A Low-cost "Plant-Scanner" Platform for Automated Detection of Ustilago Maydis Infection in Maize Using Deep Learning

Marvin Christ
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Seyed Amir Hossein Tabatabaei
seyed.amir.hossein.tabatabaei@ges.thm.de
Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan
Niklas Ostwald
Faculty of Health Sciences, University of Applied Sciences (Technische Hochschule Mittelhessen)
Oskar Seifert
Faculty of Health Sciences, University of Applied Sciences (Technische Hochschule Mittelhessen)
Itzel Rubio Elizalde
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Paul Klemm
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Clemens Thölken
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Marcus Lechner
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Nasim Faridnia
Department of Computer Science, University of Texas
Gert Bange
Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg
Keywan Sohrabi
Faculty of Health Sciences, University of Applied Sciences (Technische Hochschule Mittelhessen)

Article

Keywords:

Posted Date: April 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8667319/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on July 11th, 2026. See the published version at https://doi.org/10.1038/s41598-026-60714-4.

A Low-cost "Plant-Scanner" Platform for Automated Detection of *Ustilago Maydis* Infection in Maize Using Deep Learning

Marvin Christ[1,$], Seyed Amir Hossein Tabatabaei[2,3,*,$], Niklas Ostwald[3,$], Oskar Seifert[3], Itzel Rubio Elizalde[1], Paul Klemm[1], Clemens Thölken[1], Marcus Lechner[1], Nasim Faridnia[4], Gert Bange[1,5,+], and Keywan Sohrabi[3,+]

[1]Center for Synthetic Microbiology (SYNMIKRO) and Department of Chemistry, University of Marburg, Karl-von-Frisch Strasse 14, 35043 Marburg, Germany
[2]Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan, Namjoo St. 41938-33697, Rasht, Iran
[3]Faculty of Health Sciences, University of Applied Sciences (Technische Hochschule Mittelhessen), Wiesenstrasse 14, 35390, Giessen, Germany
[4]Department of Computer Science, University of Texas, San Antonio, USA
[5]Max-Planck-Institute for Terrestrial Microbiology, Karl-von-Frisch Strasse 14, 35043 Marburg, Germany
[*]Corresponding author (Email: seyed.amir.hossein.tabatabaei@ges.thm.de, amirhossein.tabatabaei@guilan.ac.ir)
[$]These authors contributed equally to this work and share first authorship.
[+]These authors have contributed equally to this work.

ABSTRACT

Ustilago maydis is a biotrophic fungus that causes smut disease in maize, leading to tumor formation on aerial parts of the plant. While *U. maydis* has been a model for plant-fungal interaction studies, no tool has existed to automatically quantify infection symptoms under laboratory conditions for deep learning analysis. To address this, we developed a rotating camera system that captures videos of plants under customized lighting and shutter settings. These videos were used to train machine learning models to distinguish between healthy and infected plants. Two machine learning models have been presented. In the first approach, by employing a naive masking technique and combining classical machine learning with deep learning classifiers, the model achieved a reasonable performance, with an Area Under the Curve (AUC) of 0.90 on the Receiver Operating Characteristic (ROC), displaying high sensitivity and specificity. The second approach utilizes pre-trained YOLO11 model for object detection and further classification. The YOLO11-based approach outperforms traditional methods, achieving near-perfect accuracy (AUC: 0.99-1.00), demonstrating its superiority for real-time, scalable applications. Our toolset, featuring a cost-efficient and customizable scanning platform with open building-blocks design, provides a valuable resource for unbiased disease symptom detection and scoring, with potential applications in other plant pathology studies. This point enables easy replication and adaptation by other research laboratories which makes the platform robust, scalable and practical beyond our specific application.

1 Introduction

1.1 The Corn Smut Fungus *Ustilago Maydis* as a Model System to Study Biotrophic Pathogens

Eradicating plant diseases presents a significant challenge in agriculture and global food security, as these cause an approximate annual loss of up to 40% global crop production [1, 2]. This problem is set to increase as temperatures rise, promoting the spread of pathogens such as fungi [3]. To address this problem, the study and understanding of plant-fungi interactions through model organisms is required for the development of accurate solutions. The biotrophic fungus *Ustilago maydis* has been included in the list of top ten fungal model organisms that have been established to study phytopathogen infection [4]. This is not particularly because of its economic importance, but because *U. maydis* is well-suited for genetics and molecular biology approaches, its genome is completely annotated and accessible for almost twenty years now [5] and genetic manipulation via homologous recombination and/or CRISPR/Cas9 is well-established [6, 7, 8]. Furthermore, it is a representative of obligate biotrophic fungi which include smut, rust, and powdery mildew fungi. *U. maydis* specifically infects aerial parts of maize (*Zea mays*; Figure 1 : A-E) and teosinte and triggers the formation of large tumors filled with diploid teliospores [9] *Ustilago maydis*). This pathosystem has been the subject of intense research for more than three decades attaining insight into different

effector proteins secreted by *U. maydis* in order to manipulate the host plant (e.g. [10, 11, 12] and how maize counteracts these challenges (e.g. [13, 14]). Disease severity is often evaluated on young maize seedlings according to criteria developed by Kämper et al., 2006 ([5]) (Figure 1 : F). Infection symptoms are assessed by eye and categorized according to different criteria (Figure 1 : F). This process is time-consuming, labor-intensive and prone to subjective or biased classification. An automated symptom-detection system combined with a deep learning model provides a valuable aid for members of the *U. maydis* research community and allows for comparable results in a uniform system. Recent works [15, 16] underscore the need for automated tools to study *U. maydis* effector proteins, which our platform addresses.

1.2 Plant Disease Detection by RGB imaging

Imaging in the visible light spectrum enables the large-scale, cost-effective and uncomplicated detection of plant disease symptoms (RGB imaging) [17, 18, 19]. RGB images can be used to extract the specific features for a particular type of plant infection of interest thereby enabling classification. Major features for classification are color, shape (reflected by changes in volume) and texture (e.g. homogeneity) [20]. RGB imaging of infected plants can then be used for training algorithms for plant disease detection in systematic studies [21]. Any state of the art digital camera is sufficient for many approaches, although other sensors have proven useful depending on the approach [18, 19, 22]. Selection of a suitable sensor depends not only on the desired level of detail accuracy of the features but also on factors such as acquisition costs, usability, integration capability, environmental influences and the object to be examined. Hyperspectral imaging and RGB imaging have proven to be particularly suitable in the last 10 years, as can be seen from existing image databases of infected plants used for training algorithms for plant disease detection in systematic review [21]. In laboratory environments or plant facilities, static or swiveling camera systems fixed on a tripod construction are often used [23, 24, 25]. Cell phone apps for detecting plant infections have also been developed in this context. Images or videos taken with the cell phone camera are then analyzed on the device in real time or via a connected data center [26, 27]. Commercial systems such as the "WIWAM Phenotyping System" [28] or the "Photon Systems Instruments Plant Screen System" [29] were developed by others. Drones have been used for larger agricultural areas, which stand out from other systems in particular due to their spatial flexibility [30]. There are also approaches with mobile ground robots [31]. Furthermore, there are also mobile rail systems with a frame on top that travels over different parts of the field, both for large and small agricultural areas [32, 33]. However, we see a need for cost-efficient systems, which laboratories can build on their own.

1.3 Plant Disease Detection Methods based on the Machine learning and Pattern Recognition

Automated detection and analysis of plant diseases and pest identification are mainly based on image processing techniques, computer vision, traditional machine learning algorithms, and deep learning. In image processing and computer vision, conventional image processing algorithms and/or manual design of features are used, with the occasional use of miscellaneous classifiers [34, 35, 36]. Methods based on machine learning often begin with image processing techniques to enhance images and extract relevant features. Common techniques include image segmentation, edge detection, and color space transformations. These preprocessing steps help in detecting and isolating the infected regions and extracting discriminative features for subsequent analysis. In such methods, different properties of plant diseases and pests are used. However, fine-tuning the system, including adjusting the light source and shooting angle to achieve uniform illumination, is essential-though it can be a challenging task [37]. Image processing-based methods detect and classify the infected parts of the leaves by applying pre-processing techniques followed by clustering and segmentation of the leaves. The features including three key types, namely, color, shape, and texture are extracted to feed the possible augmented classifier. Several traditional machine learning algorithms have been applied to infection detection on plant leaves. These include decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and random forests. These algorithms can effectively classify infected and healthy leaf regions based on extracted features. The proposed work in [20] presents a survey reviewing the literature associated with plant disease detection based on machine learning. According to this work, the methods detect and classify the diseases with regard to the extracted feature types and the utilized machine learning model. For example in [38], an HSI color conversion followed by color slicing technique is used to extract regions of interest achieving an accuracy of 0.966 [20, 38]. Alternatively, other color features including YCbCr, HIS and CIELAB can be used to recognize disease spots [39]. The work presented in [40] utilized image color analysis and k-means clustering technique to detect and identify some special rice disease spots and leaves. The authors of [41] have engaged the naive statistics of a wide variety of color image features augmented by traditional classifiers like SVM, KNN, and RF to achieve an accuracy of about 0.9465. Alternative works using color features for disease area segmentation can be named as [40, 42, 43, 44]. As another example, color, shape and texture features including geometrical and texture indicators like area, perimeter, contrast, entropy, etc., have been extracted from the color features of the image in [45]. The feature set has been augmented by an SVM classifier achieving the accuracy about 0.972. In the work presented in [46], three rice leaf diseases have been recognized using a combination of all features including color, shape, and texture types. The SVM classifier has been applied on the image wherein some initial background removals were applied to achieve the accuracy of about 0.8857. Other works combining the features can be mentioned as [20, 47, 48, 49]. All of these methods and similar

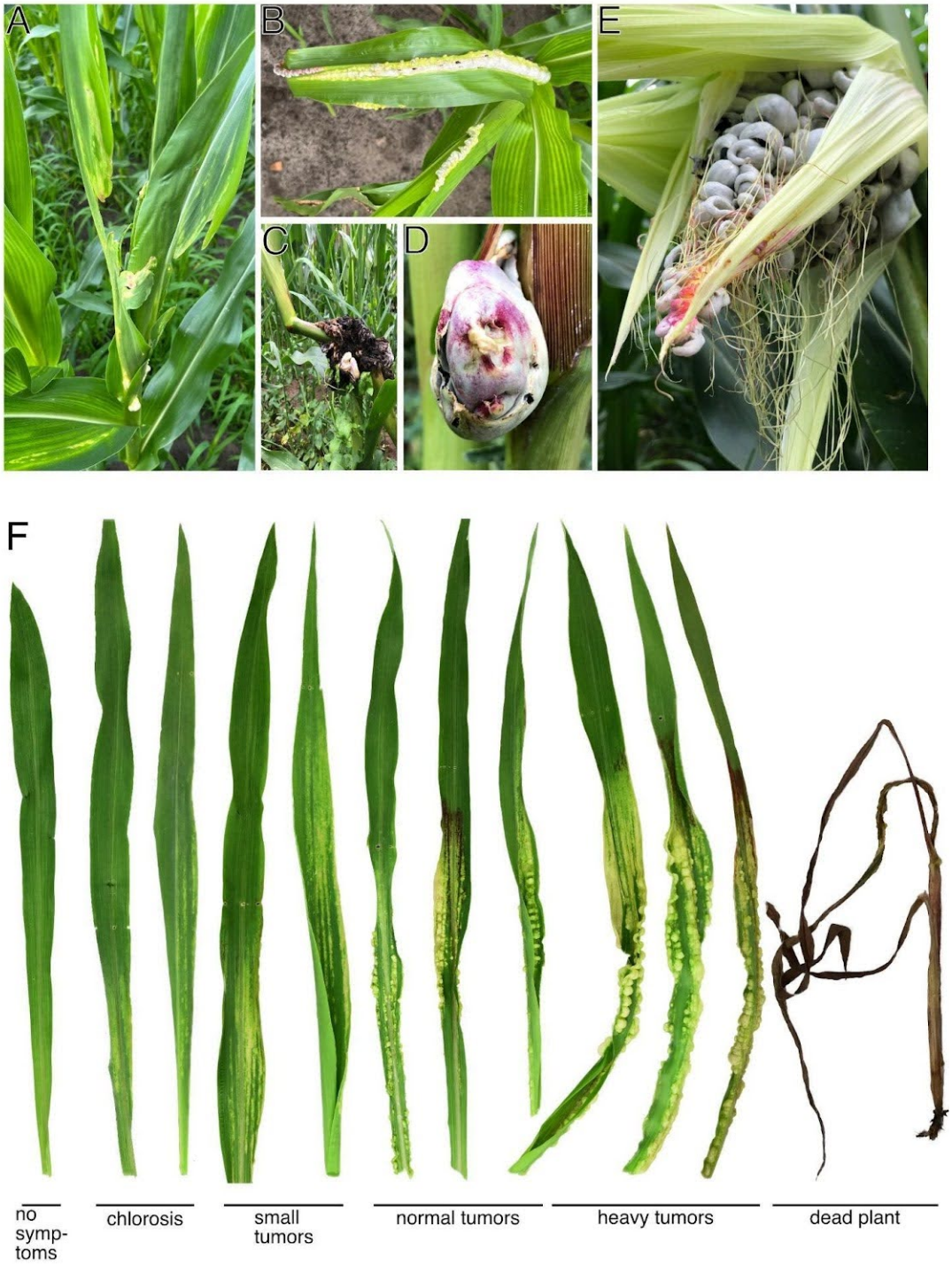


Figure 1. Infection symptoms of *U. maydis* on maize plants in the field and in the greenhouse. Panels A-E show field-cultivated maize plants with natural infection symptoms in increasing stages from chlorosis to spore-filled galls. The pictures were taken during summertime in corn fields in Germany. Panel F exemplifies the different infection symptoms in artificially infected maize plants cultivated under controlled conditions in a greenhouse.

ones may struggle with complex and high-dimensional feature spaces and may not generalize well to unseen data. Also, it might be very difficult, even unrealistic to expect the classical image processing algorithms to completely eliminate the impact of background and/or scene change on the detection and recognition results [37]. There is a lot of instability when collecting plant disease and pest images shed by the natural light [34]. The aforementioned concerns as well as the incapability of classical and traditional image processing methods in effective feature extraction motivate the domain experts to elaborate the feature engineering part in disease detection and classification pipeline by the use of deep learning. Deep learning methods, particularly convolutional neural networks (CNNs), have shown remarkable success in various image analysis tasks, including infection detection on plant leaves. They have been applied to detect and classify the region of interest (infected parts) throughout two sequential phases consisting of the feature map generation and the classification followed. CNNs can automatically learn hierarchical representations from raw leaf images, eliminating the need for handcrafted features extraction. Transfer learning, where pre-trained models are fine-tuned on plant leaf datasets, has also proven effective in cases with limited training data. Using models from the CNN-based architecture YOLO (you only look once) [50] has been shown to have promising results. Works recently presented in [51, 52, 53] are some examples.

A survey is given in [54] wherein the leaf disease detection methods based on deep learning and traditional machine learning as well as the challenges, limitations and applications are discussed. Similar survey has been presented in [55] in which some recommendations to the farmers as well as researchers have been given as well. The presented survey in [56] analyses the plant disease detection works based on traditional image processing and deep learning approaches as well. The methods are not limited to the individual ones stated as above and the hybrid methods which combine the aforementioned techniques are used as well. For example, machine learning algorithms, such as SVM and deep learning models, can be applied to the hyperspectral image data which captures spectral information at multiple wavelengths to identify spectral traces associated with infections. As another example, proposed framework in [57] incorporates several different hybrid deep learning models containing pre-trained models to detect tomato early blight disease. The proposed framework performs extraordinary performance and achieves the accuracy in the range of 87.55 up to 1.00.

Although the shortlisted methods are utilized in order to detect the leaf disease in several different plants, there are some works which have been dedicated to detect the diseases appearing on maize leaves [58, 59]. The proposed work in [58] utilizes traditional machine learning algorithms to detect the maize leaf diseases based on the plant leaf images. The highest accuracy has been achieved via random forest classifier. The work presented in [59] utilized the architecture of some pre-trained models for detection of three different leaf disease symptoms in maize plants. Also, they have embedded the detection framework in a mobile application for ease of use. With the emergence of attention mechanisms and transformer architectures, many plant disease detection and classification schemes have been developed upon these baselines. For example, the presented works in [60, 61] use Vision Transformers (ViTs) in detecting diseases in plant images captured in natural environments. The presented work in [62] enhances interpretability in detecting and classifying leaf disease based on ViTs. The work presented in [63] uses ensemble technique in integrating several deep learning architectures and transformers to identify and classify leaf diseases. With the combination of CNN and transformers, the proposed work in [64] is able to detect the plant, leaf disease and well as severity of disease. Plant disease detection in Internet of Things (IoT) has been proposed based on similar combination approach in [65].

2 Materials and Methods

2.1 Plant-Sample Generation

One to five seeds of the *Zea mays* cultivar 'Early Golden Bantam' (Urban Farmer) were planted 5 - 10 cm under the surface in 14 cm diameter and 11.5 cm high pots filled with peat moss (Heinrichs Agrar). Plants were grown in a plant chamber with 60% humidity, day/night temperatures of $28^\circ/20^\circ C$, and 14 hs photoperiod, with watering as needed. Under these conditions, after 7 - 8 days of growth, the plants reach the three-leaf stage and a height of about 15 cm. *Ustilago maydis* strains were grown in YEPSlight (1% w/v yeast extract, 0.4%w/v peptone, and 0.4% w/v sucrose) with a starting OD_{600} = 0.2. Upon reaching OD_{600} = 0.6 - 0.8 the cells were harvested and resuspended in sterile water to OD_{600} = 1. 500 μL were syringe-infected into the stems of the maize seedlings [5]. The videos of the plants were made over a period of up to 12 days post infection at various time points.

2.1.1 Video-Sample Generation of Plants

During the development of the hardware, adjustments were continuously made to improve the quality of the individual raw images for better algorithm results. As listed here, this results in four different development phases of the scanner platform. In the following, we describe the development and improvement of the scanner, i.e., the intermediate stages of the project. For a detailed list of specifications, see the Results chapter phase four.

2.1.2 Phase One

Like in the PlantVillage [66] database, we started to take pictures of infected plants with private cell phones. Recording parameters such as angles and distances were not uniformly chosen. The sodium vapor lamps in the plant chamber were the only light source, which made the images appear yellowish. This had a detrimental effect on the recognition of the fungal symptoms on the plant since some of them are naturally yellowish. Next, a small scanning device was designed in which a conventional cell phone was placed horizontally to the plant. In this setup, the camera was fixed, and the plant rotated around the camera. The rotation and vibrations caused exposed parts of the plant to shake, resulting in blurred video footage (no significant amount of video material was obtained).

2.1.3 Phase Two

For phase 2 (Figure 2), a platform was developed that uses a designated cell phone as a camera to move 360° around the plant to capture sharp images of the plant standing as still as possible. The ball-bearing rotary unit was driven by a stepper motor and an Arduino so that the desired rotational speed could be achieved. In order to achieve a constant light setting and thus remove unwanted background light, a conventional black cardboard paper was glued to the frame of the scanner as background. An adjustable plant platform was developed to position the plant to be scanned at the optimum height. The camera suspension could also be adjusted in height.

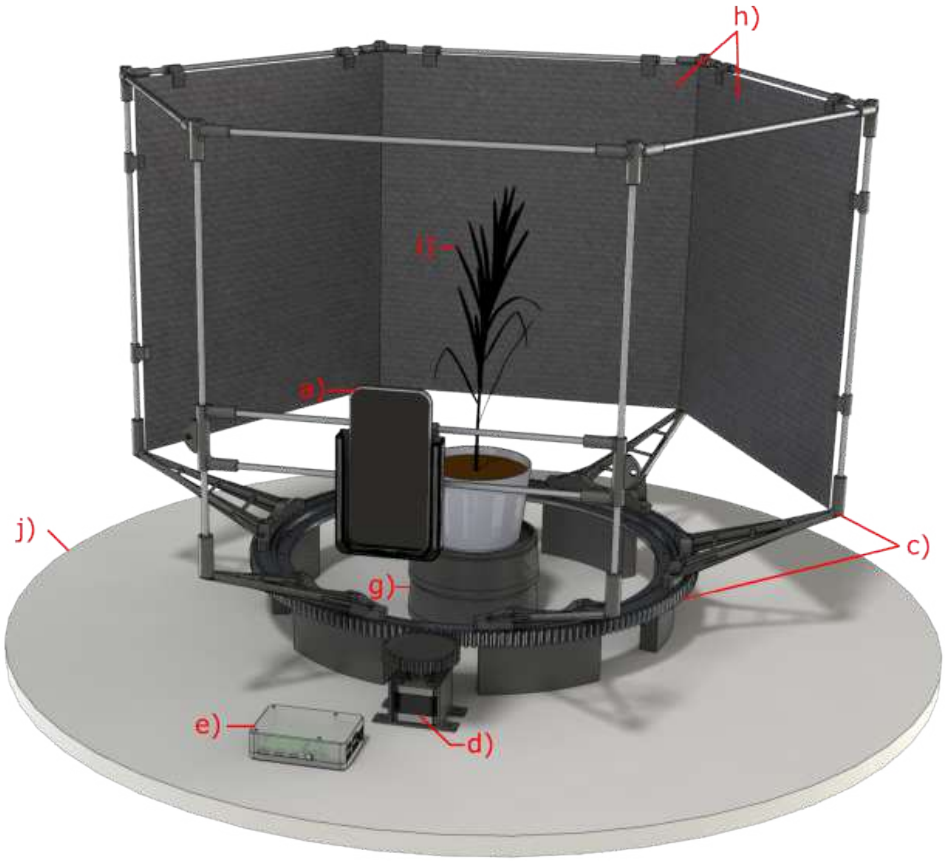


Figure 2. Phase Two Plantscanner for 360° image capturing of the infected plant with 150 L scan volume: a) side-view camera (iPhone XRMobilephone), c) 360° camera rotation unit, d) stepper motor, e) control unit (Arduino), g) adjustable plant platform, h) high-contrast background (black cardboard), i) scan object, j) base plate

2.1.4 Phase Three

In phase 3, we opted for an Insta360 Go 2 action camera, which is suitable for capturing images in motion/vibration. We also increased the distance between the camera and the plant to capture entire plants at an advanced stage of growth. In addition, we installed an LED light source with adjustable light intensity to generate a consistent light environment. The description is depicted in Figure 3.

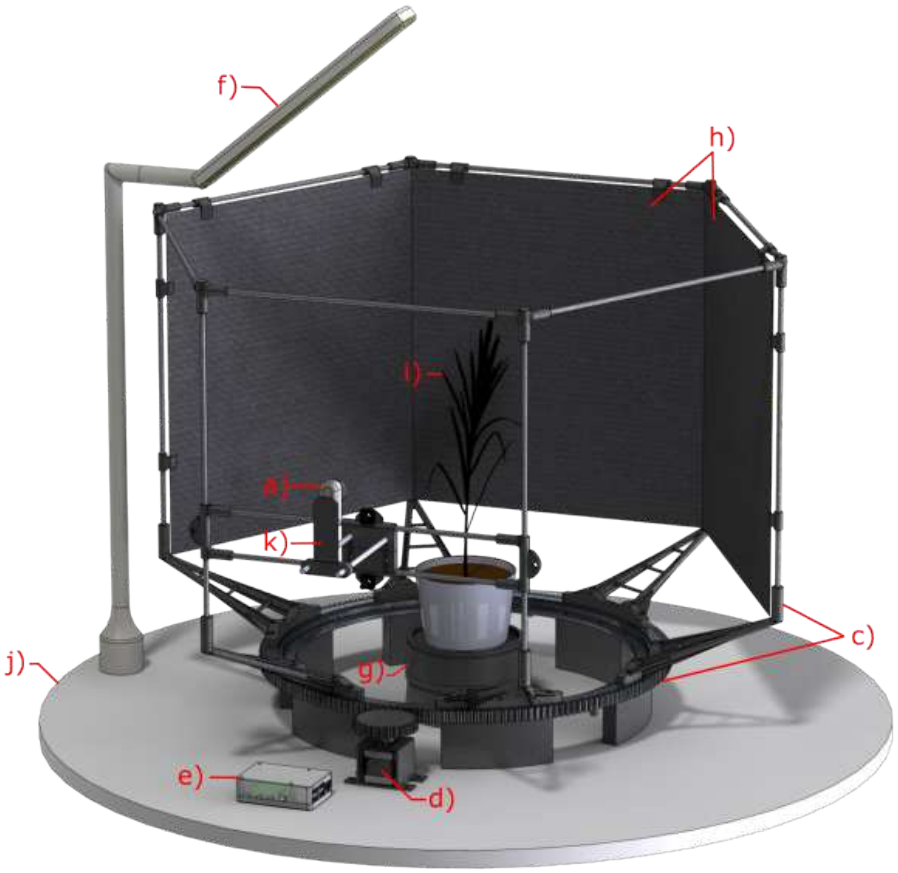


Figure 3. Phase Three Plantscanner for 360° image capturing of the infected plant with 150 L scan volume: a) side-view camera (Insta360 Go 2), c) 360° camera rotation unit, d) stepper motor, e) control unit, f) lamp, g) adjustable plant platform, h) high-contrast background, i) scan object, j) base plate, k) adjustable camera rail

2.1.5 Phase Four

Phase four was the final development phase and served as a validation phase for the algorithm. In this phase, the bottom of the scanner was also lined with black cardboard paper as a final improvement. This ensured that even large leaves, which are not held horizontally by the plant and bend downwards, could be captured on a uniform background. This also made it possible to record video footage with a camera pointing vertically at the plants from above, for which we added a Raspberry Pi camera on top due to its ease of use and modularity. This top-view video material has not yet been used to analyze the plants.

3 Approach 1: The Model for Detecting the Infected Areas of the Plants Based on Image Segmentation

3.1 Segmentation Algorithm

The proposed algorithm for segmenting and detection is based on the frame analysis of each recording wherein the segmentation analysis is applied to each frame of the video individually. The segmentation algorithm consists of three main steps including pre-processing of the video, region of interest segmentation, and removal of miss-segmented areas followed by extraction of

required statistical features. The pre-processing step consists of noise removal of the video, cropping and frame-extraction. In the second step, the possible infected parts of the plant will be segmented. The segmentation algorithm is performed based on masking/filtering operation, morphological operation and thresholding applied on each extracted frame. In the third step, the segmented frames will be analyzed, and some statistical features will be handcrafted for the sake of candidate selection phase. Description of the segmentation algorithm is depicted in Algorithm 1.

Algorithm 1 Segmenting the infected leaves

Input: Video V recorded by an embedded moving camera in the platform depicted in Figure 3
Output: Segmented infected areas $R_{Infected}$ of the leaves

START Algorithm

1. **Pre-processing step:**
   a. **Frame extraction :** $F = \{f_1, \ldots, f_n\}$
   b. **Frame cropping and contrast adjustment and histogram equalization:**
   F : $F_{PreProcessing} = \{f'_1, \ldots, f'_n\}$
2. **Detection and localization:**
   a. **Convert RGB frames to HSV frames:**
   $G = RGB2HSV(F_{PreProcessing}) = \{g_1, \ldots, g_n\}, where, g_i = RGB2HSV(f_i)$
   b. **Apply yellow and dark-yellow masking operation on HSV frames:**
   $H = MASK(G) = \{h_1, \ldots, h_n\}, where\, h_i = MASK(g_i)$
   c. **Apply thresholding for binary segmentation:**
   $B = BIN(G, threshold) = \{b_1, \ldots, b_n\}, where, b_i = BIN(h_i, threshold_i)$
3. **Removing the outliers:**
   a. **Apply morphological operation:**
   $M = MORPH(B, Obj) = \{M_1, \ldots, M_n\}, where\, M_i = MORPH(M_i, Obj)$
4. **Output the results:** $R_{Infected} = \{R_1, \ldots, R_n\}$

END Algorithm

3.2 Candidates Selection

The handcrafted features in the presented segmentation algorithm above are used to feed the candidate selection algorithm to label the plant as infected or non-infected. Several classifiers were trained to classify the plant based on the extracted features. The choices of classifiers include support vector machine (SVM), random forest (RF), and a deep feed forward network (FFN). SVM is a classifier algorithm from the family of statistical learners commonly used for classification or regression tasks. The classification task on non-linear separable data can be performed on a higher dimensional space through kernel functions. The wide popularity of the SVM is due to its relative robustness against overfitting phenomenon. RF, another classifier used in our problem, is an ensemble method introduced in 2001 based on the bagging (bootstrap aggregating) concept for both classification and classification tasks. Its resistance against overfitting and its strength in training through different data types are important characteristics of this classifier. One of the main important extensions of RF is the optimized gradient boosting decision tree [67]. Feed forward network is one of the main categories of artificial neural network (ANN) classifiers which have been established since the 60s [68]. The ANNs are non-linear models of neurons trying to mimic the functionality of biological neurons in humans. The structure of an ANN consists of neurons in which both classification and regression tasks can be performed by flowing the features inside the trained network layers. FFNs are also commonly utilized as the front end of deep learning architectures [69]. The popularity of FFNs has increased mainly due to the ability to approximate any continuous function with high accuracy indicating their universal approximation capability as well as their training capability which enables them to approximate the Bayes discriminant [69, 70]. Besides the data quality, the performance of the network highly depends on the number of hidden layers and neurons, the choice of activation functions, the number of connections, the hyperparameters (e.g. learning rate) and the utilized optimization algorithm. Many algorithms that exist in the literature which carried out the FFN training. Such algorithms range from the first order to second order and are either the gradient-based or not which determines the required computational and/or memory complexity of each algorithm. Finally, an ensemble model

based on the combination of classifiers has been introduced and trained for the sake of classification enhancement. Three classifiers including SVM, RF and FFN have been combined through an ensemble learning method after the training as depicted in Figure 4. The performance of the classifiers has been measured in a stratified cross fold validation setting. The optimum utilized meta parameters of each classifier are selected on a greedy search basis. The performance of the hybrid model has been evaluated based on the average voting method in which all of the weights have been considered equally: let v_i denote the output for each classifier in the hybrid model, then the final prediction vote of the model is computed as follows:

$$V = \frac{1}{3}(v1 + v2 + v3)$$

It must be also pointed out that in the rare case of a tie, the label vote corresponding to the classifier with the higher validation accuracy will be selected. The description of the candidate selection algorithm is depicted as follows in Algorithm 2.

Algorithm 2 Candidate selection

Input: Extracted frames $R_{Infected} = R_1, ..., R_n$ containing the infected areas detected by Algorithm I from all recording $V_1, ..., V_n$
Output: Labels 0 for non-infected plant and 1 for infected plant

START Algorithm

1. **for** each V_i in $V_1, \ldots, V_n$ **do**
   - **Feature extraction step:**
     - **Extract statistical features from the frames set** $R_{Infected}$**:** $F = \{F_1, F_2, \ldots, F_9\}$
     - **append** V_i **label to** $F : F = \{F_1, F_2, \ldots, F_9, l_i\}$
2. **end for**
3. **Feature selection from** $F : F_{sel} = \{F_1, \ldots, F_k\}$
4. **Performing stratified CV strategy:**
   - **for** each training set T_i in $T_1, \ldots, T_K (T_1 \cup \cdots \cup T_K) = F_{sel}$: **do**
     - **Fit classifier models** $M(SVM, RF, DNN, FFN, Hybrid)$ **on** T_i **using weights** $w_1, \ldots, w_l : M_{fit} = M(T_i, w_1, \ldots, w_l)$
     - **Evaluate the prediction result of the corresponding fold on test set** $S_i : Predict_i = M_{fit}(S_i, w_1, \ldots, w_l)$
     - **Find the optimal values for the weights**
   - **end for**
5. **Aggregate and output the prediction results all classifiers:** $Accuracy, AUC - ROC, F_1, Sensitivity, Specificity, Precision = Mean(Accuracy_i, AUC - ROC_i, F_{1i}, Sensitivity_i, Specificity_i, Precision_i)$

END Algorithm

The overall detection scenario including both segmentation and candidate selection algorithms is shown in the following figure (Figure 4).

4 Approach 2: The Model for Detecting the Infected Areas of Plants Based on Pre-trained Model YOLO11

In this approach, a CNN-based pre-trained model YOLO11 is used in the object detection phase to detect the infected areas in the plants. YOLO11 is almost the latest version of Ultralytics YOLO which is a pre-trained model based on a deep learning architecture utilized for object detection tasks. YOLO11 proposes some advances on previous YOLO versions in terms of architecture and training procedure which enables a wider range of object detection jobs. The YOLO11 is initially used as a pre-trained model and further fine-tuned to be adapted to our problem use-case. In fact, the YOLO11 is used for object detection in frames of videos and later on a decision based on a defined threshold is used to label the videos as infected or healthy. General pipeline of the proposed approach is shown as follows (Figure 5).

The YOLO11 has been trained using 2400 frames without any augmentation from original frames. The fine-tuned model has been further utilized for classifying the recorded videos into infected or uninfected.

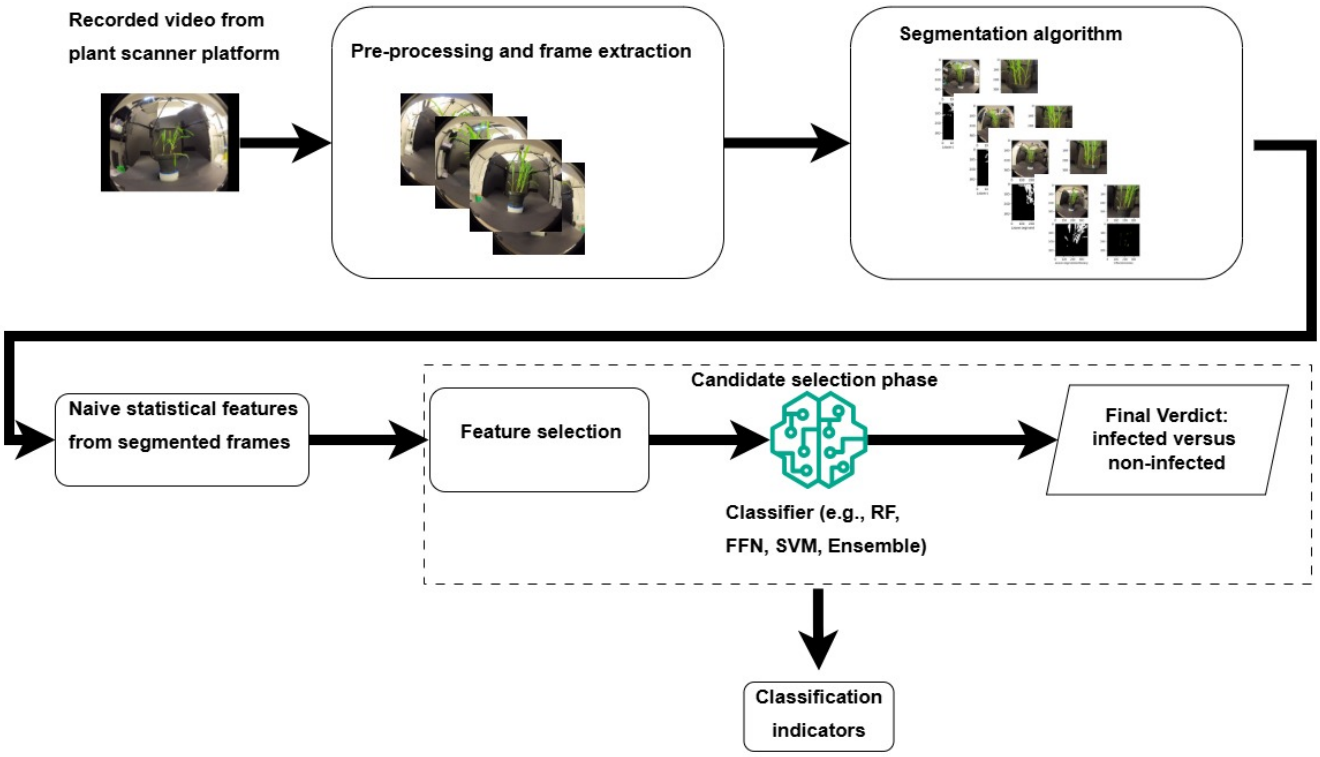


Figure 4. The complete pipeline of the platform (first approach)

5 Results

5.1 The Final Plant Scanning Platform

Plants are sensitive subjects for scanning. Minor movements or vibrations cause the flexible plant material to move and, therefore, reduce the recording quality. To generate high-quality video material and consistent test environments, we built a scanning platform that allows 360° rotation of cameras around the plant (Figure 6). The final Plantscanner system with 150 L hexagon scan volume contains light-hole aluminum round rods with a diameter of 6 mm for the struts. The connections between the rods and the camera mounts were printed with NYLON 12 POWDER using an SLS 3D printing process (Fuse 1, Formlabs GmbH, Berlin, Germany). Further final components are (Figure 6):

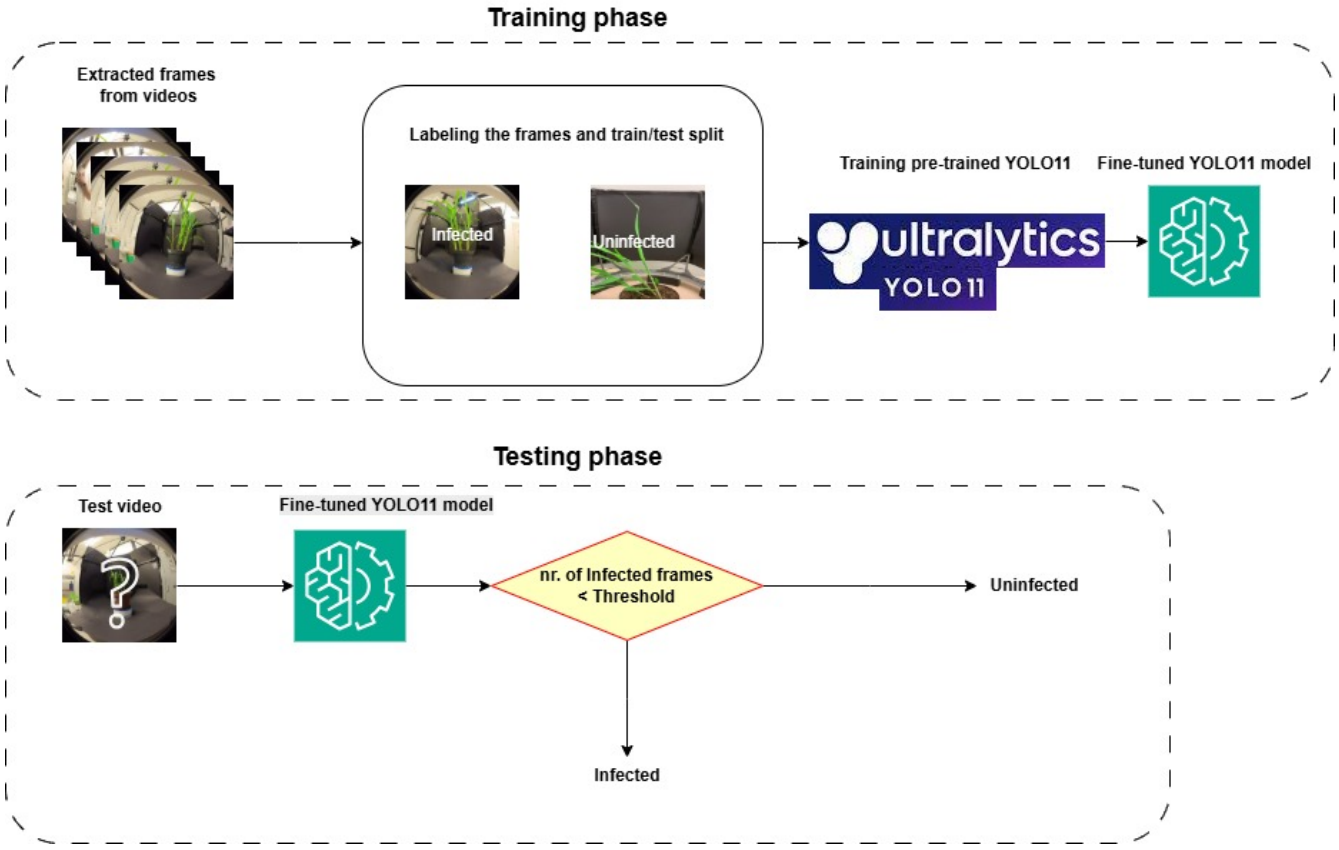


Figure 5. The complete pipeline of the proposed platform

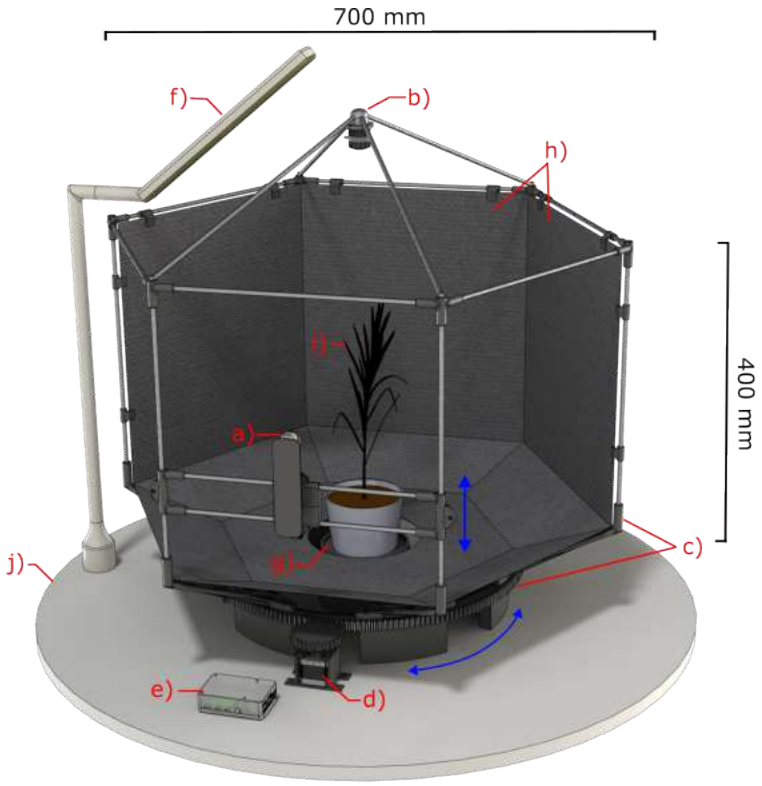


Figure 6. Phase Four – Final Plantscanner for 360° image capturing of the infected plant with 150 L scan volume: a) side-view camera (Insta360 Go 2), b) top-view camera (Raspberry Pi camera), c) 360° camera rotation unit, d) stepper motor, e) control unit, f) lamp, g) adjustable plant platform, h) high-contrast background, i) scan object, j) base plate

a) Insta360 Go 2 side-view camera (GING2XX/A, Arashi Vision Inc., Shenzhen, China) with integrated wide-angle lens and iPhone XR (MRY92ZD/A, Apple Inc., California, U.S.) with an adjustable camera mount in height and angle

b) Raspberry Pi top-view camera with stacked 12.3 megapixel Sony sensor (IMX477R, Cambridge, Great Britain) and a mounted Raspberry Pi 6 mm wide-angle lens (PT361060M3MP12)

c) camera rotation unit based on a slewing ring driven by a stepper motor (d) for vibration-free 360° camera movement around the scanned object which does not experience any movement between the series of images

d) Polulu Nema 17 size bipolar stepper motor with 200 steps per revolution

e) control unit with Arduino Nano microcontroller (Arduino SA, Chiasso, Switzerland) and DRV8824 stepper motor controller (Texas Instruments, Dallas, Texas) for controlling rotational movement speed and Raspberry Pi 4 (Model B 2GB RAM 2018, Cambridge, Great Britain) for image capturing

f) 14 W LED light source with adjustable light intensity to ambient conditions with light color settings 3000K, 4500K, 6000K (YTZZ6SSTQGGMDN9QPM5Y-G, SerDa-Run)

g) adjustable plant platform height to exclude plant pot from scan volume

h) non-reflecting and high-contrast black background (replaceable)

In the designing of our low-cost scanner, we focused on using only conventional, easily accessible materials. Compared to other studies [17, 18, 19, 25, 30], the low-cost factor makes it possible to replicate and establish the scanner in smaller studies. In addition, the handling of the scanner is intended for anyone without in-depth training. The base plate is a conventional wooden plate that provides a secure stand and makes it easy to mount additional features, such as additional light sources. We used light-hole aluminum rods as the frame material, which are well-suited for this purpose due to their strength and lightness. All connectors and the brackets for the rotation unit are made of SLS 3D printed material like nylon. Waterproof materials such as ABS or PETG are also possible with FDM 3D printing. These parts, in combination with a ball-bearing slewing ring and a stepper motor as the drive, enable us to rotate the camera 360° with low vibration at a speed of 12 s/revolution. These settings made it possible to reduce image blurriness to a minimum. To create a sharp, high-resolution image with a large field of view, we used a wide-angle lens with a high FPS and 2k resolution setting. The camera itself can be adjusted in height and angle to the scan object. The LED lamp that we use avoids shadows caused by overlapping plant leaves and overexposure, which would result in unrecognizable image information. In addition, it enables consistent illumination and, therefore, consistent color intensity for the further algorithm. The currently used 45° offset to the lens can be freely adjusted. The non-reflecting and high-contrast black background was designed from cost-efficient cardboard paper. The black background ensures that only the plant is considered in the later analysis. The scanner system created in this manner therefore, allows for cost-effective and rapid capture of smaller plants in a uniform and consistent test environment.

5.2 Generated Database

During the further development of the scanning platform and the algorithm, insight was gained that led to improvements in the recording as well as the detection process. As a result, in the four development phases, a total of 2090 pictures and 2238 videos were generated. In phase one, pictures of plants were taken with private mobile phones. The aim was to visualize infected areas clearly. To generate defined parameters for recording, an iPhone 10 was used as a camera in Phase 2, as shown in Figure 2. For Phase 3, a more vibration-resistant camera was used. To deal with bigger plants, the camera was moved further away from the scanned object, and an external light was installed. In Phase 4, the last background optimization was done. Samples of the generated data are shown in Figure 7. Also, the summary of data generation information is presented in Table 1 below:

Table 1. Generated picture/video material during development phases

Phase	camera system	camera position	material gained	plant condition	
				healthy	infected
One	Cell phones from different manufacturers and of different quality	random	2090 pictures	random collection of healthy and infected plants	
Two	iPhone 10	sideview	88 videos	39	49
Three	ActionCam	sideview	1893 videos	200	1693
Four	ActionCam	sideview	157 videos	79	78
	RaspiCam	topview	100 videos	50	50

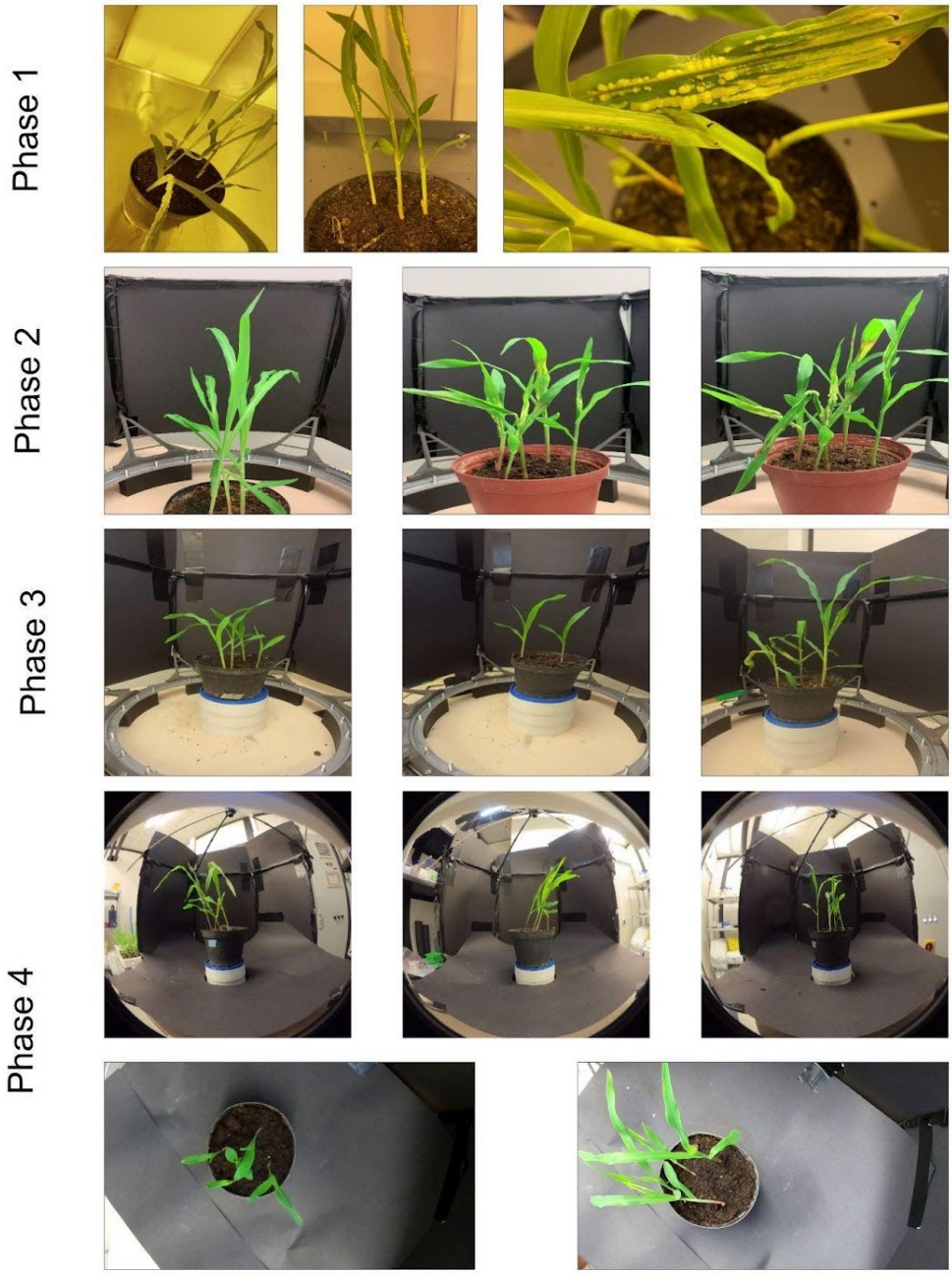


Figure 7. Example of recordings during the development phases: Images shown are sample images of maize plants from the four stages of development. The technical details are described in Section 2.

5.3 Performance Measurement

To evaluate the classification models presented in this work, the elements of the extracted confusion matrix including true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values will be considered. TP indicates the truly validated infected plants correctly predicted by the prediction method, and TN represents the healthy plants correctly predicted.

FP denotes the healthy plants incorrectly predicted as the infected ones and finally, FN indicates the experimentally validated infected plants incorrectly predicted as the healthy plants. The classification performance is evaluated by accuracy (ACC), sensitivity (SN), specificity (SP), and precision (PR) indicators. Furthermore, the receiver operating characteristic (ROC) curve representing the trade-off between the TP rate or sensitivity and the FP rate or 1-specificity has been extracted. The definition of the aforementioned classification parameters is as follows:

Accuracy: It is the percentage of the correct predictions calculated below.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

Sensitivity (Recall): It indicates the percentage of infected plants sites that have been predicted correctly.

$$Sensitivity = \frac{TP}{TP+FN} \tag{2}$$

Specificity: It shows the percentage of non-infected (healthy) sites that have been predicted correctly as non-infected.

$$Specificity = \frac{TN}{TN+FP} \tag{3}$$

Precision: It indicates the positive predictive value and is calculated as follows.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

F1-score: The F1-score calculates a score based on the combination of precision and recall by computing the harmonic mean of those metrics as follows.

$$F1 = 2\frac{Precision.Recall}{Precision+Recall} \tag{5}$$

5.4 Models Evaluation

In this section the experimental results corresponding to two introduced approaches are presented. In the first approach, the algorithms in the classification models have been trained by handcrafted features extracted from the train dataset and the models have been frozen by evaluation using k-fold cross-validation setting for k=3,5,10. The evaluation on the test set has been presented by calculating the mean of all classification parameters achieved per each fold. Although there are nine statistical features (mean, standard of deviation, min, max, the first quartile, the third quartile, median, mode and the number of frames whose infected ratio is bigger than the mean of infected ratios of all frames) have been calculated from the segmentation algorithm in the first phase, a subset of the features is used for the candidate selection phase. All of the features have been calculated from the pixel values of the infected areas of the extracted frames. For example, to calculate the first feature i.e., mean for one recording, the ratio of the infected area of each extracted frame to the whole area of the plant is calculated and then the mean is taken over all of these ratios. The same process is performed for other features as well. To select the best features, a feature selection algorithm is used. According to the feature selection algorithm, the following features have been found to present the best classification performance among other combinations of the features:

1. Max
2. The first quartile
3. Mode
4. Number of frames whose infected ratio is bigger than the mean of infected ratios of all frames

The corresponding FFN model consists of 3 layers and the parameters are shown in Table 2 as follows. According to the evaluation of the trained models, in the first approach, the best performance is achieved by the SVM model in a 10-fold stratified CV training scenario.

Table 2. Settings of the proposed deep FFN architecture

Sequential neurons	(64, 32, 1)
Activation function	(ReLU, ReLU, Sigmoid)
Optimizing function	adam
Loss function	binary cross entropy
Epochs	100
Batch size	16

The achieved accuracy is 0.90 with the maximum accuracy of 1.00 in one fold. To experimentally evaluate the second approach, at first the YOLO11 pretrained model has been trained using frames extracted from the training dataset. To reduce the training cost, we have extracted just 10% of the frames of each video of the set. Fine-tuning included adjusting learning rates (0.01), batch size (32), and epochs (50), with no augmentation to preserve biological relevance. The model has been trained to distinguish the infected frames from non-infected frames in real-time. The fine-tuned model based on this frames set is then utilized to label the extracted. The model contains 28.358.626 learning parameters corresponding to the same amount of gradient values through 176 layers with a learning rate of 0.01. The image frames are of size 224 x 224 pixels. Each epoch iteration takes about 1.1 seconds on average on a Tesla T4 GPU platform using 57 MB in an optimized weights setting. The final model shows an excellent accuracy AUC-ROC of 0.9998 validating on a test set of size 30 videos. Summary of the results corresponding to the first approach and the first phase of the second approach are presented in Table 3 as follows. It is worth mentioning that the results corresponding to the best fold with the highest accuracy for each classifier is depicted in the table.

Table 3. Classification parameters achieved through cross-fold validation for $k = 3, 5, 10$ using four different classifiers

Model	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUC-ROC
SVM-k=10	0.90	0.85	0.95	0.85	0.73	0.90
FFN-k=3	0.78	0.73	0.83	0.83	0.77	0.78
RF-k=3	0.76	0.73	0.78	0.89	0.73	0.77
Hybrid-k=3	0.83	0.73	0.93	0.93	0.79	0.85
Approach 2	**0.99**	**0.99**	**0.99**	**0.98**	**0.99**	**0.99**

According to the results shown in the Table 3, the second presented approach shows an excellent performance with accuracy 0.99 (top-1 accuracy 1.00). The ROCs and corresponding AUC are shown individually in Figure 8 as follows. It is worth mentioning that the extracted ROCs correspond to the value of K (in the CV setting) wherein the highest accuracy is achieved. Therefore, they are presented per each plot independently.

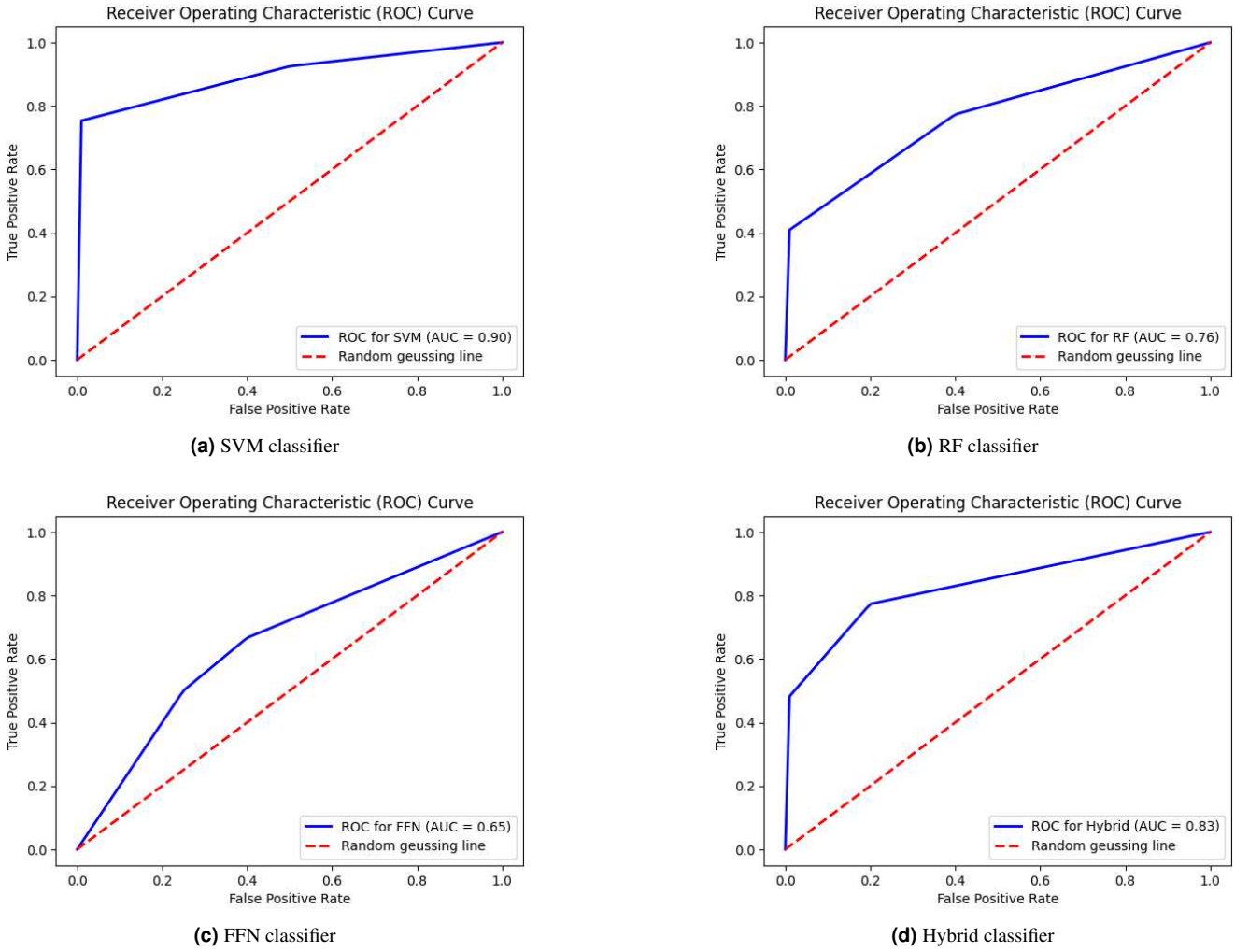
Intermediate results corresponding to both approach models are shown in Figure 9 and Figure 10 as follows.

Also, the results corresponding to the train/loss trend as well as achieved accuracy in the second approach is shown in Figure 11 as follows.

Furthermore, to provide a concrete evidence highlighting the accuracy and discriminating performance of the proposed model, a comparison between the achieved results and similar models are made. Table 4 compares the results with former YOLO model (YOLO8) and ViT [71] model in classifying infected versus healthy frames.

Table 4. Classification parameters achieved through cross-fold validation for $k = 3, 5, 10$ using four different classifiers

Model	Accuracy	Sensitivity	Specificity	Precision	F1-score	AUC-ROC
YOLO8	0.88	0.93	0.71	0.92	0.92	0.93
ViT	0.78	0.82	0.77	0.66	0.73	0.93



(a) SVM classifier

(b) RF classifier

(c) FFN classifier

(d) Hybrid classifier

Figure 8. ROC of four different classifiers corresponding to the best fold

5.5 Implementation Details

The implementation of the proposed approaches has been done with Python version 3.12 leveraging its robust ecosystem of libraries tailored for machine learning and computer vision tasks. The primary frameworks used include Pandas, Numpy, Scikitlearn, tensorflow, and OpenCV. For implementing the second approach, package ultralytics has been imported in order to automate the whole pipeline.

6 Discussion

This study presents a platform for detecting tumor-based symptoms in corn caused by the pathogenic fungus *Ustilago maydis* under laboratory conditions. The *Ustilago maydis-Zea mays* pathosystem is an attractive system to study pathogens [72, 73, 74]. Much of our understanding of *U. maydis* infection behavior comes from genetic manipulations of the fungus, and a comparison of resulting mutants to wildtype strains in infection assays. While the symptoms in the wild differ greatly from the symptoms developed in this infection assay (see Figure 1), differences in symptoms achieved during this assay, provide insights into the gene's potential role in the fungus. Homology studies further allow extrapolation of these findings to other fungal pathogens, helping to identify common mechanisms of infection. This knowledge can contribute to the development of targeted strategies to mitigate pathogen infestations in crops through precise treatments. The problem with these infection assays is that the symptoms are read out manually by the respective scientists, which can lead to subjective biases. Furthermore, in order to make the workload manageable, publications often only show a few pictures of the symptoms of hundreds of plants, which is accepted in the community as standard [72, 73, 74]. To solve these problems, we have generated a platform and an algorithm that visualizes the infected plants and can then distinguish between infected and non-infected plants. The developed scanning

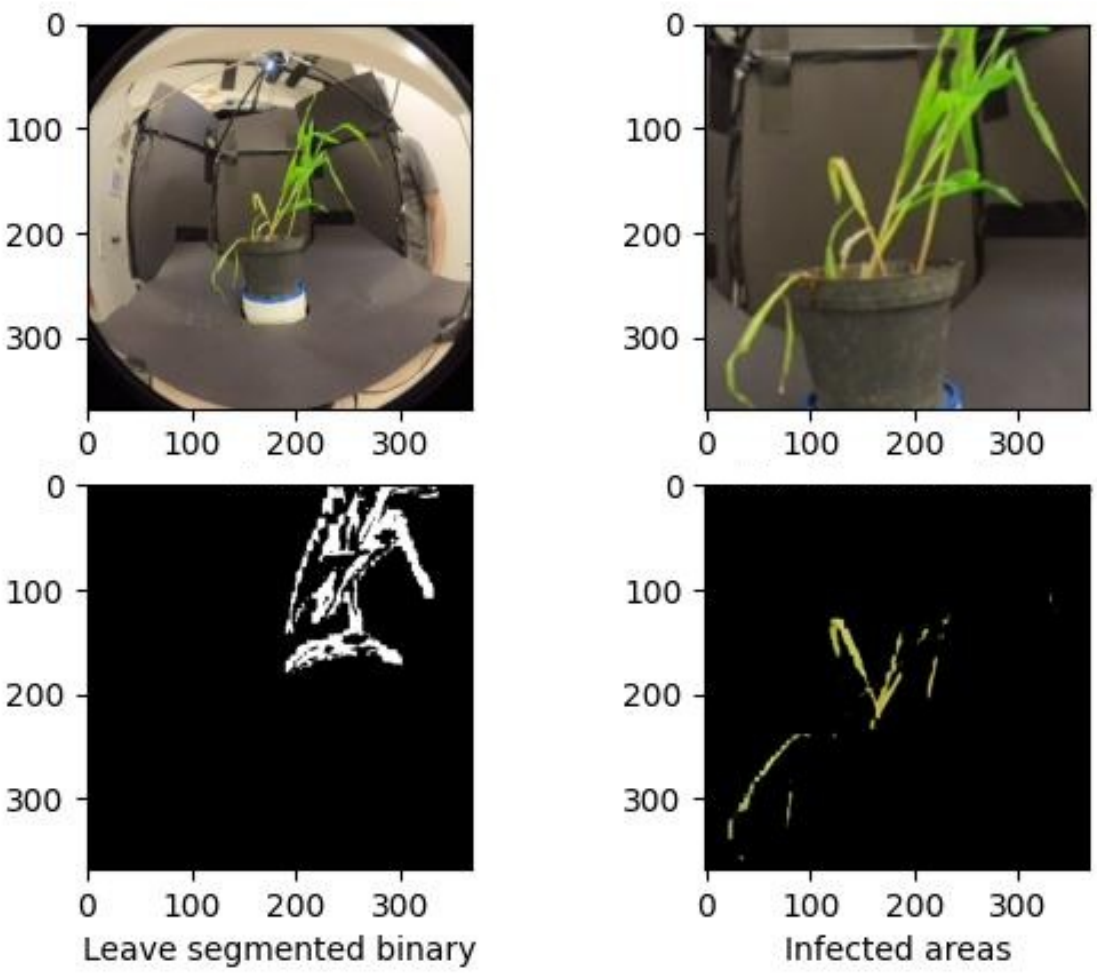


Figure 9. Infected areas by the first approach based on computer vision

platform was used to generate video material of the plants. The aim was to generate a modular platform that can be used by any trained person in laboratory environments, greenhouses, or other premises due to its high adaptability. The open-source platform allows for modifications to the camera systems, orientation, rotation speed, backgrounds, and lighting, making it highly customizable. To its flexible camera settings, users can scan a large number of plants in an adequate amount of time. To achieve this goal, we have placed great emphasis on designing the platform to be cost-efficient. Unlike commercial systems (e.g., WIWAM, €20,000), our open-design scanner reduces costs by 95%, enabling wider adoption. With a production value of less than 1000€ (inclusive camera, that made the biggest position), the device opens the possibility for stable plant recordings to a wide range of users with different needs. Because existing infection detection systems do not apply to tumor-based plant pathogens, we have created our own data collection with video and images. During the different development phases we ended up with more than 2000 pictures and 2000 videos of infected and non-infected maize plants. This database was created under controlled laboratory conditions, providing video/image data that accurately represents infection stages in maize plants. To clearly observe the stages, the plants were monitored for a duration of up to 9 days, during which the infection became more visible. In the future, this database may help to develop further approaches for detecting fungal infections on plants. Using an RGB imaging system combined with the self-made scanning device, we were able to achieve a classification accuracy of over 0.99. This point will clearly discriminate between the infected plants from the non-infected ones. In Figure 12, one of the selected features (number of frames whose infected ratio is bigger than the mean of infected ratios of all frames) has been plotted for two samples of recordings corresponding to the infected and non-infected plants.

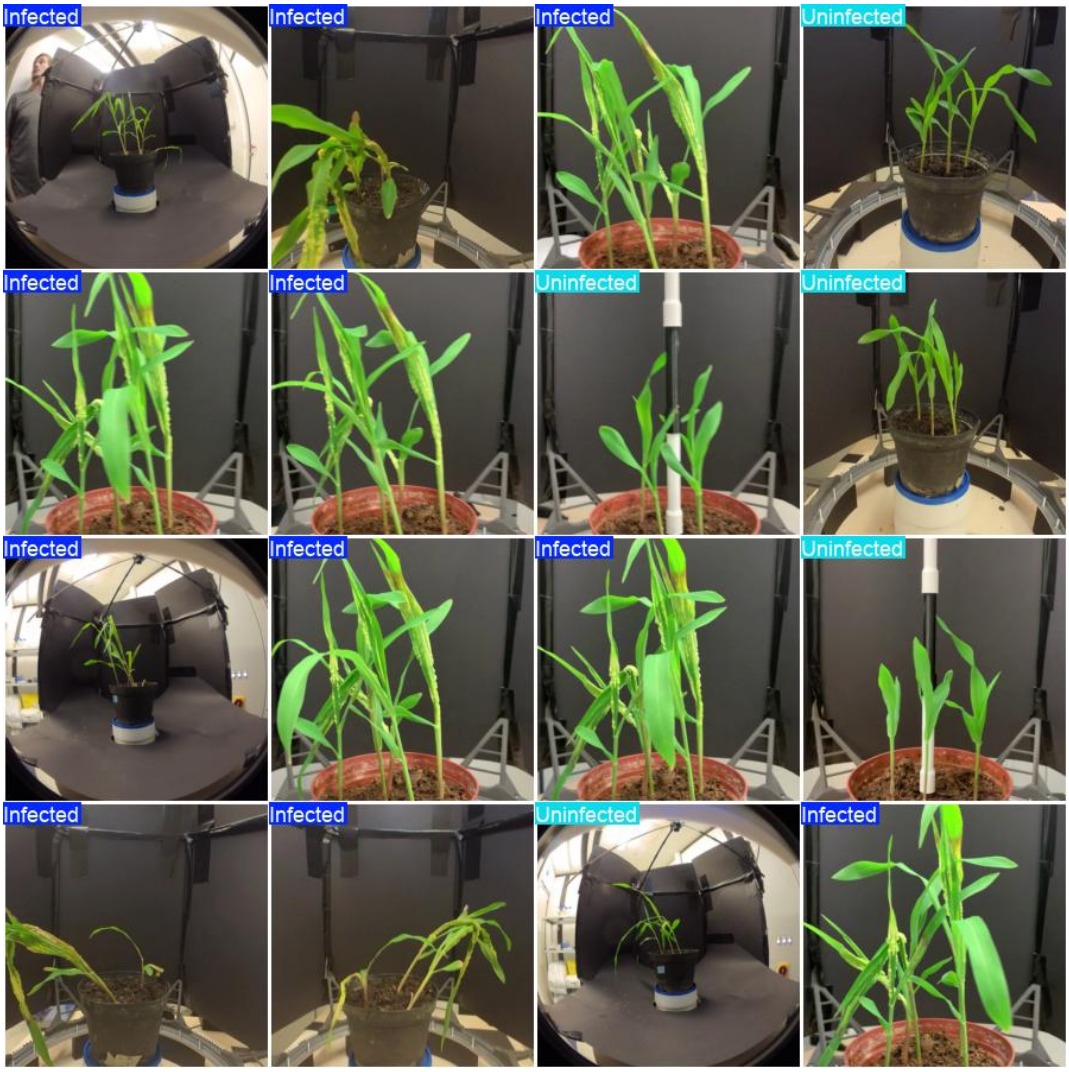


Figure 10. Detection results output from the second approach based on transfer learning

As depicted in the above plot (Figure 12), the number of frames whose infected ratio is bigger than the threshold corresponding to an infected plant in the 1, 3, 5, 7, and 9 days after infection is increasing, while the similar feature corresponding to a non- infected one has an opposite trend as expected. Similar plots might be expected when extracted for the other features, however it is not a must. This is an indication of the synergy between the segmentation algorithm and the infection process, showing an empirical justification of the consistency of the algorithm. In summary, our system provides a strong basis for further work on scanning systems on plants. The device we have generated can help to develop other devices that can also be used to scan other types of plants. Furthermore, the complete package developed by us offers the possibility to easily share video and image material of maize plants infected with *U. maydis* among different scientists. It is possible to save the obtained image and video material and to review it long after the experiment was conducted.

7 Conclusions

In this paper, we present a platform for detecting tumor-based symptoms in maize leaves caused by pathogenic fungi. The prototype integrates various camera systems to capture comprehensive recordings of the plant. These videos are processed by an infection detection algorithm, which operates in two consecutive phases to identify infected plants. The platform achieves an accuracy of approximately 0.99, with an AUC-ROC of 0.99 utilizing a presented approach based on the pre-trained CNN-based model YOLO11. In addition to enhancing the detection algorithm, the platform generates a versatile dataset using different hardware configurations, providing a valuable resource for researchers. The platform is highly adaptable, cost-efficient, and capable of integrating multiple camera systems, making it suitable for diverse plant species and environments. Future

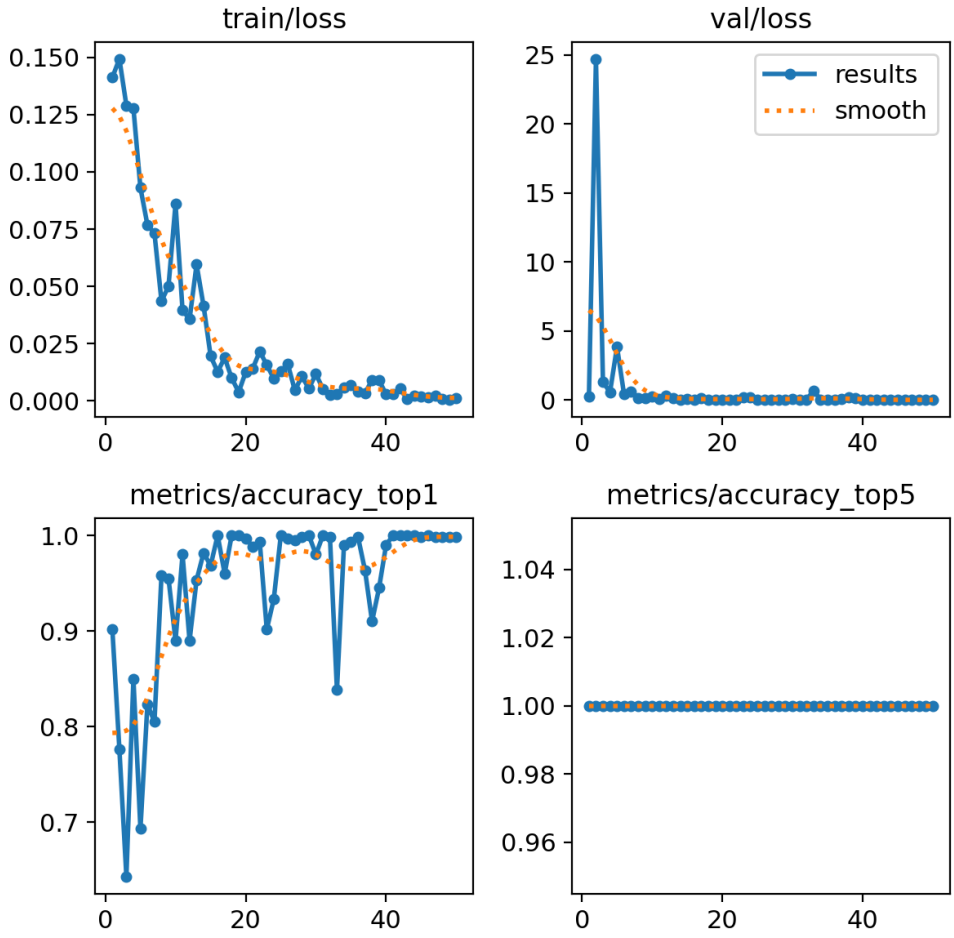


Figure 11. Loss/Accuracy trend of the second approach based on transfer: x-axis denotes the epochs and y-axis (above) denotes the loss value and y-axis (below) denotes the accuracy value

developments include embedding the detection algorithm directly into the camera system for industrial applications, creating a mobile app version, and advancing the algorithm to assess infection stages. To address the limitations of the current work, future work will test the platform in field conditions to assess robustness against environmental variability as well. Plans also involve incorporating cutting-edge techniques like Vision Transformers [75] to further improve detection accuracy and performance.

Data Availability

The generated database are provided as some frame samples in https://github.com/aTabaIMI/Plant-Scanner-THM-Marburg-MaizeLeaf with full data availability upon request.

Authors Contributions

Conceptualization was carried out by M.C., S.A.H.T., N.O., M.L., G.B., and K.S. Data curation was performed by M.C., S.A.H.T., N.O., O.S., I.E., P.K., C.T., M.L., and N.F. Formal analysis was conducted by M.C., S.A.H.T., N.O., and I.E. Investigation involved M.C., S.A.H.T., N.O., I.E., P.K., C.T., M.L., and N.F. Methodology was developed by M.C., S.A.H.T., O.S., and K.S. Project administration was managed by G.B. and K.S. Resources were provided by M.C., O.S., P.K., C.T., and M.L. Software development was carried out by S.A.H.T., N.O., and N.F. Supervision was provided by G.B. and K.S. Validation was performed by S.A.H.T., N.O., I.E., P.K., C.T., and N.F. Visualization was prepared by O.S. and C.T. The original draft of the manuscript was written by M.C., S.A.H.T., N.O., I.E., P.K., C.T., M.L., N.F., G.B., and K.S. Review and editing of the

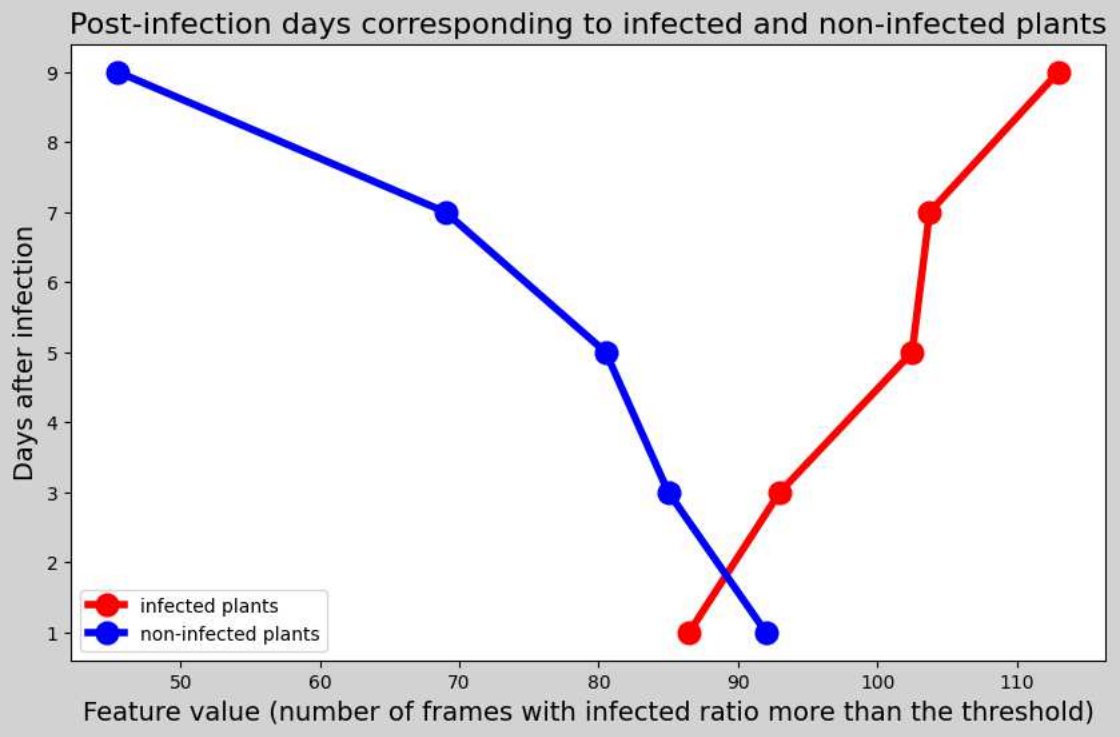


Figure 12. Comparison of detected frames with the ratio of the detected pixels bigger than the specified threshold value between the infected and healthy plants

manuscript were carried out by M.C., S.A.H.T., N.O., O.S., I.E., P.K., C.T., M.L., N.F., G.B., and K.S. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors would like to acknowledge Abolfazl Keshavars and Erfan Tarehkar from Department of Computer Science at University of Guilan for their great support in implementing and validating the main detection model pipeline proposed in this manuscript as the second approach. The script is accessed through:
https://github.com/abolfazlkeshavarz/Classification-of-plant-infection.

Conflict of Interests

Authors declare no conflicts of interest.

Funding

This research received no external funding.

References

1. United nations. (2019) 2019 revision of world population prospects. https://population.un.org/wpp/. [Accessed 30-07-2024].

2. Food and agriculture organization of the united nations (fao) (2019). https://www.fao.org/news/story/en/item/1187738/icode/ (2019). [Accessed 30-07-2024].

3. Bebber, D. P., Ramotowski, M. A. & Gurr, S. J. Crop pests and pathogens move polewards in a warming world. *Nat. climate change* **3**, 985–988 (2013).

4. Dean, R. *et al.* The top 10 fungal pathogens in molecular plant pathology. *Mol. plant pathology* **13**, 414–430 (2012).

5. Kämper, J. *et al.* Insights from the genome of the biotrophic fungal plant pathogen ustilago maydis. *Nature* **444**, 97–101 (2006).

6. Wege, S.-M. *et al.* Versatile CRISPR/Cas9 systems for genome editing in ustilago maydis. *J. Fungi (Basel)* **7**, 149 (2021).

7. Schuster, M., Schweizer, G., Reissmann, S. & Kahmann, R. Genome editing in ustilago maydis using the CRISPR-Cas system. *Fungal Genet. Biol.* **89**, 3–9 (2016).

8. Brachmann, A., König, J., Julius, C. & Feldbrügge, M. A reverse genetic approach for generating gene replacement mutants in ustilago maydis. *Mol. Genet. Genomics* **272**, 216–226 (2004).

9. Christensen, J. J. *Corn smut caused by Ustilago maydis /* (American Phytopathological Society,, [Worcester, Mass.] :, 1963).

10. Djamei, A. *et al.* Metabolic priming by a secreted fungal effector. *Nature* **478**, 395–398 (2011).

11. Tanaka, S. *et al.* A secreted ustilago maydis effector promotes virulence by targeting anthocyanin biosynthesis in maize. *Elife* **3**, e01355 (2014).

12. Khan, M. *et al.* Tip of the iceberg? three novel TOPLESS-interacting effectors of the gall-inducing fungus ustilago maydis. *New Phytol.* **244**, 949–961 (2024).

13. van der Linde, K. *et al.* A maize cystatin suppresses host immunity by inhibiting apoplastic cysteine proteases. *Plant Cell* **24**, 1285–1300 (2012).

14. Han, X. *et al.* A kiwellin disarms the metabolic activity of a secreted fungal virulence factor. *Nature* **565**, 650–653 (2019).

15. Ludwig, N. *et al.* A cell surface-exposed protein complex with an essential virulence function in ustilago maydis. *Nat. Microbiol.* **6**, 722–730 (2021).

16. Schuster, M. *et al.* Novel secreted effectors conserved among smut fungi contribute to the virulence of ustilago maydis. *Mol. Plant. Microbe. Interact.* **37**, 250–263 (2024).

17. Singh, V., Sharma, N. & Singh, S. A review of imaging techniques for plant disease detection. *Artif. Intell. Agric.* **4**, 229–242, DOI: 10.1016/j.aiia.2020.10.002 (2020).

18. Mahlein, A.-K. Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping. *Plant Dis.* **100**, 241–251, DOI: 10.1094/PDIS-03-15-0340-FE (2016). Publisher: Scientific Societies.

19. Sankaran, S., Mishra, A., Ehsani, R. & Davis, C. A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* **72**, 1–13, DOI: 10.1016/j.compag.2010.02.007 (2010).

20. Hassan, S. M. *et al.* A survey on different plant diseases detection using machine learning techniques. *Electronics* **11**, DOI: 10.3390/electronics11172641 (2022).

21. Abade, A., Ferreira, P. A. & de Barros Vidal, F. Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* **185**, 106125, DOI: 10.1016/j.compag.2021.106125 (2021).

22. Bölker, M. Ustilago maydis–a valuable model system for the study of fungal dimorphism and virulence. *Microbiology* **147**, 1395–1401 (2001).

23. Nagasubramanian, K. *et al.* Plant disease identification using explainable 3D deep learning on hyperspectral images. *Plant Methods* **15**, 98, DOI: 10.1186/s13007-019-0479-8 (2019).

24. Nguyen, C. V. *et al.* 3D Scanning System for Automatic High-Resolution Plant Phenotyping. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1–8, DOI: 10.1109/DICTA.2016.7796984 (2016).

25. Schor, N. *et al.* Development of a robotic detection system for greenhouse pepper plant diseases. *Precis. Agric.* **18**, 394–409, DOI: 10.1007/s11119-017-9503-z (2017).

26. Mohanty, S. P., Hughes, D. P. & Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **7**, DOI: 10.3389/fpls.2016.01419 (2016). Publisher: Frontiers.

27. Pethybridge, S. J. & Nelson, S. C. Leaf Doctor: A New Portable Application for Quantifying Plant Disease Severity. *Plant Dis.* **99**, 1310–1316, DOI: 10.1094/PDIS-03-15-0319-RE (2015). Publisher: Scientific Societies.

28. Home - WIWAM — wiwam.com. https://www.wiwam.com/. [Accessed 12-02-2025].

29. Plant Phenotyping | Photon Systems Instruments — plantphenotyping.com. https://plantphenotyping.com/. [Accessed 12-02-2025].

30. Neupane, K. & Baysal-Gurel, F. Automatic Identification and Monitoring of Plant Diseases Using Unmanned Aerial Vehicles: A Review. *Remote. Sens.* **13**, 3841, DOI: 10.3390/rs13193841 (2021). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute.

31. Baratov, R. & Valixanova, H. Smart system for early detection of agricultural plant diseases in the vegetation period. *E3S Web Conf.* **386**, 01007, DOI: 10.1051/e3sconf/202338601007 (2023).

32. Vigneau, N., Ecarnot, M., Rabatel, G. & Roumet, P. Potential of field hyperspectral imaging as a non destructive method to assess leaf nitrogen content in Wheat. *Field Crop. Res.* **122**, 25–31, DOI: 10.1016/j.fcr.2011.02.003 (2011).

33. de Luna, R. G., Dadios, E. P. & Bandala, A. A. Automated Image Capturing System for Deep Learning-based Tomato Plant Leaf Disease Detection and Recognition. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, 1414–1419, DOI: 10.1109/TENCON.2018.8650088 (2018). ISSN: 2159-3450.

34. Liu, J. & Wang, X. Plant diseases and pests detection based on deep learning: a review. *Plant Methods* **17**, DOI: 10.1186/s13007-021-00722-9 (2021).

35. Thapa, R., Zhang, K., Snavely, N., Belongie, S. & Khan, A. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Appl. Plant Sci.* **8**, DOI: 10.1002/aps3.11390 (2020).

36. Tsaftaris, S., Minervini, M. & Scharr, H. Machine learning for plant phenotyping needs image processing. *Trends Plant Sci.* **21**, 989–991, DOI: 10.1016/j.tplants.2016.10.002 (2016).

37. Fuentes, A., Yoon, S. & Park, D. *Deep Learning-Based Techniques for Plant Diseases Recognition in Real-Field Scenarios*, 3–14 (2020).

38. Singh, D. S. & Singh, M. Automated blast disease detection from paddy plant leaf — a color slicing approach. 339–344, DOI: 10.1109/ICITM.2018.8333972 (2018).

39. El Sghair, M., Jovanovic, R. & Tuba, M. An algorithm for plant diseases detection based on color features. https://mail.ipb.ac.rs/~rakaj/home/014-0001(2017).pdf (2017). [Accessed 31-07-2024].

40. Pugoy, R. A. & Mariano, V. Automated rice leaf disease detection using color image analysis. *Proc. SPIE - The Int. Soc. for Opt. Eng.* **8009**, DOI: 10.1117/12.896494 (2011).

41. Shrivastava, V. K. & Pradhan, M. K. Rice plant disease classification using color features: a machine learning paradigm. *J. Plant Pathol.* **103**, 17 – 26 (2020).

42. Husin, Z., md shakaff, a. y., Aziz, A. & Farook, R. Feasibility study on plant chili disease detection using image processing techniques. *Proc. - 3rd Int. Conf. on Intell. Syst. Model. Simulation, ISMS 2012* 291–296, DOI: 10.1109/ISMS.2012.33 (2012).

43. Majid, K., Herdiyeni, Y. & Rauf, A. I-pedia: Mobile application for paddy disease identification using fuzzy entropy and probabilistic neural network. *2013 Int. Conf. on Adv. Comput. Sci. Inf. Syst. ICACSIS 2013* 403–406, DOI: 10.1109/ICACSIS.2013.6761609 (2013).

44. Dey, A., Sharma, M. & Meshram, D. Image processing based leaf rot disease, detection of betel vine (piper betlel.). *Procedia Comput. Sci.* **85**, 748–754, DOI: 10.1016/j.procs.2016.05.262 (2016).

45. Yao, Q. *et al.* Application of support vector machine for detecting rice diseases using shape and color texture features. *2009 Int. Conf. on Eng. Comput.* 79–83 (2009).

46. Prajapati, H., Shah, J. & Dabhi, V. Detection and classification of rice plant diseases. *Intell. Decis. Technol.* **11**, 357–373, DOI: 10.3233/IDT-170301 (2017).

47. Prasad, S., Peddoju, S. K. & Ghosh, D. Multi-resolution mobile vision system for plant leaf disease diagnosis. *Signal, Image Video Process.* **10**, DOI: 10.1007/s11760-015-0751-y (2015).

48. Singh, V. & Misra, A. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **4**, 41–49, DOI: https://doi.org/10.1016/j.inpa.2016.10.005 (2017).

49. Zhang, S., Zhu, Y., You, Z. & Wu, X. Fusion of superpixel, expectation maximization and PHOG for recognizing cucumber diseases. *Comput. Electron. Agric.* **140**, 338–347 (2017).

50. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788, DOI: 10.1109/CVPR.2016.91 (2016).

51. Soeb, M. J. A. *et al.* Tea leaf disease detection and identification based on YOLOv7(YOLO-T). *Sci. Rep.* **13**, 6078 (2023).

52. Liu, Y., Liu, Y., Guo, X., Ling, X. & Geng, Q. Metal surface defect detection using SLF-YOLO enhanced YOLOv8 model. *Sci. Rep.* **15**, 11105 (2025).

53. Yang, Z. *et al.* Stomayolo: A lightweight maize phenotypic stomatal cell detector based on multi-task training. *Plants* **14**, DOI: 10.3390/plants14132070 (2025).

54. Sarkar, C., Gupta, D., Gupta, U. & Hazarika, B. B. Leaf disease detection using machine learning and deep learning: Review and challenges. *Appl. Soft Comput.* **145**, 110534, DOI: https://doi.org/10.1016/j.asoc.2023.110534 (2023).

55. Ahmad, A., Saraswat, D. & Gamal, A. A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agric. Technol.* **3**, 100083, DOI: 10.1016/j.atech.2022.100083 (2022).

56. Upadhyay, A. *et al.* Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artif. Intell. Rev.* **58**, 1–64 (2025).

57. Chug, A., Bhatia, A., Singh, A. & Singh, D. A novel framework for image-based plant disease detection using hybrid deep learning approach. *Soft Comput.* **27**, DOI: 10.1007/s00500-022-07177-7 (2022).

58. Panigrahi, K. P., Das, H., Sahoo, A. K. & Moharana, S. C. *Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms*, 659–669 (Springer Singapore, 2020).

59. Khan, F. *et al.* A mobile-based system for maize plant leaf disease detection and classification using deep learning. *Front. Plant Sci.* **14**, DOI: 10.3389/fpls.2023.1079366 (2023).

60. Parez, S., Dilshad, N., Alghamdi, N. S., Alanazi, T. M. & Lee, J. W. Visual intelligence in precision agriculture: Exploring plant disease detection via efficient vision transformers. *Sensors* **23**, 6949 (2023).

61. De Silva, M. & Brown, D. Plant disease detection using vision transformers on multispectral natural environment images. In *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 1–6 (IEEE, 2023).

62. Prashanthi, B., Krishna, A. P. & Rao, C. M. Levit-leaf disease identification and classification using an enhanced vision transformers (vit) model. *Multimed. Tools Appl.* 1–32 (2024).

63. Li, L.-H. & Tanone, R. Ensemble learning based on cnn and transformer models for leaf diseases classification. In *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1–6 (IEEE, 2024).

64. Yang, B. *et al.* A novel plant type, leaf disease and severity identification framework using cnn and transformer with multi-label method. *Sci. Reports* **14**, 11664 (2024).

65. Dai, G. *et al.* Lightweight vision transformer with lite-avpso hyperparameter optimization for agricultural disease recognition. *IEEE Internet Things J.* (2025).

66. Hughes, D., Salathé, M. *et al.* An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060* (2015).

67. Zhang, Y. *et al.* Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Briefings Bioinforma.* **20**, 2185–2199, DOI: 10.1093/bib/bby079 (2018).

68. Verhelst, M. & Moons, B. *IEEE Solid-State Circuits Mag.* **9**, 55–65, DOI: 10.1109/MSSC.2017.2745818 (2017).

69. Rane, C., Tyagi, K., Malalur, S., Shinge, Y. & Manry, M. Optimal input gain: All you need to supercharge a feed-forward neural network (2023). 2303.17732.

70. Ruck, D., Rogers, S., Kabrisky, M., Oxley, M. & Suter, B. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE transactions on neural networks / a publication IEEE Neural Networks Counc.* **1**, 296–8, DOI: 10.1109/72.80266 (1990).

71. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale (2021). 2010.11929.

72. Ludwig, N. *et al.* A cell surface-exposed protein complex with an essential virulence function in ustilago maydis. *Nat. microbiology* **6**, 722–730 (2021).

73. Weiland, P. & Altegoer, F. Identification and characterization of two transmembrane proteins required for virulence of ustilago maydis. *Front. plant science* **12**, 669835 (2021).

74. Schuster, M. *et al.* Novel secreted effectors conserved among smut fungi contribute to the virulence of ustilago maydis. *Mol. Plant-Microbe Interactions* **37**, 250–263 (2024).

75. Mehdipour, S., Mirroshandel, S. A. & Tabatabaei, S. A. Vision transformers in precision agriculture: A comprehensive survey. *Intell. Syst. with Appl.* **29** (2026).