the object to be counted. Besides, the density map estimation outperforms the counting approaches by object detection since overlapped objects are also considered.

Some works can be found in the literature. (Arteta et al., 2016) propose a solution to count penguins in the wild. The images they deal with present high variability in illumination, perspectives, object size, occlusions and overlaps. They propose the incorporation of several annotation points produced by different annotators in noisy crowd images and background removal to prevent from errors. They also estimate the counting by means of density maps generation. (Kang et al., 2018) focus on a comparison of density map estimation methods and their performance on counting and two other methods based on localization tasks, such as detection and tracking. Recent work has been found for maize tassels counting in the wild (Lu et al., 2017), which helps to the correct estimation of the growth of the plant. Multiple techniques have been proposed in the last years to improve the accuracy of the counting, such as the fully convolutional redundant counting, proposed by (Cohen et al., 2016) or the counting in small datasets of large images (Aich and Stavness, 2018).

Insects counting use case has not been deeply explored till the moment, although being a relevant problem for the field researcher. The fast and accurate counting of the number of bugs in leaves makes it possible to prevent from future damage in the crop through early application of proper insecticide. This insects' counting task is currently performed manually. Easy-to-use mobile applications would be valuable for the field researcher. The processing capability of mobiles and current deep learning techniques makes this possible.

There are some experiences in the literature for insect location, classification and counting. For instance, (Barbedo, 2014) proposes a method for whitefly counting in soybean leaves. Therein classical image processing methods are applied based on morphological operations over different planes outcoming from transformation of the input image on diverse color spaces. (Liu et al., 2016) proposes a system for aphids' detection based on image processing and Support Vector Machine (SVM) techniques. Another method to locate and count Pyralidae insects in maize plant is proposed in (Gabriel de Oliveira et al., 2015). The proposed algorithm is based on color space transformation from RGB to HSV and the extraction of features of the insects that can describe their contour and shape. Hu moments for the histogram-based extracted shapes are obtained and the object is classified through a similarity measure (Huang and Leng, 2010). Hu Moments (or rather Hu moment invariants) are a set of numbers calculated using central moments that are invariant to image transformations. The first 6 moments have been proved to be invariant to translation, scale, and rotation, and reflection. While the 7th moment's sign changes for image reflection (Huang et, al, 2010). These approaches with classical image processing techniques imply the careful selection of features that further serve as input of a machine learning classifier, such as SVM, decision tree, Bayes, etc. These solutions based on classical image processing techniques usually lack the possibility of including additional classes in an easy way. Correct classification and detection of the object rely deeply on the proper extraction of the meaningful and discriminative features, which is often a complex task (Guo et al., 2016)(O'Mahony et al., 2020).

Deep learning-based approaches aim at overcoming these limitations. A system capable of detecting and counting different flying insects in the greenhouse is proposed in (Zhong et al., 2018). This work proposes a method based on You Only Look Once (YOLO) object detection

(Redmon et al., 2015). However, extended methods such as YOLO for object detection do not provide accurate counting in crowded images where overlaps exist, or the objects are closed to each other. Recently, (Roosjen et al., 2020) have proposed a deep learning based solution to detect the specie Drosophila commonly found in fruits. The drawback of this solution is that it is highly dependent on the background in the images. A red colored sheet is placed as background, so drosophila can be easily detected by classical image processing techniques and is further classified by an ad-hoc trained ResNet based model. Same approach is proposed to locate tomato whitefly and its predatory bugs on yellow sticky traps (Nieuwenhuizen et al., 2018). These traps are imaged in controlled light conditions with a digital camera. Similar approach based on stick-traps analysis by means of deep learning based techniques has recently been proposed in the greenhouse for multiple pest control (Rustia et al., 2021). Controlled environments can be also found in (Bjerge et al., 2020), (Lins et al., 2020), and (Wang et al., 2020) where a light trap and computer vision system are used to detect and classify live moths (Lepidoptera), aphids and the species stored in Pest24 dataset, respectively. These last methods, although providing reasonably good results in very controlled environments, are far from being applied over leaves into the wild for variable and non-uniform lighting conditions with different backgrounds and acquisition distances.

Few experiences can be found in the literature for insects' detection in the wild. In (W. Li et al., 2019), the authors propose a deep learning-based pipeline for localization and counting of agricultural pests in images by self-learning saliency feature maps with good results in controlled environment over wheat leaves. Detection of aphids in the leaves by deep learning techniques is tackled in (R. Li et al., 2019). Main limitation the authors refer is that the detection is not very accurate whenever the aphids are in dense distributions with severe overlaps. Object detection-based approaches perform poorly in these situations. On the contrary, density map estimation outperforms object detection methods mainly in situations of object overlapping and small objects (Heinrich et al., 2019; Ilyas et al., 2020).

In our work we face up to these limitations and propose a new density map based approach to count and detect insects over different crop leaves in the wild and compare this technique with the existing object detection approaches.

3. Materials & methods

In this Section a detailed description of the available dataset is provided. Additional information for the complete understanding of the problem is also given, such as the difficulties in processing images in wild conditions and the specific characteristics of the object to locate that makes its identification and counting challenging.

a. Dataset

An acquisition campaign took place in greenhouse in Spain. Images were acquired with different illumination conditions and in different seasons with 3 different devices: Olympus 24 M pixels camera, Nikon 20 Mpixels camera, and Samsung Galaxy A8 smartphone. This variability in acquisition devices assures different acquisition conditions, regarding focusing, resolutions, illumination, etc. which is necessary to develop a robust algorithm capable of working in wild conditions. The summary of the dataset is gathered in next Table 1.

Table 1
Overview of the images available in the dataset for BEMITA (whitefly) counting.

	Milestone	Target	Device	N. required pictures &description	Total done
Identify & counting the adults	Counting the total number of insects/ adults	Full leaf	Digital camera x2 (20 and 24 MP)	60 adults + 76 (extra dead adults)	136
			Smart Phone	30 adults + 38 (extra dead adults)	68
	Leaf automated segmentation		Digital camera x2 (20 and 24 MP)	500–1000 (damage not related with adults)	821 + 161 (clean leaf) = 982

Eggplant (Solanum melongena, SOLME) leaf has been used for validation purposes. The insect to be located is *Bemita tabacci* (commonly named as whitefly). The following image in Fig. 1 shows adult whiteflies on eggplant leaves in a photo taken by a mobile phone.

Both alive and dead whiteflies can be present in the leaf at a time, presenting subtle visual differences. Dead whiteflies often appear knocked down, and their legs can be appreciated. Sometimes the wings are unfolded in an inanimate way. Alive whiteflies appear well rounded, wings usually folded, standing firmly over their legs.

One additional difficulty is that only alive whitefly has to be taken into account on insect counting for the farmer and is to be counted. Example of alive and dead whitefly can be appreciated in Fig. 2.

Classic image processing techniques can difficultly tackle with these challenging images (Johannes et al., 2017). The solution based on a feature extraction and classification approach would probably get confused mainly due to the difficulty for extracting unambiguous discriminative characteristics for these two visually similar classes. However, we as human being can distinguish them, so deep learning-based techniques are expected to do so with proper design and training of the network models.

Specific concern is related to the size of the insects within the image. It is easy to locate a big element in a photo, but the tiny insects that infest the crop leaves are smaller than 1 mm size. This means that it can be represented by 20–30 pixels size or even smaller in a 4000x6000 image which precludes the use of classical object detection algorithms.

Moreover, images acquired in wild conditions present high variability in illumination and background. Smartphones do not usually have very powerful cameras, and images may be unfocused. Acquisition distance is also unpredictable. Therefore, the algorithm must recognize unfocused objects of different sizes in unexpected background and non-uniform illumination conditions, sometimes partially occluded and overlapped. Deep learning-based approaches can deal with these problems.

b. Annotation process

All the images in the dataset were accurately annotated by expert technicians. The annotation process was performed with the LabelMe tool (Russel et al., 2008) that was adapted to our problem. For the density map proposed approach it is not necessary to segment the bugs. Each alive insect on the image was tagged with its pixel location on the image. The annotation process consists of marking with a point the presence of an insect in the leaf. These dots will be used later for the generation of the models and testing purposes. Fig. 3 shows an example of an input image and the output of the annotation process.

4. Proposed solution. Insect counting algorithm

As it has been exposed in Section 2, there are few algorithms proposed in the literature capable of detecting insects in crops. Regarding our insect counting problem, they could be located on an object detection basis. In crowd counting problem, recent work has demonstrated that object detection and location approaches are not enough (Redmon et al., 2015)(R. Li et al., 2019) (Ilyas et al., 2020). An additional problem is found in most images: more than one infested leaf can appear in the

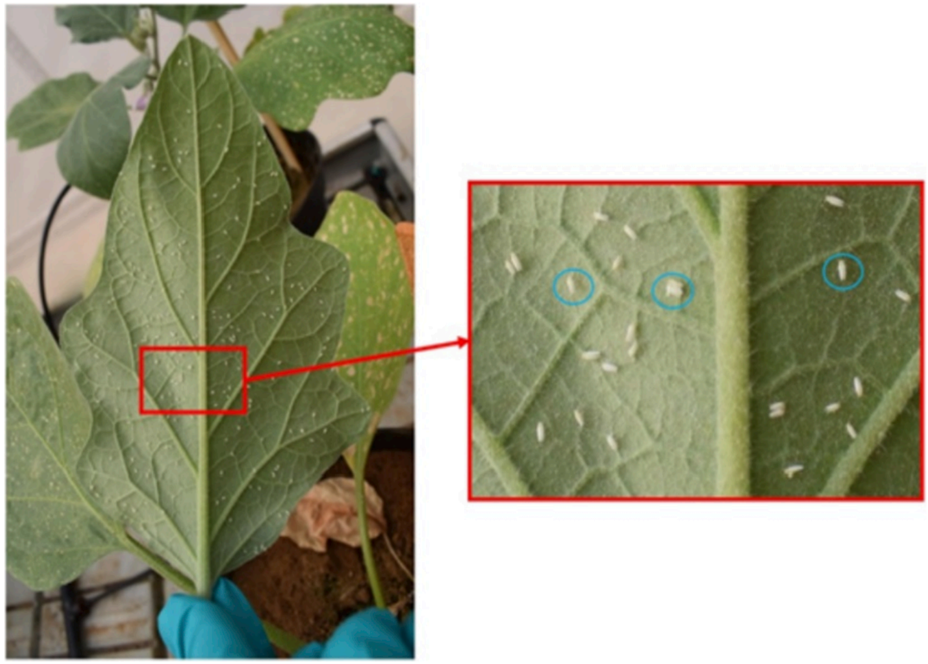

Fig. 1. Eggplant leaf infested with adult alive whitefly.

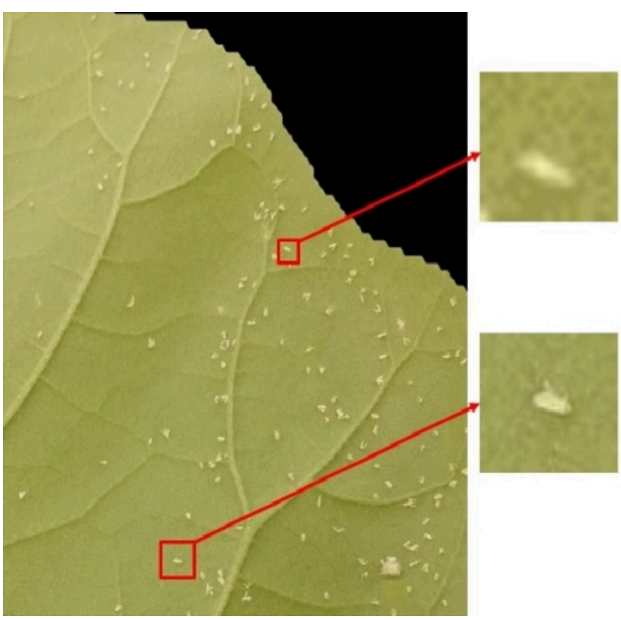

Fig. 2. Example of dead (bottom) and alive (up) whiteflies together in the leaf.

image. There is a main leaf but there may be other leaves overlapped. In the case the algorithm is applied over the whole image as it is, the final number of insects can include bugs placed in more than one leaf which is not a standardized metric for experimental trials.

Our proposal is to develop a model that combines leaf segmentation method to segment the foreground leaf on the image discarding other leaves on the image to prevent from the appearance of false positives in adjacent leaves. In a second stage, insect counting will be performed on the image.

A two-step algorithm is proposed:

- o Leaf segmentation: the main leaf is segmented, so only the insects lying on that leaf are considered for counting. This eliminates the adjacent leaves that may introduce false positives.
- o Insects counting. Two approaches are proposed. First approach is on a basis of candidate selection and classification algorithm. Second approach is strictly to apply deep learning-based density map estimation.

Fig. 4 shows this pipeline.

Each stage is detailed in the following sub-Sections:

a. Leaf segmentation

Leaf segmentation is a complex problem that is out of the scope of this work. Nevertheless, it is necessary to tackle the problem minimally to guarantee that the number of counted insects lie in the main leaf, and no insects are included from adjacent leaves.

Preliminary tests for the leaf segmentation process was done based in color space transformation and processing over channel *a* of Lab color space. In channel *a* green color can be identified in an easier way. This is a very good solution in the case the leaf is isolated over a uniform background. However, in the case there are other leaves in the background, inhomogeneous illumination conditions, or damaged leaves incorrect segmentation was obtained, since adjacent regions of other leaves are attached to the main leaf, as it can be seen in Fig. 5.

These overlaps cause great difficulty in distinguishing the edges of two green adjacent leaves, a second approach for leaf segmentation problem is tackled based on deep learning techniques on a basis of semantic segmentation. To do so, a dataset of 1947 images was used for training. They were manually annotated through previously referred LabelMe tool. A Fully Convolutional DenseNet model was trained with a semantic segmentation purpose (Huang et al., 2017) in a similar way as in (Picon et al., 2019a) (Picon et al., 2019b). The input to the network is an RGB image 224x224 size. The output has 2 channels: leaf / no leaf. It is a mutually exclusive classes problem, so activation of last layer is “*softmax*”, loss function is “*binary_crossentropy*”. No fine tuning is applied. It has been trained from scratch.

Average Dice coefficient of 0.95 is achieved in leaf segmentation process. Performance of the trained model is very good for healthy leaves (Dice coefficient of 0.98) whereas it decreases slightly for damaged leaves (Dice of 0.94). Some examples are shown in Fig. 6.

As it can be appreciated from the examples in Fig. 6, the model can

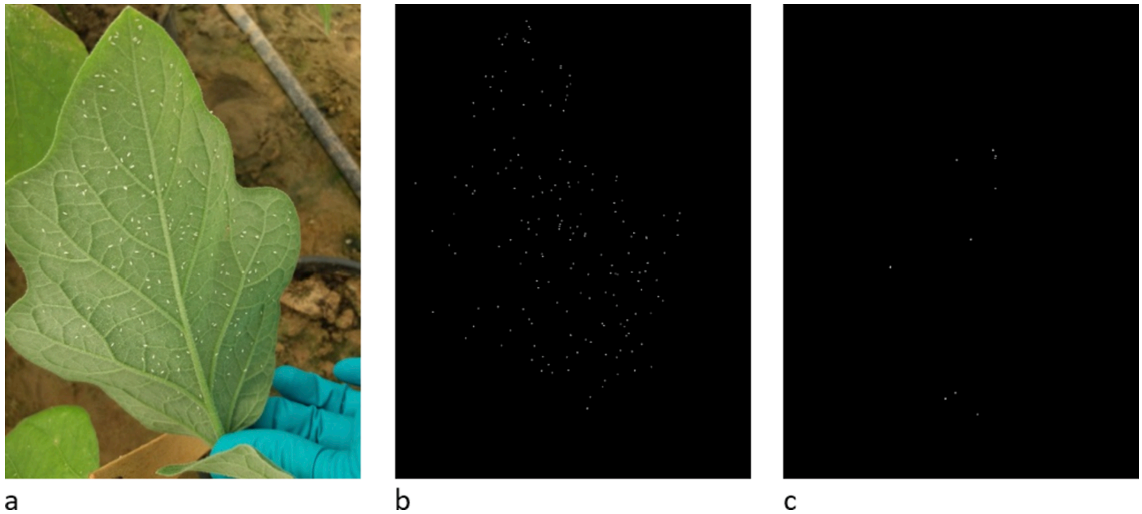

Fig. 3. Example of outputs in the annotation process; a) original image, b) annotated dots of alive whiteflies, c) annotated dots of dead whiteflies.

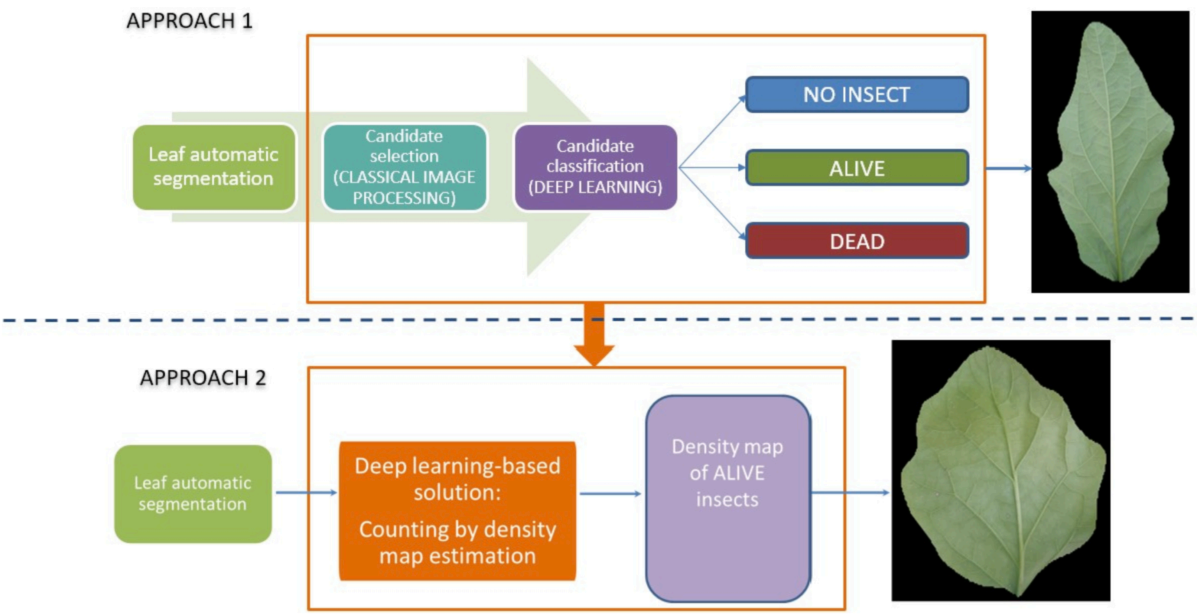

Fig. 4. Pipeline of approaches 1 and 2 for insects' counting.

deal with non-uniform illuminations in the image and in the leaf itself, the overlap of several leaves, and it is capable of segmenting properly the front leaf on which insects are to be counted.

Once the leaf is properly segmented, second stage is to count the insects. As it has been shown previously in Fig. 4, this problem has been tackled with two different approaches that will be compared. First approach is on a basis of candidate selection and classification; and second approach is on a basis on density map estimation with deep learning techniques.

b. Approach 1: Candidate selection and classification

First approach aims at applying deep learning-based techniques in a classification basis. The idea is to locate initially all the possible candidates through a color-based blob extraction procedure. Later on, each candidate and established neighborhood will pass through a deep learning-based classifier that will decide whether it is alive whitefly, dead whitefly or nothing.

Whiteflies are always white colored, so it could be thought that they can be detected by means of the RGB image analysis. This is not so easy in images taken in the wild, since there is high variability in the scene and illumination conditions. Moreover, shines, sparkles and damages, also white colored, may appear over the leaf surface, and they can be easily confused with small white elements as the whiteflies are. Leaf color and texture and damage stage is variable. The transformation of the image to another color space could minimize the impact of this problem.

Additionally, there might appear dead whiteflies together with alive whiteflies. This is not very frequent since dead insects fall from the leaf, but it might happen. So, it is needed to find a way of rejecting the dead individuals and count only the alive whiteflies, which gives information about the leaf infestation level. The differences between alive and dead whiteflies are subtle and the complexity to characterize them properly by means of a set of features is high. Fig. 7 shows several alive and dead whiteflies together with other elements in the leaves that present visual similarity.

Candidate selection is based in Blue plane (RGB space) and v plane (Luv space), inspired by (Comaniciu and Meer, 1998). In v plane insects

Fig. 5. Example of good results (left) and incorrect results (right) for leaf segmentation based on *Lab* color transformation. In the right image some leakages due to the adjacent leaves can be appreciated in the non-perfect segmentation.

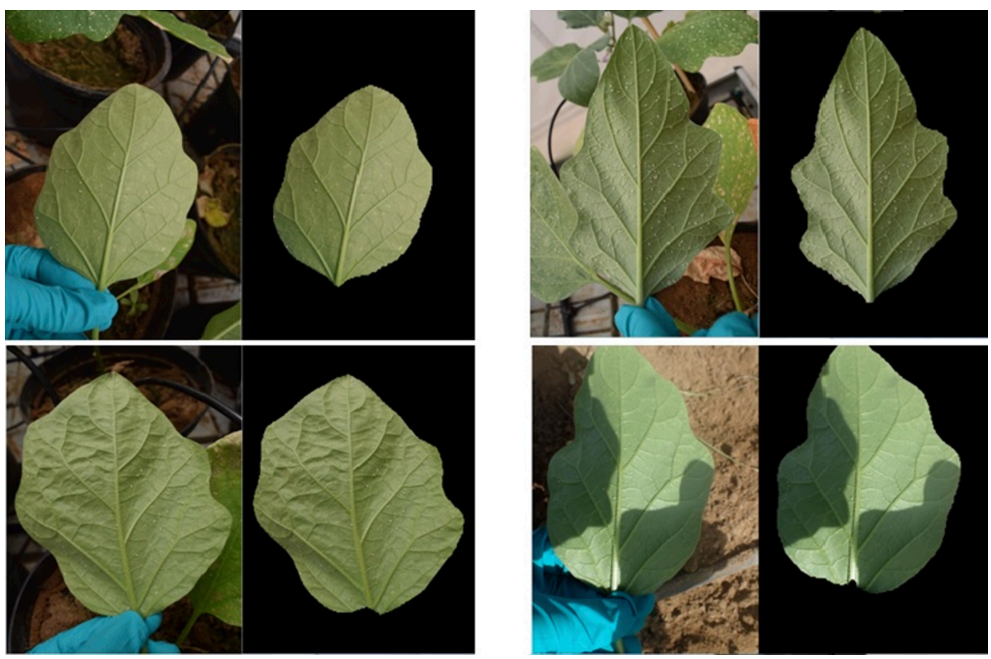

Fig. 6. Examples of leaf segmentation performed by Fully Convolutional DenseNet network model.

appear with a higher contrast with respect to the leaf surface. The illumination irregularities are minimized in this plane. The whiteflies in v plane can be recognized over a more uniform background. This situation can be seen in Fig. 8.

The higher uniformity of the light conditions in the leaf surface makes it easier the extraction of the potential candidates through a global thresholding process. The connected regions above the threshold are again analyzed by means of morphological operations of classical image processing techniques. Filtering of blobs is made by several criteria of size. This way very small (<10 pixels size) and very large (>200 pixels size) regions are removed, since they don't correspond to insects. The remaining objects in the image are candidate to be insects. The centroid of each region is obtained and is the center of a 96x96 pixels sized image that constitutes the input to the classification network. The pipeline is represented in the following Fig. 9.

The available dataset is of 731 images, and they have been split up this way: 554 images are used for training, 54 images are used for validation and the remaining 123 images are for testing.

As we have explained, once the candidates are extracted, they enter a deep learning-based classifier. The 534 annotated images available for training containing thousands of whiteflies allow the creation of a dataset generated with examples of the three classes: class 1 – alive, class 2- dead, class 3 – no insect. The dataset contains 14,443 images of alive whiteflies, 4721 of dead whiteflies and 18,397 images without insects. In every batch in the training process the three classes are shown. Data augmentation is used to increase the number of examples of dead whiteflies, since there is a smaller number of them. The network architecture chosen for classification is well-known VGG16 (Simonyan and Zisserman, 2015) with a three-classes output. Categorical cross-entropy is used as loss function since the three classes are exclusive. Input image size is 96×96 pixels. Learning rate is 0.001.

Drawback in this approach is that the extraction of the candidates is weak, since it relies on traditional feature extraction procedures based on color analysis and morphological operations. In images with changeable and non-uniform illumination conditions, and besides with the aim of finding tiny elements, traditional feature extraction seems inaccurate and insufficient and add false positives (shines, sparkles, damages) as candidates. This is the reason why a classifier is added next

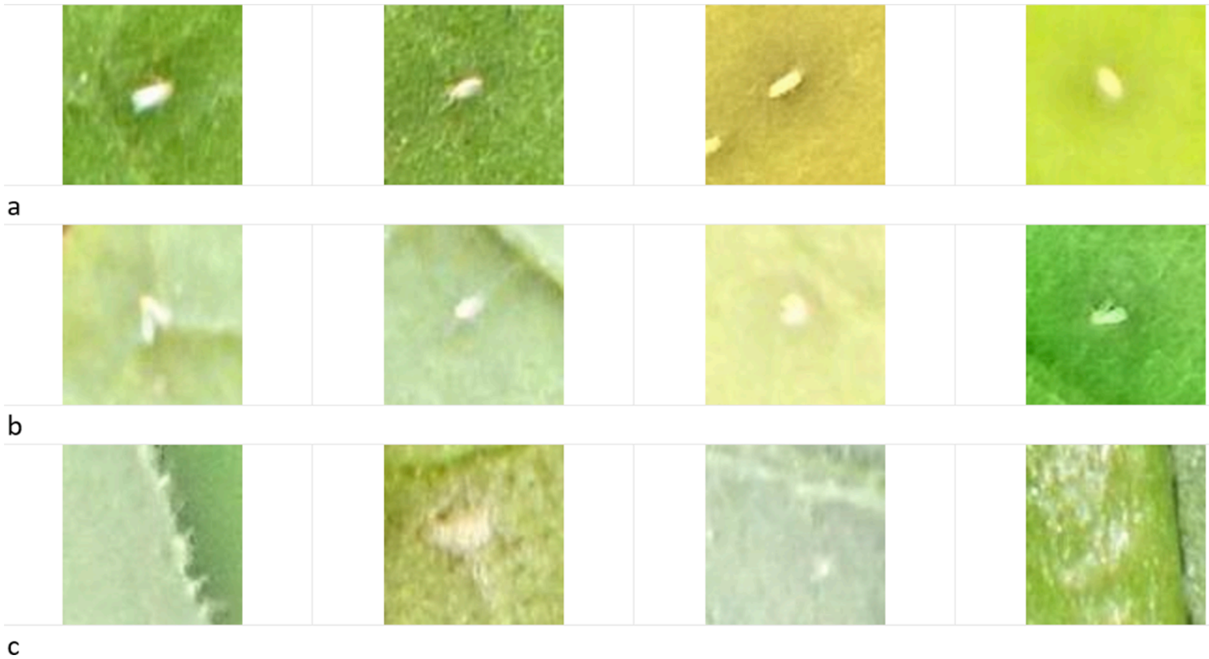

Fig. 7. Example of: a) alive whiteflies, b) dead whiteflies, c) other white elements detected in the leaf.

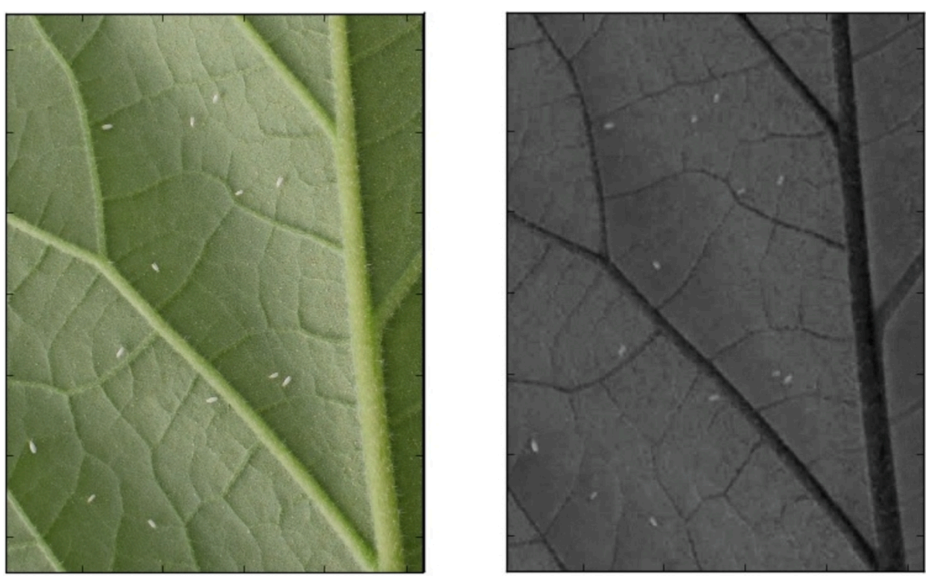

Fig. 8. Detailed example of how whiteflies appear in v plane (from RGB to Luv color space). a) Original image in RGB space; b) original image represented in v plane. Illumination variability decreases.

to remove the majority of false positives as possible, this is, the elements not being alive whitefly. The advantage of this solution is that its implementation is easy and fast.

c. Approach 2: Density maps estimation for bug counting

Density map estimation approach for counting purposes was first described in (Lempitsky and Zisserman, 2010). After annotation of every object in the image with dots (no contour segmentation is needed), a density map is constructed where each object is described by a normalized Gaussian. The sum of all the pixel values of that Gaussian is equal to 1. A Fully Convolutional Regression Network aims at learning this prediction values whenever an object is present. This method naturally accounts for overlapping objects.

The density map estimation is a pixel-by-pixel regression problem. Every bug (object to be counted) is represented by a normalized Gaussian. A normalized Gaussian implies that the sum of all the pixels values is 1. Therefore, one object is equivalent to one complete Gaussian. If there are two Gaussians together the sum will be up to 2. Although the elements are partially overlapped, if the model has properly converged, the sum will be up to 2, as well. The big advantage of density map estimation over an object detection approach is that it can count 2 instead of 1. When there are overlaps it is very difficult for an object detection approach to identify the two elements, and generally only one is detected. So, density map estimation seems to be suitable in this counting problem, because insects, and very commonly in whiteflies, tend to be in pairs or even three elements appear together, as it can be shown in Fig. 10.

Fig. 9. Pipeline for candidates' extraction from v plane (from RGB to Luv color space) and morphological processing. a) Original image in RGB space; b) original image represented in v plane. Illumination variability decreases; c) thresholding; d) size based filtered image where candidates are extracted.

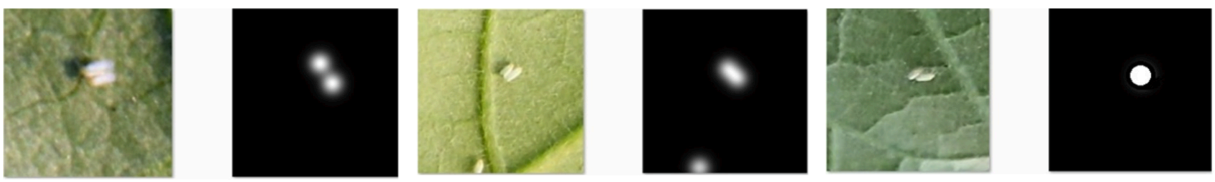

Fig. 10. Example of advantage of density map estimation over object detection approach for two insects overlapped. a) two whiteflies closed each other imply two gaussians; b) two very closed and small whiteflies imply two overlapped gaussian response; c) one whitefly as result in an object detection basis.

Density map estimation techniques are feasible for small σ. Values of high σ imply high number of pixels that represent every object, and it becomes more difficult for the model to be trained and to predict proper values for all the pixels. For objects that need to be represented by gaussians of $\sigma > 15$, it is recommended to work on an object detection or instance segmentation basis (depending on the number of object classes) but not with density map estimation approach.

We have adapted the method in several aspects to make it suitable for our problem. Training process, model and inference process has been designed ad-hoc to fulfil our special requirements.

The proposed solution to our specific problem is as follows. Since the bugs are very small, about 20–30 pixels size in a 4000x6000 image, resize of the image is not possible because bugs would be lost. Full image cannot be processed by a convolutional neural network, it is too big to be supported by GPUs memory. Full image size as input would imply that the network model must have many layers to appreciate the small bugs into their receptive field. Therefore, the image is processed tile by tile. The size of the tile is established at 256x256 pixels. Each patch overlaps with the neighboring patches in all directions. Overlapping margin is 40 pixels. The inference process is the following. The original full-sized image is split up in the necessary number of tiles according to the input image size, that can be up to 396 tiles in a 4000x6000 image. Prediction is made for all of them. Final predicted image is reconstructed by means of the central activations of the predicted tiles. Margins of 40 pixels are not considered in the final image reconstruction. This methodology provides more coherent response.

Images in the dataset range from 12 MB to 24 MB pixels size. Considering focal distance in mobiles are similar so will be the field of view, therefore, in the inference process, the images with higher resolution should be automatically rescaled to fit the training range. In order to maximize image quality, images will be rescaled to fit the maximum training size range, this is, 4000x 6000 pixels.

The network model could be any fully convolutional model, since density map is a kind of semantic segmentation in combination with pixel-by-pixel regression model. Instead of predicting the pixels that belong to a specific element and assign a single value label, it predicts a normalized Gaussian every time an element is detected. Therefore, Fully Convolutional Network is needed but understood as a regression problem. Last layer has a '*linear*' activation. Output is a one-channel image, for density map representation. Loss function has been fixed to '*mean square error - mse*'.

Several networks topologies were tested, among others UNet (Ronneberger et al., 2015), DenseNet (Huang et al., 2017) and Fully Convolutional Regression Network (FCRN) proposed by (Xie et al., 2016) for cell counting. Modifications have been made over this last model. Two more layers were added in the encoding part and consequently, two more layers were also added in the decoding part by means of upsampling and deconvolution procedures. Slight modifications imply changes in sigma to represent properly the size of the element to be counted. In the network model, we propose to use global sum pooling (GSP) instead of global average pooling (GAP) or fully connected (FC) layers at the backend of a convolutional neural network. GSP allows convolutional networks to learn the counting task as a simple linear mapping problem generalized over the input shape and the number of objects present, and the answer is much better than when using GAP. This idea was inspired in (Aich and Stavness, 2018).

Best network model has been trained with the following parameters. Input image is 256x256 size, $lr = 0.01$, $\sigma = 9$, $l_2 = 10^{-5}$ and no dropout is used. The training was launched with 2000 epochs and the model weights providing smaller validation loss were chosen. Stochastic Gradient Descent was selected as optimizer. For training the network, a GeForce GTX TITAN X with 12 GB of memory was used. Considering the size of the input images and the network the batch size was set to 12 images. Sigma values for the generation of the density map is directly related with the size of the insect. Final value of sigma is 9 for whitefly.

The available dataset is of 731 images, and they have been split up this way: 554 images are used for training, 54 images are used for validation and the remaining 123 images are for testing. In the training stage, only tiles containing insects have been provided to the network. In the case there are several insects in a leaf, random selection of the insect coordinates is made. The tile is cropped from the image taking these insect coordinates as center of the image. To discard any relation of the insect with its position in the center of the image, translation affine transformation is always applied together with data augmentation operations. This way, insects can appear in any position and orientation in the image. Therefore, it could be considered that the training images are not only 554, but the total number of annotated whiteflies. In all training batches, every image tile has been extracted from a different image, this is, it comes from a different leaf. This guarantees additional variability in every training batch.

Data augmentation techniques were applied during the training process to increase the variability of the images seen by the network. Affine transformations were applied except zooming. Changes in scales are not adequate since the pixel values are modified in the mathematical operations and this causes that it stops being a normalized Gaussian, this is, sum up of pixels in the Gaussian is no longer exactly equal to 1.0. This would be a problem in terms of correct prediction and learning capabilities of the networks.

One of the main problems encountered during the development is the difficulty in making the network learn during the training process with such small activation values. Maximum peak value of a Gaussian distribution with sigma = 9 is 0.0067. This value is so small that the model doesn't know what has to be learnt between 0.0 and 0.0067 in float images of possible maximum values of 65536. First attempts failed. To solve this problem, a scale factor of 50,000 was applied to the whole image. This enables the visibility of the gaussians in the image during the training process and the model manages to learn this way. Once the prediction is obtained, the whole image is divided by the same scale factor, so real image value is recovered. Number of bugs is the total sum of pixel values of the whole image.

By design, the estimated density map corresponds to an image where the sum of all the pixels provides the total number of insects present in the image. However, for the validation of the system the field researchers would like to know where the insects are, so from the density map, the independent gaussians have been located and shown with a red point in the estimated centroid and rounded with a circle. Note that in overlapped whiteflies only one blob is considered and thus, only one point and circle is marked for identification, even if it may add two or more elements in the total counting. Fig. 11 shows the results obtained for alive whitefly detection.

In the final image, every located gaussian should sum up 1 in the case it represents one insect and should sum up 2 or 3 in the case they represent 2 or 3 overlapped insects, respectively. Nevertheless, due to small or unfocused insect, there might happen a low confidence in prediction because sometimes an alive adult can be confused with a dead whitefly or a damage region in the leaf. This can make that the sum of the pixels in the Gaussian can be far from the expected value. It has been stated that gaussians with sum of the pixels higher than 0.5 are considered as 1 additional insect, whereas activation values for the gaussian lower than 0.5 are discarded (not enough confidence in prediction). Same approach for gaussians representing two insects, being values smaller than 1.5 considered as single element, and higher than 1.5 considered two overlapping elements. This post-processing step, a kind of "filtering", has contributed to the improvement of the results and the rejection of some dead whiteflies.

5. Results

The described algorithms were developed on Python 3.7 programming language and deployed as a service on a Linux based processing server. These algorithms are based on Deep learning paradigm using Tensorflow 2.3 framework as backend.

The application is provided as a docker image. The deployed service was prepared with REST Application Programming Interface (API) that manages the connections from smartphone applications. Processing time of the algorithm is 1–2 s depending on the resolution of the input images, being the higher time obtained for images 4000 × 6000 size. The images cannot be resized to increase response time, because the reduction of resolution could imply the disappearance of the bug. As it has been explained above, the image is not processed at full size, but it is cropped in patches of 256x256 pixels. Every tile is predicted, and final full image is built again with output predicted patches. This is the task that takes longer. Anyway, response time is acceptable for mobile applications even if it could be improved in the future. Fig. 12 shows the interface of the mobile app develop to be used by field expert technicians.

Exhaustive validation tests were performed on Utrera experimental field where expert technicians could validate the results of the proposed approaches. Metrics were established to make comparison possible, to evaluate the performance of the different models and decide which of them performs better.

As it has been said previously, the available dataset is of 731 images,

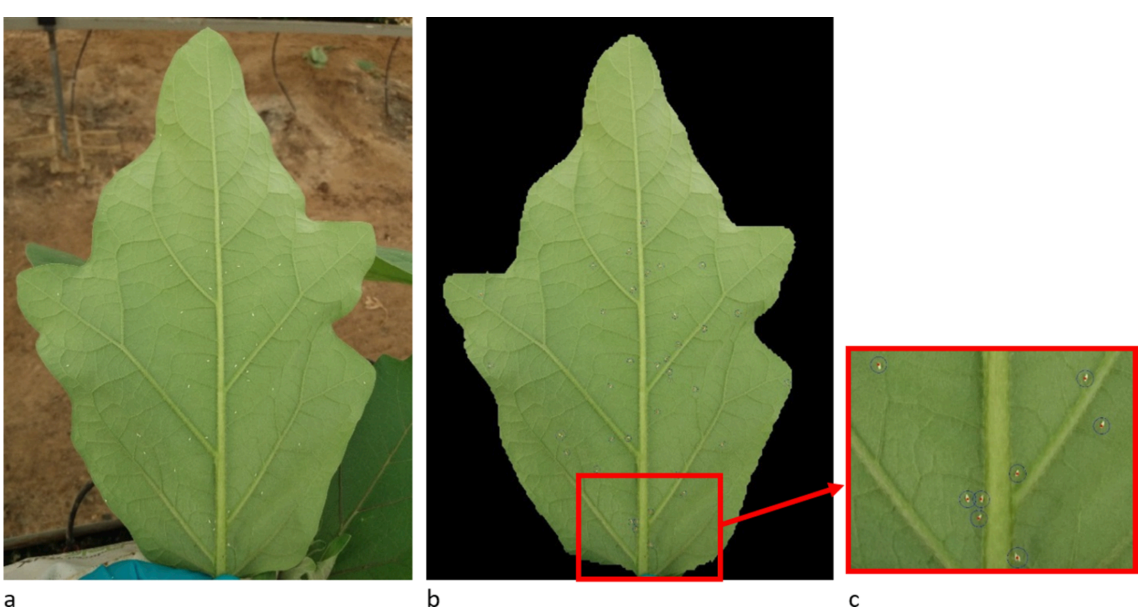

Fig. 11. Example of eggplant leaf processed with the algorithm pipeline. a) original image, b) final segmented leaf with the alive whiteflies counted and marked; c) detail on location of whiteflies (by location of predicted gaussians).

Fig. 12. Interface of the developed app for the testing of the insect counting solution.

and they have been split up this way: 554 images are used for training, 54 images are used for validation and the remaining 123 images are for testing. It was verified that every image contains a different leaf. The three subsets contain images acquired with the three different acquisition devices mentioned in Section 3.a. Distribution of leaves in training, validation and test subset is randomly made.

Regarding the execution of the experiments, both approaches for the alive whitefly counting algorithm were tested over same 123 testing images. Only this way the metrics and regression graphics obtained for both approaches can be compared. In all cases, the provided metrics are the average values obtained for the testing dataset.

RMSE and MAE have been established as metrics, together with R^2. This R^2 value is a statistical measure of how close the data are to the fitted regression line. This value shows the correlation between the predicted values and the real values (ground truth). It is also known as the coefficient of determination. A desirable value of R^2 is 1.0. It means that the predicted values fit a perfect line with slope of value 1.0 in relation to the ground truth values. An R^2 of value 0 means that the predicted values are not better than taking the mean value of the × axis values. If the R^2 value is negative, it means that the model is performing worse than the mean value. For the interpretation of the regression graphics, it is necessary to point out that one point can represent many images, and one isolated point might only represent one single situation.

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) present similarities with the human understanding, since they represent the direct relation between the predicted and the real value. The metrics are calculated this way:

$$MAE = \frac{\sum_{i=1}^{N}\left(y_i^{EST} - y_i^{GT}\right)}{N}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y_i^{EST} - y_i^{GT}\right)^2}{N}}$$

The regression graphics for the first approach based on candidate selection and classification is shown in Fig. 13. This graphics gives visual feedback about how the predicted number of whiteflies fits the ones counted by the experts. The smaller figure on the right focuses on the behavior of the algorithm in low infested leaves containing up to 50 whiteflies, where it is of higher importance not to retrieve false positives.

The regression graphic for the density map estimation approach are shown in Fig. 14.

The metrics for both approaches are gathered in Table 2.

6. Discussion

The results shown in previous Section for both approaches to count insects in leaves described in this paper reveal that density map estimation approach outperforms candidate selection and classification approach. All the metrics values are improved. R^2 value increases from 0.85 to 0.97 and MAE decreases from 8.22 to 3.36. The regression graphics show that predicted number of insects fits better with the number indicated by the experts in the images used in the testing dataset.

First approach is based on candidate identification and deep learning-based classification. The selection of candidates is performed by means of color space transformation and morphological operations, this is, classical image processing techniques. The difficulty of identifying correctly all the whiteflies in the leaves is high, since it is a tiny and rounded white colored object, that can be easily confused with other white colored elements in the leaf, such as shines, sparkles or damaged points. Moreover, some dead whiteflies can occasionally appear in the leaf. These dead whiteflies are visually very similar to alive whiteflies, so the proper characterization for differentiation from alive whiteflies is a too exigent objective to be accomplished by feature extraction procedure in reduced area of pixels, being this around 30 × 30 pixel, at most, in a high resolution image. This approach assumes the drawbacks of classical image processing techniques and aims at filtering the candidates by a deep learning-based classifier. The model is not capable of filtering properly all the candidates. The candidate number can be quite big (mainly in bright images), that even if the number is highly reduced, there are still some false positives. In other occasions, irregular illumination conditions do not allow a proper detection of candidates in the first proposed approach, and those insects are lost.

Fig. 15 shows an example where density map estimation method is capable of counting (and thus locating) better the insects in the image in the case illumination is extremely non-uniform.

Fig. 16 shows an example where density map estimation approach is capable of ignoring properly false positives due to shines and damages in the leaf.

In the crowded images the natural behavior of the whiteflies is to lay one close to each other. Whenever the infestation in the leaf is high, they tend to be in pairs or even grouped in three. These groups are identified as candidates by the initial stage of the algorithm. However, if they are not discarded by the classifier, the alive whitefly counter only is increased by one element, since it is not detected at this point whether there are one or more bugs together. In fact, this is quite difficult for a classical image processing algorithm, they appear totally overlapped.

The second approach proposed is based on density map estimation as described in Section 4. If training is done through representation of normalized gaussians, it is expected same normalized output. Therefore, the sum of the values of all the pixels in the image provides the estimated number of insects. In this approach feature identification and extraction procedure is removed from the algorithm. Convolutional Neural Network are applied, and they are in charge of extraction of meaningful features through the convolutions in the different layers. These techniques allow to obtain more subtle differences that the human made algorithm might provide through color transformation and morphological operations. The prediction is much better for this approach.

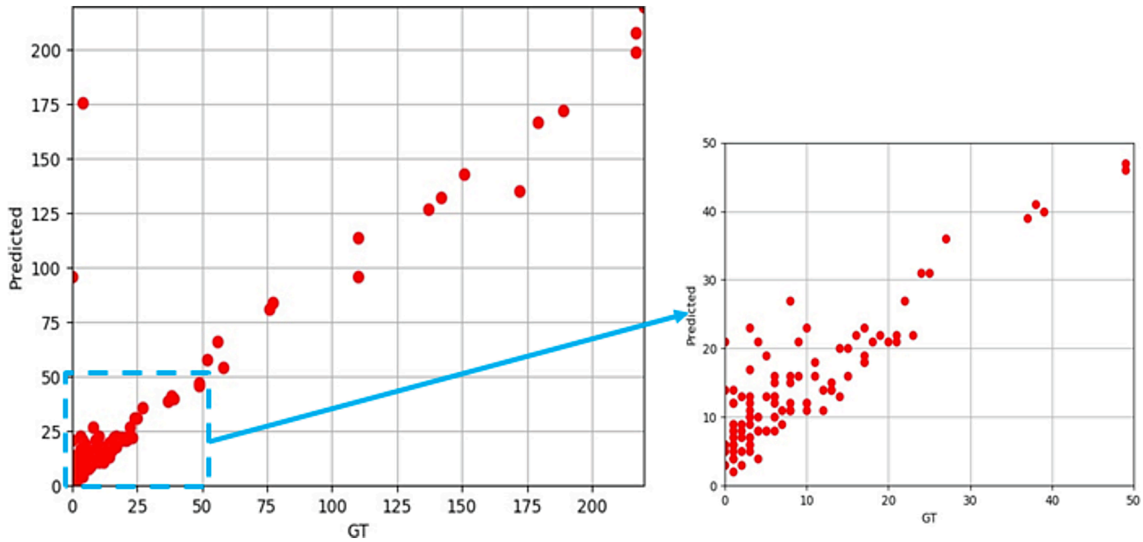

Fig. 13. Regression graphic over 123 images for Bemita alive whitefly counting candidate detection and classification approach. The × axis represents the real number of insects or Ground Truth (GT) and the y axis represents the number predicted by the algorithm. The graphic on the right zooms the curve for leaves with number of insects up to 50.

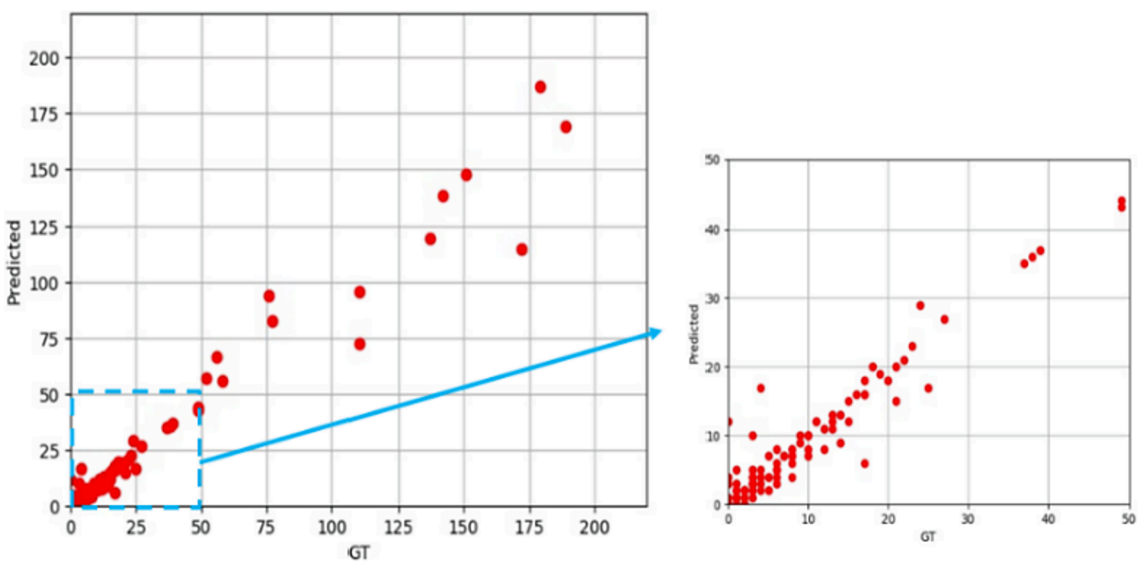

Fig. 14. Regression graphic over 123 images for Bemita alive whitefly counting with density map estimation approach. The × axis represents the real number of insects or Ground Truth (GT) and the y axis represents the number predicted by the algorithm. The graphic on the right zooms the curve for leaves with number of insects up to 50.

Table 2
Metrics for baseline approach (whitefly counting by candidate selection and classification) and whitefly counting by density map estimation approach. Intermediate stage for raw density map output has been included.

	RMSE	MAE	R2
Candidate selection and classification	19.51	8.22	0.85
Density map estimation (raw)	11.77	5.93	0.96
Density map estimation (post-processing)	7.84	3.36	0.97

Besides, density map estimation outperforms candidate selection and classification approach whenever two or more whitefly are overlapping each other. It is possible to predict two overlapping gaussians whose pixel values sum up to 2 in the case two whitefly lay together. The estimated number of insects is more precise to achieve up to a R^2 of 0.97.

So far, the advantages and good results of density map approach in relation to candidate selection and classification approach have been shown in Fig. 15 and Fig. 16. However, some limitations have been also identified in the proposed approach. Fig. 17 shows a perfect location of whiteflies (left) and an image with incorrect counting due to small white damages in the leaf surface (right).

Fig. 18 shows two other possible situations that present errors, such as the incorrect leaf segmentation or the unfocused image.

The main detected limitations are:

- The whitefly final counting value deeply depends on the good output of the leaf segmentation stage. Incorrectly segmented leaf, either showing under-segmentation or over-segmentation, provides inaccurate region of interest to the object detection and counting stage. In our work, object counting is done by density map estimation approach, but this limitation would apply to any other method based

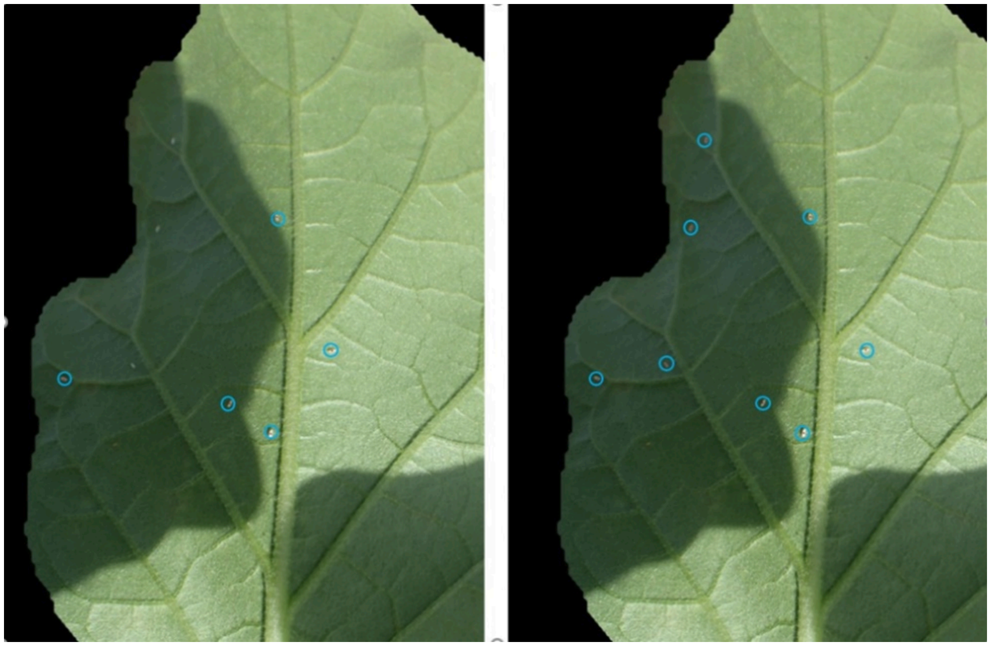

Fig. 15. Leaf affected by non-uniform illumination. Left) Insects detected and counted by candidate selection and classification approach; right) insects detected and counted by density map estimation approach.

Fig. 16. Leaf affected by shines and damages. Left) Insects detected and counted by candidate selection and classification approach; right) insects detected and counted by density map estimation approach.

on object detection or instance segmentation whenever a leaf segmentation is obtained first.

- Density map estimation outperforms object detection method mainly in situations where adjacent or overlapping objects appear. However, the object of interest in this work is a tiny whitefly. This makes sometimes especially difficult to distinguish properly whether two or more whiteflies are together or one on top of the other or it is a whitefly with the wings unfolded. This is also difficult even for an expert looking at the image (not difficult looking directly over the leaf, of course, the 3D view perceived by the eye can appreciate this).
- There are some false positives due to objects that are difficult to be differentiated and that present similar features than the whitefly. This is the case of shines, sparkles, white spotted damages in the leaf that look very similar to whiteflies and that make the model get confused (Fig. 16).

Finally, and as it can be appreciated in Fig. 18 (right), the image quality is key issue for a correct counting and detection of whiteflies. Unfocused images or blurred regions avoid the correct location of the insects. This fact is especially remarkable in our work since the object to be identified is really small.

As last, we wanted to compare our results with a YOLO-based object detection solution. We have adapted the annotation and prepared some code to apply a YOLO model to our problem. Anchors have been adapted to the small object range through a k-Means approach to make them fit the objects size of the dataset. We have used Yolov5. Unfortunately, the R^2 achieved is 0.34 and MAE of 18.83.

In this application case, YOLO presents three main drawbacks: 1) it cannot count two or more whiteflies that are too close, it only counts one; 2) Non-Maximal Suppression (NMS) algorithm cannot fuse properly different bounding box in different scales, therefore the number of

Fig. 17. Left) Example of perfect whitefly counting; right) leaf with some false positives in the upper part of the leaf due to small white colored damages.

Fig. 18. Left) example of wrong counting due to bad leaf segmentation; right) example of leaf with no counting due to unfocused image.

detected insects in doubled; 3) it detects some damage regions as whiteflies, thus adding false positives to the counting. Two figures have been added to illustrate these situations.

As it has been explained before, in our solution every found gaussian peak is marked with a red dot and rounded with a blue circle for understanding purposes. Notice that this gaussian can correspond to more than one insect in case they appear very close one to the other. In Fig. 19, it can be appreciated that YOLO model cannot detect all these small objects. However, our approach can detect properly almost all the whiteflies.

In next Fig. 20, a leaf image with higher infestation level is shown. It can be appreciated that many whiteflies are not detected with YOLO solution whereas our approach can do it.

We can conclude that, in this specific application, our proposed density map estimation approach can perform better than an object detection-based solution, such as YOLO, and also better than a candidate selection and classification approach.

7. Conclusions

In this work, leaf segmentation solution and two approaches for insect counting have been presented. First approach is based on candidate identification and deep learning-based classification. The second approach is a novel method derived from density map estimation. The algorithm was validated in eggplant leaves for alive adult whitefly insect counting. Methodology has proven to be feasible. The density-based estimation solution outperforms the candidate selection approach based on feature extraction and further classification, even if that

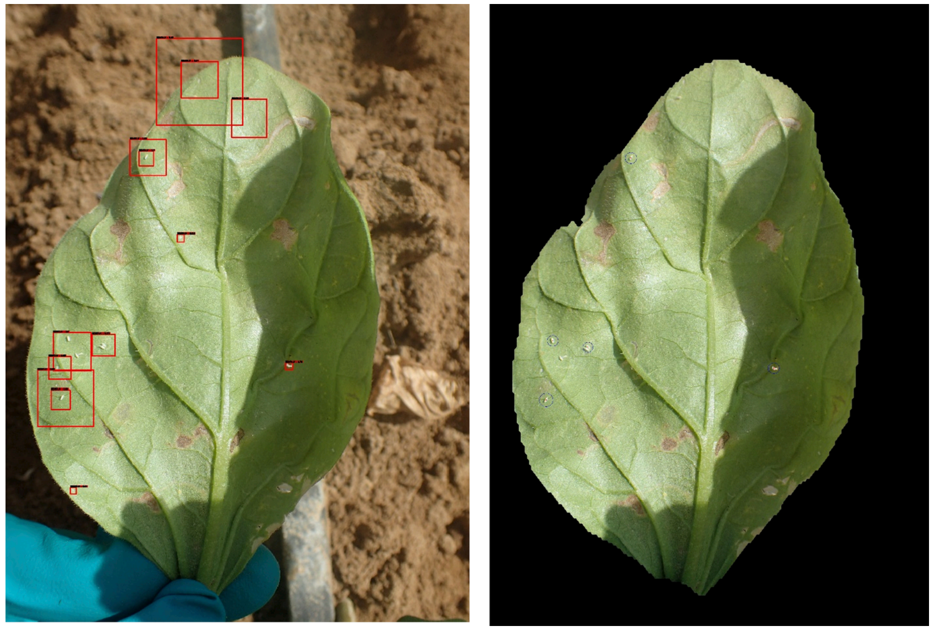

Fig. 19. Left) output image with YOLO model; right) output image with our density map estimation approach.

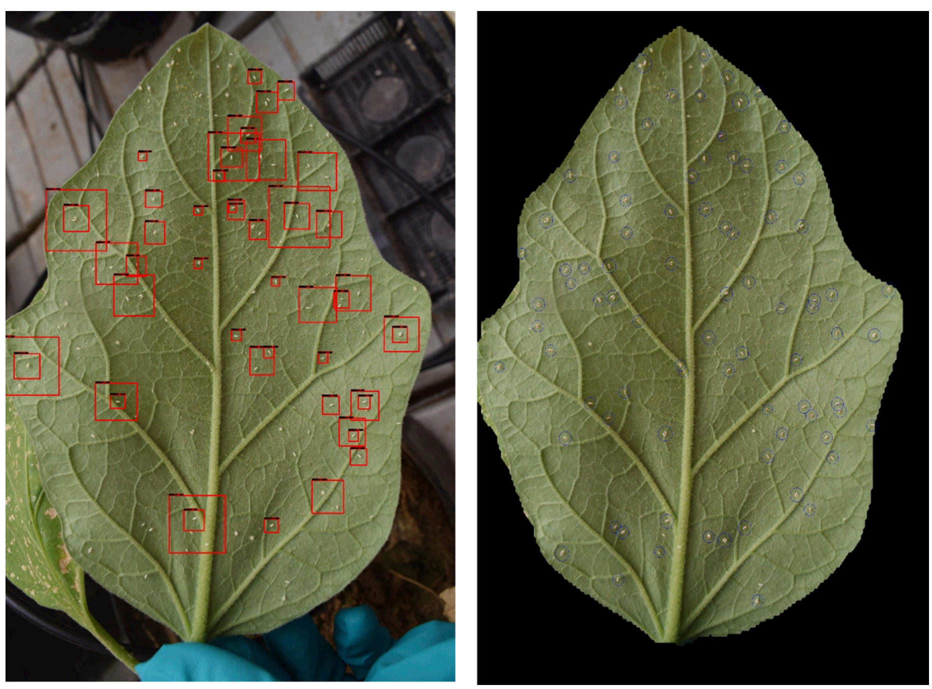

Fig. 20. Left) output image with YOLO model; right) output image with our approach.

classifier is also a deep learning-based model.

The designed and developed algorithm allows real time performance with the following pipeline: leaf segmentation (background removal) and density map estimation for counting. The network models have been developed with deep learning-based techniques. The results on real field tests obtained MAE of 3.36, RMSE of 7.84 and R^2 of 0.97 values.

The algorithm has been deployed on a real smartphone application and exhaustively validated under real field conditions in a pilot study located in Spain and Germany. The dimensions of these leaves may vary from few centimeters up to 20–30 cm. The size and appearance of the insects are very changeable due to the variability inherent to a capture done with the mobile in wild conditions. Leaf size and acquisition distance and orientations is different in every image. The premise is to acquire the whole leaf area in the most perpendicular way and trying to focus correctly. Moreover, the whitefly is alive so it can move and provoke unfocused insects.

The future work could tackle the currently identified limitations shown in section 6 and other researching lines. On the one hand, further work could validate and extend the methodology to other types of pests, with similar or different appearance, and over leaves of different crops. At the same time, it would be interesting to validate the performance of the model in leaves with different infestation degrees. On the other hand, it could be tackled the efficiency and deployment of the model. To this aim, it could be addressed the migration of the models to embedded stations (such as mobiles) to make the application independent of internet connections and with adequate inference time. Another line of future work, among other possible ideas, is the development of models achieving similar performance with a reduced number of training images.

Uncited references

CRediT authorship contribution statement

Arantza Bereciartua-Pérez: Conceptualization, Formal analysis, Investigation, Software, Writing – original draft. **Laura Gómez:** Investigation, Software, Writing – review & editing. **Artzai Picón:** Conceptualization, Investigation, Methodology, Software, Writing – review & editing. **Ramón Navarra-Mestre:** Conceptualization, Investigation, Writing – review & editing. **Christian Klukas:** Conceptualization, Investigation. **Till Eggers:** Conceptualization, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank all field researchers that generated the dataset, carried out the annotation process, performed the validation in the wild, and in general, supported the work in Tecnalia and BASF specially to Javier Romero, Carlos Javier Jiménez, Amaia Ortiz, Aitor Alvarez, Jone Echazarra and Corinna Spangler.

References

Aich, S., Stavness, I., 2018. Object Counting with Small Datasets of Large Images. Comput. Vis. Pattern Recognit. 1–11.

Akintayo, A., Tylka, G.L., Singh, A.K., Ganapathysubramanian, B., Singh, A., Sarkar, S., 2018. A deep learning framework to discern and count microscopic nematode eggs. Sci. Rep. 8, 1–11. https://doi.org/10.1038/s41598-018-27272-w.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2020. Few-Shot Learning approach for plant disease classification using images taken in the field. Comput. Electron. Agric. 175, 105542.

Arteta, C., Lempitsky, V., Zisserman, A., 2016. Counting in the wild. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 483–498. https://doi.org/10.1007/978-3-319-46478-7_30.

Barbedo, J.G.A., 2014. Using digital image processing for counting whiteflies on soybean leaves. J. Asia. Pac. Entomol. 17, 685–694. https://doi.org/10.1016/j.aspen.2014.06.014.

Bjerge, K., Nielsen, J.B., Sepstrup, M.V., Helsing-Nielsen, F., Høye, T.T., 2020. A light trap and computer vision system to detect and classify live moths (Lepidoptera) using tracking and deep learning. bioRxiv 2020.03.18.996447.

Chattopadhyay, P., Vedantam, R., Selvaraju, R.R., Batra, D., Parikh, D., 2017. Counting everyday objects in everyday scenes. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-Janua, 4428–4437. https://doi.org/10.1109/CVPR.2017.471.

Cohen, J.P., Lo, H.Z., Bengio, Y., 2016. Count-ception : Counting by Fully Convolutional Redundant Counting. International Conference on Computer Vision (ICCV). 18–26.

Comaniciu, D., Meer, P., 1998. Bimodal System for Interactive Indexing and Retrieval of Pathology Images. Proc. Fourth IEEE Work. Appl. Comput. Vis. Cat. No.98EX201, 268–269. https://doi.org/10.1109/ACV.1998.732900.

Fuentes, A., Yoon, S., Kim, S., Park, D., 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors (Switzerland) 17 (9), 2022.

Gabriel de Oliveira, E.C.M., Gabriel de Oliveira, Elisabete C. Moraes, Nathaniel A. Brunsell, Y.E.S., Nathaniel A. Brunsell, Y.E.S., Luiz E.O.C. Aragão, G.A.V.M. and, Santos, G.A.V.M. and T.V. dos, Santos, T.V. dos, Ava, A. information is, 2015. Agricultural robot for intelligent detection of pyralidae insects, in: Agricultural Robots. IntechOpen, p. 64. https://doi.org/10.5772/32009.

Guirado, E., Tabik, S., Rivas, M.L., Alcaraz-Segura, D., Herrera, F., 2018. Automatic whale counting in satellite images with deep learning. bioRxiv 443671. https://doi.org/10.1101/443671.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S., 2016. Deep learning for visual understanding: A review. Neurocomputing 187, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116.

Hamwood, J., Alonso-Caneiro, D., Sampson, D.M., Collins, M.J., Chen, F.K., 2019. Automatic detection of cone photoreceptors with fully convolutional networks. Transl. Vis. Sci. Technol. 8, 1–8. https://doi.org/10.1167/tvst.8.6.10.

Heinrich, K., Roth, A., Zschech, P., 2019. Everything Counts: A Taxonomy of Deep Learning Approaches for Object Counting Maintenance Analytics View project Understanding and Interpreting Deep Learning Networks View project 0–16.

Høye, T.T., Ärje, J., Bjerge, K., Hansen, O.L.P., Iosifidis, A., Leese, F., Mann, H.M.R., Meissner, K., Melvad, C., Raitoharju, J., 2021. Deep learning and computer vision will transform entomology. Proc. Natl. Acad. Sci. U. S. A. 118, 1–10. https://doi.org/10.1073/PNAS.2002545117.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Huang, Z., Leng, J., 2010. Analysis of Hu's moment invariants on image scaling and rotation. ICCET 2010–2010 Int. Conf. Comput. Eng. Technol. Proc. 7 https://doi.org/10.1109/ICCET.2010.5485542.

Ilyas, N., Shahzad, A., Kim, K., 2020. Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation. Sensors (Switzerland) 20 (1), 43.

Johannes, A., Picon, A., Alvarez-gila, A., Echazarra, J., Rodriguez-vaamonde, S., Díez, A., Ortiz-barredo, A., 2017. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput. Electron. Agric. 138, 200–209. https://doi.org/10.1016/j.compag.2017.04.013.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

Kang, D., Ma, Z., Chan, A.B., 2018. Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks - Counting, Detection, and Tracking. IEEE Trans. Circuits Syst. Video Technol. XX, 1–14. https://doi.org/10.1109/TCSVT.2018.2837153.

Karar, M.E., Alsunaydi, F., Albusaymi, S., Alotaibi, S., 2021. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. Alexandria Eng. J. 60, 4423–4432. https://doi.org/10.1016/j.aej.2021.03.009.

Lempitsky, V., Zisserman, A., 2010. Learning To Count Objects in Images. Neural Inf. Process. Syst. 1–9.

Li, R., Wang, R., Xie, C., Liu, L., Zhang, J., Wang, F., Liu, W., 2019a. A coarse-to-fine network for aphid recognition and detection in the field. Biosyst. Eng. 187, 39–52. https://doi.org/10.1016/j.biosystemseng.2019.08.013.

Li, W., Chen, P., Wang, B., Xie, C., 2019b. Automatic Localization and Count of Agricultural Crop Pests Based on an Improved Deep Learning Pipeline. Sci. Rep. 9, 1–11. https://doi.org/10.1038/s41598-019-43171-0.

Lins, E.A., Rodriguez, J.P.M., Scoloski, S.I., Pivato, J., Lima, M.B., Fernandes, J.M.C., da Silva Pereira, P.R.V., Lau, D., Rieder, R., 2020. A method for counting and classifying aphids using computer vision. Comput. Electron. Agric. 169, 105200 https://doi.org/10.1016/j.compag.2019.105200.

Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., Zhu, X., 2016. Detection of aphids in wheat fields using a computer vision technique. Biosyst. Eng. 141, 82–93. https://doi.org/10.1016/j.biosystemseng.2015.11.005.

Liu, X., van de Weijer, J., Bagdanov, A.D., 2018. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. https://doi.org/10.1109/CVPR.2018.00799.

Lu, H., Cao, Z., Xiao, Y., Zhuang, B., Shen, C., 2017. TasselNet: Counting maize tassels in the wild via local counts regression network. Plant Methods 13, 1–14. https://doi.org/10.1186/s13007-017-0224-0.

Marsden, M., McGuinness, K., Little, S., O'Connor, N.E., 2017b. ResnetCrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017. https://doi.org/10.1109/AVSS.2017.8078482.

Nieuwenhuizen, A., Hemming, J., Suh, H., 2018. Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN. Proc. Netherlands Conf. Comput. Vis. 1–4.

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2020. Deep Learning vs. Traditional Computer Vision. Adv. Intell. Syst. Comput. 943, 128–144. https://doi.org/10.1007/978-3-030-17795-9_10.

Oñoro-Rubio, D., López-Sastre, R.J., 2016. Towards Perspective-Free Object Counting with Deep Learning. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 615–629.

Patrício, D.I., Rieder, R., 2018. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Comput. Electron. Agric. 153, 69–81. https://doi.org/10.1016/j.compag.2018.08.001.

Picon, A., Alvarez-gila, A., Seitz, M., Ortiz-barredo, A., Echazarra, J., Johannes, A., 2019a. Deep convolutional neural networks for mobile capture device-based crop disease classi fi cation in the wild. Comput. Electron. Agric. 161, 280–290. https://doi.org/10.1016/j.compag.2018.04.002.

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., Echazarra, J., 2019b. Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. Comput. Electron. Agric. 167, 105093.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2015. You Only Look Once: Unified. Real-Time Object Detection. https://doi.org/10.1109/CVPR.2016.91.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Roosjen, P.P.J., Kellenberger, B., Kooistra, L., Green, D.R., Fahrentrapp, J., 2020. Deep learning for automated detection of Drosophila suzukii: potential for UAV-based monitoring. Pest Manag. Sci. 76, 2994–3002. https://doi.org/10.1002/ps.5845.

Russel, B.C., Torralba, A., Murphy, K.P., Freeman, W.T., 2008. 2008 LabelMe. Int. J. Comput. Vis. 77, 157–173. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-407.

Rustia, D.J.A., Chao, J.-J., Chiu, L.-Y., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., Lin, T.-T., 2021. Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. J. Appl. Entomol. 145, 206–222. https://doi.org/https://doi.org/10.1111/jen.12834.

Segui, S., Pujol, O., Vitria, J., 2015. Learning to count with deep object features. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. 2015-Octob, 90–96. https://doi.org/10.1109/CVPRW.2015.7301276.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. 1–14.

Song, H., Liang, H., Li, H., Dai, Z., Yun, X., 2019. Vision-based vehicle detection and counting system using deep learning in highway scenes. Eur. Transp. Res. Rev. 11 https://doi.org/10.1186/s12544-019-0390-4.

Tayara, H., Soo, K.G., Chong, K.T., 2017. Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network. IEEE Access 6, 2220–2230. https://doi.org/10.1109/ACCESS.2017.2782260.

Tian, H., Wang, T., Liu, Y., Qiao, X., Li, Y., 2020. Computer vision technology in agricultural automation —A review. Inf. Process. Agric. 7, 1–19. https://doi.org/10.1016/j.inpa.2019.09.006.

Wang, Q.J., Zhang, S.Y., Dong, S.F., Zhang, G.C., Yang, J., Li, R., Wang, H.Q., 2020. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. Comput. Electron. Agric. 175, 105585 https://doi.org/10.1016/j.compag.2020.105585.

Xie, W., Noble, J.A., Zisserman, A., 2016. Microscopy cell counting and detection with fully convolutional regression networks. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization. https://doi.org/10.1080/21681163.2016.1149104. In press.

Zhang, Q., Liu, Y., Gong, C., Chen, Y., Yu, H., 2020. Applications of deep learning for dense scenes analysis in agriculture: A review. Sensors (Switzerland) 20, 1–33. https://doi.org/10.3390/s20051520.

Zhong, Y., Gao, J., Lei, Q., Zhou, Y., 2018. A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors (Switzerland) 18 (5), 1489.