

Review

Not peer-reviewed version

Maximizing a Farm Yield Through Precision Agriculture utilizing Fourth Industrial Revolution (4IR) Tools and Space Technology

[Mbulelo Siyabonga Perfect Ngongoma](#)^{*}, [Musasa Kabeya](#), [Katleho Moloi](#)

Posted Date: 3 March 2023

doi: 10.20944/preprints202303.0066.v1

Keywords: Fourth Industrial Revolution (4IR); Machine Learning (ML); Precision Agriculture; Space Vector Machine (SVM); Artificial Neural Network (ANN); k-Nearest Neighbour (k-NN); Fuzzy Classification; Global Navigation and Satellite System (GNSS)



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Review

Maximizing a Farm Yield Through Precision Agriculture Utilizing Fourth Industrial Revolution (4IR) Tools and Space Technology

Mbulelo SP Ngongoma ^{1,*}, Musasa Kabeya ² and Katleho Moloi ³

¹ Durban University of Technology; South Africa; Durban

² Durban University of Technology; South Africa; Durban; MusasaK@dut.ac.za

³ Durban University of Technology; South Africa; Durban; KatlehoM@dut.ac.za

* Correspondence: mbulelo26644@gmail.com; Tel.: +27 81 012 1513

Abstract: The globe and more particularly the economically developed regions of the world are currently in the era of the fourth Industrial revolution (4IR). Conversely; the economically developing regions in the world and more particularly the African continent have not yet even fully passed through the Third Industrial Revolution (3IR) wave and its economy is still heavily dependent on the agricultural field. On the other hand, the state of global food insecurity is worsening on an annual basis thanks to the exponential growth of the global human population which continuously heightens the food demand in both quantity and quality. This justifies the significance of the focus on digitizing agricultural practices to improve the farm yield to meet up with the steep food demand and stabilize the economy of the African continent and countries like India whose economy is mainly dependent on Agriculture. The tools we have at our disposal to utilize in the digitization of farming practices include space technology and Global Navigation and Satellite System (GNSS) in particular, Machine learning (ML), precision agriculture and communication systems such as the Internet of Things (IoT) and Information And Communication Technologies (ICT). The most pressing challenges in the farming field include the monitoring of diseases, pests, weeds and nutrient deficiencies in the crops as early detection translates to swift and timely correction actions and hence more yield at the end of a farming cycle. Vast opportunities in the field of precision agriculture still exist that can amount to further research studies such as the lack of real-time monitoring and real-time corrective action focus.

Keywords: Fourth Industrial Revolution (4IR); Machine Learning (ML); Precision Agriculture; Space Vector Machine (SVM); Artificial Neural Network (ANN); k-Nearest Neighbour (k-NN); Fuzzy Classification; Global Navigation and Satellite System (GNSS)

1. Introduction

Over the last two decades, we have seen a significant increase in the discussion of the Fourth Industrial Revolution (4IR) among academics and policymakers in both developing and industrialized countries [1]. The 4IR critique is marked by the merging of the real and virtual worlds, and the collapse of almost all industries [1, 2]. For others, the assembling of cyber-physical systems, cloud technology, the internet of things, internet of services; integrating them, while interacting with humans in real-time to maximize the generation of value, is known as Fourth Industrial Revolution [3]. Some thinkers assert that some old jobs will vanish because of the alleged revolutionary power of 4IR, opening the door for a new array of jobs and markets that will necessitate the creation of new areas of expertise [1-3]. The word "fourth" typically implies that there have been three revolutions before industrial revolution 4.0 [3]. Through mechanization and steam engines, the first industrial revolution greatly increased the productivity of manufacturing methods. Because there was more electrical power available during the second, assembly lines and mass production became a reality [4, 5]. The third industrial revolution saw the widespread adoption of computing and digitalization [6]. The 4IR is currently where we are and this era is dominated by the use of cyber-physical systems to improve life-sustaining processes such as production works, refer to figure 2. Growth in automation marked each shift from one revolution to the next [3]. Productivity rose by approximately fifty-fold

with each revolution, even if many jobs from the previous industrial age were rendered obsolete [7]. All revolutions, by their very nature, are disruptive, and the preceding three revolutions brought about significant modifications to the economic and social landscape [6, 7].

In the 1970s, it was believed that automating repetitive tasks would liberate people, resulting in more free time and less working time [1]. Despite advancements in technology, this promise remained mostly unmet [1, 2]. Now, the Fourth Industrial Revolution, which builds on digitalization and information and communication technologies (ICT), is thought to revolutionize everything [6]. It is projected that new technologies, including artificial intelligence (AI), biotechnology, the Internet of Things (IoT), quantum computing, and nanotechnology, will alter how we interact with one another, perform our jobs, run our economies, and even "the mere meaning of being a human being"[7]. It should be noted that the definition of the Fourth Industrial Revolution employed in this paper brightens a technology-centric understanding of 4IR, however, one should bear in mind the other important factors including the implications for society, politics, law, and ethics. Even though 4IR has been the topic of discussion on many international forums, there haven't been many systematic attempts to analyze the state of the art of this new industrial revolution wave [6]. This situation may be more apparent in Africa, where the third industrial revolution has mostly not even fully begun [1-3]. Therefore, African academics have expressed scepticism and caution over the alleged advantages of Information and Communications Technology (ICT) in African environments. Swaminathan[8] stated the following:

"Such a dream of transforming an agro-based economy into an information society must either be a flight of fancy or thinking hardly informed by the industrial economic background of developed economies that are in transition to informational economies. For an economy with about half of its adult population engaged in the food production sector, and about 70% of its development budget sourced from donor support, any talk of transition into an information society sounds like a far-fetched dream [8]"

Monzurul[9] argues that you cannot leap into the information age. Although African Leaders and officials have spoken out in support of 4IR's goals, most of the continent's nations continue to be heavily dependent on an agrarian economy [10]. Pachade [5] states that critics frequently compared community ICT projects that failed because of the technology-reality divide to driving Cadillacs in rural areas. While a Cadillac may be a luxurious car that many people admire and desire, its utilitarian value on dusty country roads would be far inferior to that of a tough pre-Second World War van. Africa has previously been described as a technological and digital wilderness [3, 10, 11]. It is evident that Africa still lags behind the rest of the international community regarding the 4th industrial revolution. This is due to several factors, such as poor infrastructure, and over-reliance on the primary sector - agriculture [6].

Agriculture remains the backbone of the African continent, and it is a crucial part of the global economy and plays an important role in providing food for the rapidly growing population and hence its heightened food demand [8, 10]. According to the United Nations, the world's population is anticipated to reach over 10 billion people by 2050, virtually doubling global food consumption [3]. Therefore, global, agricultural productivity will need to rise by 1.75% each year to meet the resulting food demand [3, 11]. The Global Harvest Initiative (GHI) estimates that productivity is currently increasing at a rate of 1.63% annually since the farmers are already being assisted by precision agriculture and advanced technologies like automation, machine learning, computer vision and artificial intelligence in keeping up with the food demand [5]. Global Navigation Satellite System (GNSS) is playing a particularly significant role as an enabler in the transformation of the agricultural sector through precision agriculture. Prashar[12] defines precision agriculture as a smart form of farm governance using digital systems, sensors, microcontrollers, actuators, robotics, and communication systems to achieve the goals of sustainability, revenue, and environmental conservation. Swaminathan[8] defines it as the integration of different computer tools into conventional agricultural methods to maximize the farm harvest and achieve self-sufficiency in farming operations. Precision agriculture (also known as digital farming or intelligent agriculture) includes, but is not limited to, the following: pest detection, weed detection, plant disease detection, morphology, irrigation monitoring and control, soil monitoring, air monitoring, humidity monitoring, and harvesting to name a few [4, 6-8, 12]. This paper aims to study in detail the recent research trends in precision agriculture, particularly in the disease/pest/weed detection area to comprehend the Artificial Intelligence (AI) tools and scientific background required to implement these Machine

Learning (ML) based precision agriculture systems. The disease/pest/weed detection system was chosen in particular because it possesses a multi-purpose architecture that can be applied for several and diverse applications in a farm with only amendments effected in the software and limited changes to the hardware. For example, a disease, weed, pest, nutrient deficiency, or morphological feature, and detection systems all have similar working principles where a high-quality picture is acquired from a farm specimen and an ML algorithm is then fed with that picture, after processing, to make a classification of what it sees on the given picture. Therefore, these systems can have similar prototypic architectures and a farmer can have one universal robotic system that has a few change-parts (such as: cameras, sensors.) and different software that are specific to different activities. This paper aims to present and summarize the recent research trends in precision agriculture, particularly in the disease/pest/weed detection area to identify the opportunities for further research. Its general architecture can be seen in figure 1. The following research questions are addressed in this study:

- What are the recent precision agriculture research developments particularly on the disease/pest/weed detection systems?
- What are the found limitations and gaps in the literature review?
- Lastly, what are the arising opportunities for further research?
- What topological amendments can be made to the traditional precision agricultural systems to make them more economical to employ in rural farms and make them more accessible?

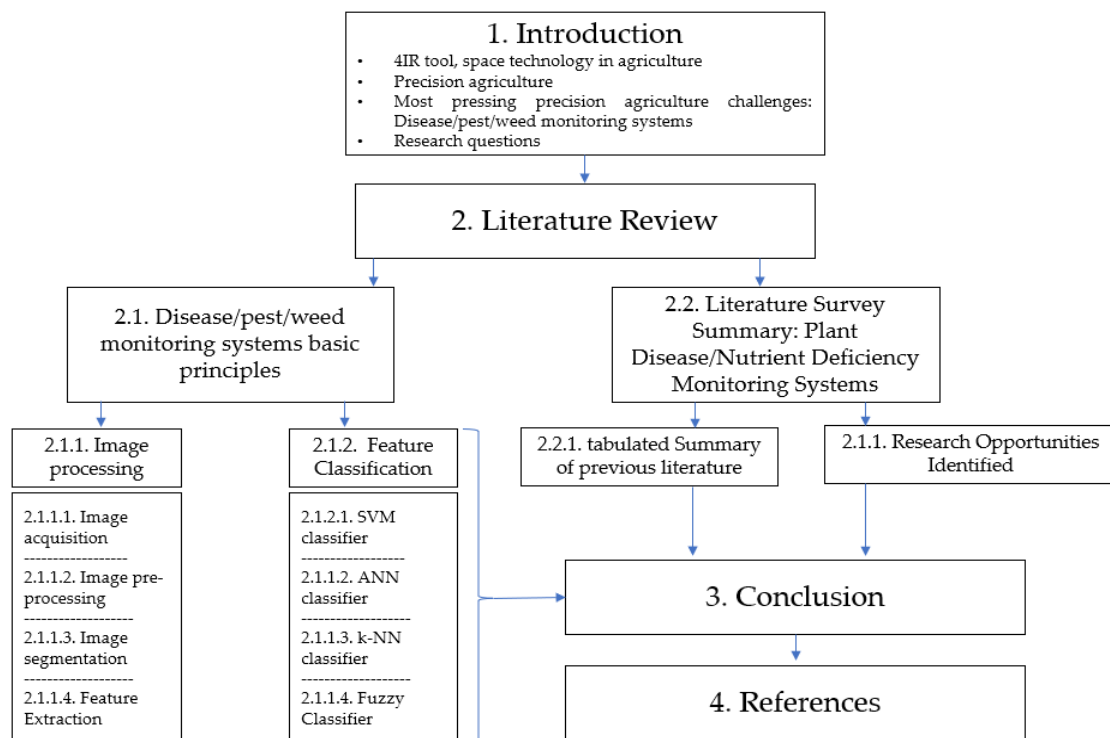


Figure 1. The general structure of this review paper

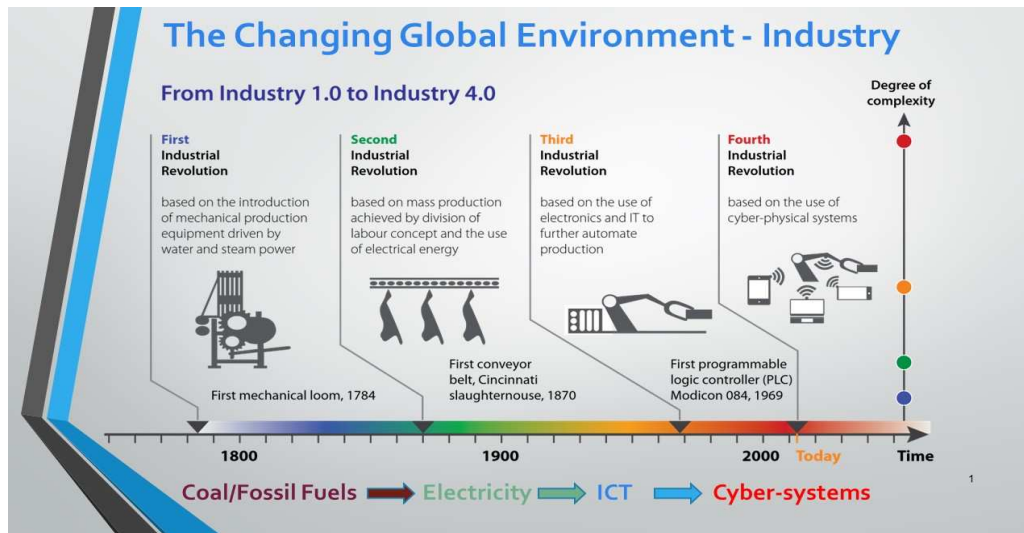


Figure 2. A historical perspective of the Changing Global Environment [DUT Inaugural Lecture – IED].

2. Literature Review: Precision Agriculture Research Developments

Disease, pest and weed monitoring and early identification are imperative in an effective farming operation [1]. In conventional agricultural practices, farmers rely upon visual observations of specimens to identify diseased leaves, fruits, roots and other parts of crops [4, 6]. However, this method is faced with several challenges ranging from the need for continuous checking and observation of specimens which is tedious and expensive for large farms but most importantly, very much less accurate [1, 2, 11]. Badage [1] asserts that agriculturalists often consult experts for the identification of infections on their crops which incur even more costs and have longer turnaround times. The earlier-stated limitations of classical farming methods coupled with the pressure to keep up with an exponentially growing demand for food both in quantity and quality have served as the push factors for researchers to devise new strategies and tools to digitize the agricultural field with the prime objective of increasing the farm yields and produce [13]. The coming subsection discusses the general plant disease detection system, and one should note that the same general topology can be used to monitor pests, weeds, morphological features and similar.

2.1. Plant Disease Detection System Basic Principles

Disease, pest or weed detection is achieved by utilizing machine learning (ML) [3, 5, 6, 11, 12]. Shruthi [3] Defines ML as an intelligent technique where a machine is capacitated to recognize a pattern, recall historical information, and train itself without being commanded to do so. Both supervised and unsupervised training strategies can be utilized for machine training [8]. While there are distinct training and assessment data sets for supervised training, there is no such distinction for unsupervised training data sets [12]. The author, Prashar [12], further states that since ML is an evolving procedure, the machine's performance becomes better with time. As soon as the machine has finished learning or training, it may classify the data, make predictions, and even generate fresh test data from which its re-trains itself and the process goes on and on [8]. Adekar [4] defines ML as a decision-making tool capable of visualizing the potentially complicated inter-relationships between important parameters in a farm and making educated predictions and/or decisions.

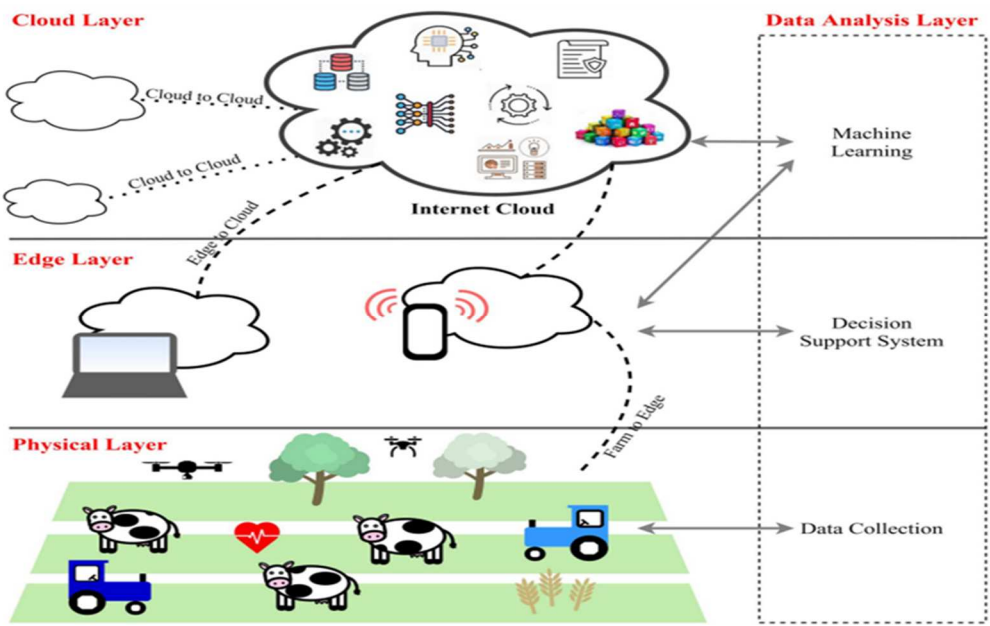


Figure 3. Format of precision agriculture system [4].

The author further makes an illustration of ML application in precision agriculture as seen in figure 1. In the three-level precision agricultural layout shown, the 1st level, which is the physical layer, represents all the field equipment such as sensors, trackers, actuators, and probes, to count a few, that is in physical connection with the farm environment collecting data for further processing [4]. In the second level, the edge layer is where the processing of the data collected in level 1 is taking place to convert the raw data into useful information which is used to inform the decision-making. Decision-making takes place at this level through computational tools like computers, microcontrollers, microprocessors, and similar [4]. In the third level, the cloud layer, the storage of data for iterative training of the machine takes place [4]. Therefore, the plant disease detection system is made up of two main sub-systems viz., the image processing system and the classification system. The image processing is further subdivided into 4 steps while there are also 4 most cited different classification protocols summarized in table 1.

Table 1. Table showing the summary of image processing steps and different classification techniques in plant disease detection.

A typical general plant disease detection system	
Summary of image processing steps	Different classification techniques
Image processing:	SVM Classifier
Image acquisition	ANN Classifier
Image pre-processing	KNN Classifier
Image segmentation	
Feature extraction	FUZZY Classifier
Machine learning classification	

Latest studies of phenomics and high-throughput picture data gathering are available however, most of the research on image interpretation and processing can be found in textbooks that dive into extensive detail into the methodologies [14]. Figure 2 summarizes the latest techniques for image acquisition and processing.

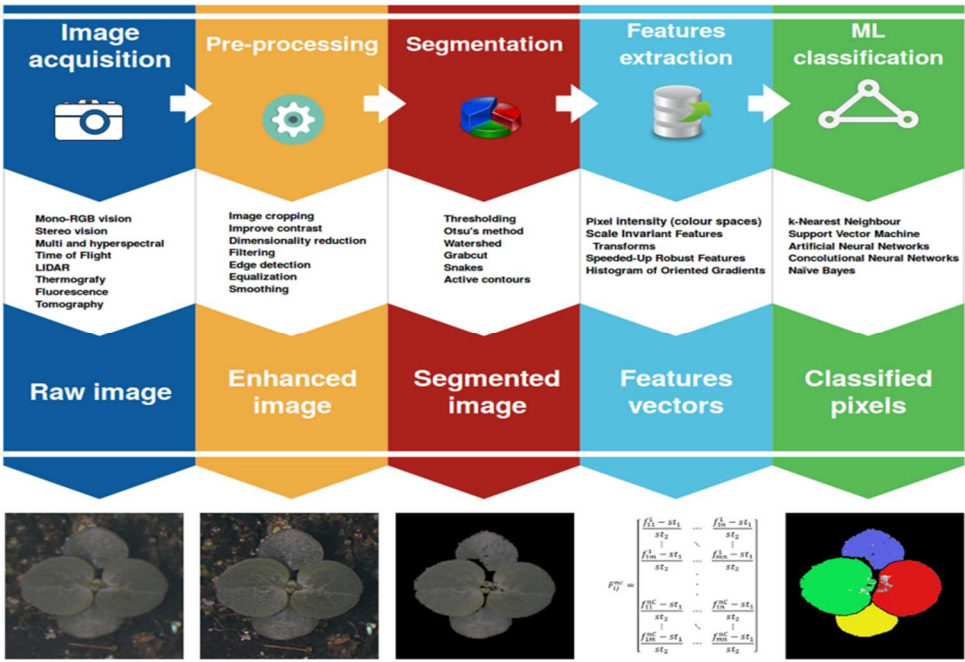


Figure 4. Steps of image processing [5].

2.1.1. Image Acquisition.

Image collection is the first step of a system for detecting plant diseases [6, 8, 12]. image sensors, scanners, and unmanned aerial vehicles (UAVs) can all be used to capture photos of plants [3]. The commonly utilized image acquisition tools are a charge-coupled device (CCD) and complementary metal oxide semiconductor (CMOS) [15]. Both these camera technology converts light signals and protons to digital data, which is then further transformed into a picture[15, 16]. However, their methods of turning the light signals into image data vary [16]. In a CCD camera, the light signals are transferred through a series of adjacent pixels before being amplified and converted into image data at the end of these pixel strings [17, 18]. This enables the CCD cameras to possess minimal degradation during the image acquisition process [19]. The CCD cameras generate sharp pictures with reduced distortion [18]. Contrarily, in CMOS cameras the light signals are collected, amplified, and converted at each pixel of the image sensor [15]. This enables the CMOS devices to generate images faster than CCD devices since each pixel can convert light signals into an image locally [17]. CMOS devices are normally preferred in projects that are low budget since they are cheap compared to CCD devices, have lower power consumption and can acquire high-quality images faster than their CCD counterparts[17-19]. Figure 3 shows the serial versus localized pixel image conversion of CCD and CMOS image sensors respectively.

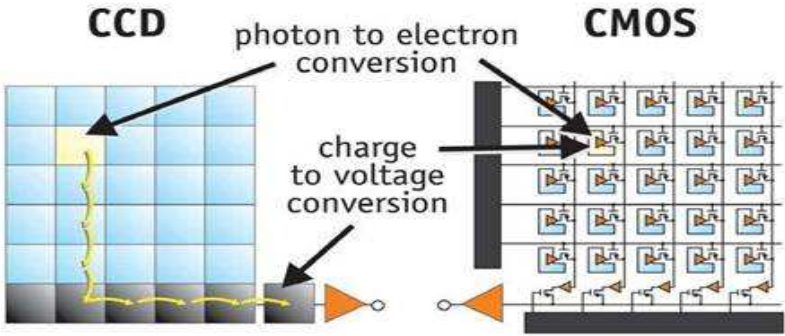


Figure 5. CCD vs. CMOS image conversion [15].

An imaging acquisition tactic known as time delay and integration (TDI) can be combined with either CCD or CMOS technology to drastically improve their image acquisition capabilities [20].

Applications involving fast-moving objects, requiring high precision and the capacity to function in extremely dim lighting environments use TDI [20, 21]. Refer to figure 3 for an example of a high-speed application of TDI technology where a high-velocity train was captured with a normal and TDI-featured camera on the left and right pictures respectively. When the camera was operated in normal mode the image of the train was a blur due to its high velocity and dim lighting conditions, however, the incorporation of a TDI mode counted these challenges and produce a clear detailed picture of a train.



Figure 6. Steps of image processing [5].

After an image has been captured with a CCD or CMOS device with or without TDI technology incorporated, a captured image should proceed to the following step of the image processing which is normally image segmentation [3, 5, 11, 12, 16]. The segmentation of an image is a process whereby the features of interest are extracted from the rest of the image and irrelevant features are masked [10]. The features of interest are referred to as the foreground while the irrelevant ones are referred to as the background [16]. The creation of the foreground versus background is dependent on picture properties like colour, spectrum brightness, edge detection, and neighbour resemblance, to count a few [17]. However, image pre-processing may occasionally be necessary before an effective image segmentation can take place [3, 8, 11, 22].

2.1.2. Image Pre-processing.

This is a crucial step in the ML-based disease detection system [14]. Pre-processing of an image deals with the correct setting of image contrast and filtration of interference signals resulting in noise and hence blurry images [18, 19]. This procedure can greatly enhance the precision of feature extraction and the correct disease detection in general [15]. Preprocessing typically involves straightforward treatments like image cutting, clipping, cropping, filtering, trimming, and deblurring; to name a few [3]. Wang[23] explained a typical image preprocessing procedure that is generally employed in image-based detection systems shown in figure 5.

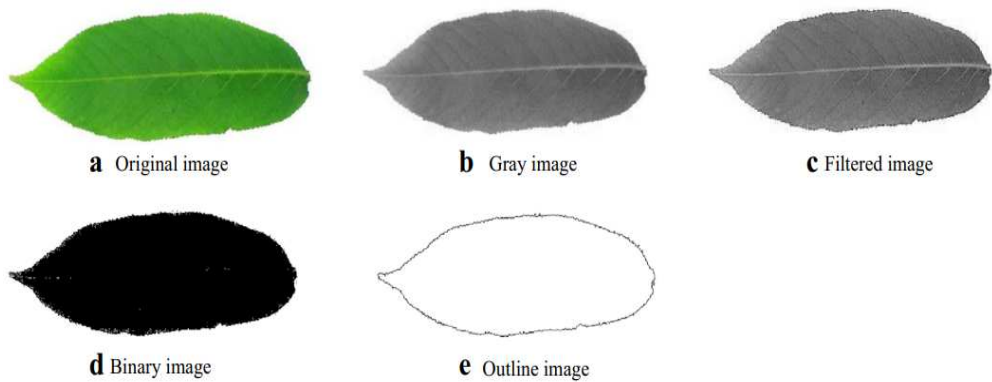


Figure 7. General preprocessing procedure for plant-based feature detection systems [23].

The first step in the procedure illustrated in figure 5 involves the transformation of a coloured image into a grey image [23]. This conversion stage into a grey image may be omitted in applications where colour features are of relevance otherwise, this step is crucial because it is much simpler and faster to process an image in a grey colour format [17]. The second stage involves the denoising of a specimen image as in most cases, images are not without interference with the noise signal which affects the visibility of the features in the specimen images [23]. The third step then includes image segmentation which will be explained more broader in 2.1.3. The last step involves the forming of an outline image which can be achieved by masking the leafstalk as well as holes while keeping the outer connected region [15, 23]. Wakhare[24] proposed a similar procedure to that illustrated in figure 5 for plant-leaf feature identification applications under real-life varying lighting conditions. This procedure involves the conversion of a specimen image into grayscale, noise suppression as well as smoothing, and formation of the image outline through edge filtering. In a comparative study conducted by Ekka[25], a histogram equalization method was proven to be the most effective form of image enhancement of the grey images that were originally colour images. Conversely, Kolhalkar[26] found that Red-Green-Blue (RGB) camera images offer more valuable image enhancement compared to those converted to greyscale in the context of identifying diseases on the plant leaves. Therefore, this study cannot conclude which image preprocessing technique is better than the other, rather the application in which the image is used, and thing kind of image involved in that application shall be considered in the selection of an appropriate preprocessing technique.

2.1.3. . Image Segmentation

Image segmentation is a pivotal part of image-based plant feature identification and phenotyping systems [23]. Segmentation of an image involves the separation between the foreground and the background [15], that is, the isolation of the feature of interest and masking of the irrelevant part from the image [24-26]. The features of interest are normally identified by comparing adjacent pixels for similarity by looking at the three main parameters, viz. the texture, colour, and shape [15, 17]. Table 2 shows a list of free data libraries available to the public for use in the image segmentation process.

Table 2. Table showing a list of image segmentation ML libraries

Software language of implementation	Library	Description	Open source
R	Kern-Lab	Mechanisms for segmentation, modelling, grouping, uniqueness identification, and feature matching using kernel-based deep learning [27].	https://cran.r-project.org/
	MICE	This method can deal with data sets with missing data by computing estimates and filling in the missing data values [28].	
	e1071	Programming package containing functions for types of statistical methods, i.e., probability and statistics [29].	
	CA-RET	Offers a wide range of tools for creating forecasting analytics utilizing R's extensive model library. it contains techniques for the	

		pre-processing learning algorithm, determining the relevance of parameters, and presenting networks [30].	
	Rweka	Data pre-processing, categorization, analysis, grouping, clustering algorithms, and image processing methods for all Java-based machine learning methods [31].	
	ROCR	A tool for assessing and displaying the accuracy of rating classifiers [32].	
	KlaR	various categorization and display functions [33].	
	Earth	Utilize the methods from Friedman's publications "Multivariate Adaptive Regression Splines" and "Fast MARS" to create a prediction model [34].	
	TREE	A library containing functions designated to work with trees [35].	
R, C	Igraph	Contains functions manipulating large graphs, and displaying [34].	
Python, R	Scikit-learn	Offers a standardized interface for putting the machine into learning algorithms practice. It comprises various auxiliary tasks like data preprocessing operations, information resampling methods, assessment criteria, and search portals for adjusting and performance optimization of methods [36].	
Python	NuPIC	Software for artificial intelligence that supports Hypertext Markup Language (HTML) learning models purely based on the neocortex's neurobiology [37].	http://numeta.org/
	Caffe	Deep learning framework that prioritizes modularity, performance, and expression [38]	http://caffe.berkeleyvision.org/
	Theano	A toolkit and processor that is optimized for working with and assessing equations, particularly those using array value [39].	http://deeplearning.net/software/theano
	Tensorflow	Toolkit for quick computation of numbers in artificial intelligence and machine learning [40].	https://www.tensorflow.org/
	PyBrain	A versatile, powerful, and user-friendly machine learning library which offers algorithms that may be used for a range of machine learning tasks [41].	http://pybrain.org/
	Pylearn2	A specially created library for machine learning to make learning much easier for developers. It is quick and gives a researcher a lot of versatility [42].	http://deeplearning.net/software/pylearn2
Java	Java-ML	A collection of machine learning and data mining techniques that aim to offer a simple-to-use and extendable API. Algorithms rigorously adhere to their respective interfaces, which are maintained basic for each type of algorithm's interface [43].	http://java-ml.sourceforge.net/
	ELKI	A data mining software that intends to make it possible to create and evaluate sophisticated data mining algorithms and study how they interact with database search architecture [44].	http://elki.dbs.uni.lmu.de/
	JSAT	A library designed to fill the need for a general-purpose, reasonably high-efficiency, and versatile library in the Java ecosystem that is not sufficiently satisfied by Weka and Java-ML [45].	https://github.com/EdwardRaff/JSAT
	Mallet	Toolkit for information extraction, text categorization, grouping, quantitative natural language processing, as well as other deep learning uses to text [46].	http://mallet.cs.umass.edu/
	Spark	Offers a variety of machine learning techniques such as grouping, categorization, extrapolation, and data aggregation, along with auxiliary features like simulation assessment and data acquisition [47].	http://spark.apache.org/
	Weka	Provides instruments for categorizing, forecasting, clustering, classification techniques, and visualization of information [48].	http://www.cs.waikato.ac.nz/ml/weka/
C#, C++, C	Shark	Includes approach for neural networks, both linear and nonlinear programming, kernel-based learning algorithms, and other methods for machine learning [49].	http://image.diku.dk/shark/
	mlpack	Gives the data processing techniques as simplified control scripts, Python bindings, and C++ objects that can be used in more extensive machine learning solutions [50].	http://mlpack.org/

LibSVM	A Support Vector Machines (SVM) library [51].	http://www.csie.ntu.edu.tw/~lin/libsvm/
Shogun	Provides a wide range of data types and techniques for deep learning issues. It utilizes SWIG to provide interfaces for Octave, Python, R, Java, Lua, Ruby, and C# [52].	http://shogun-toolbox.org/
Multiboot	offers a quick C++ solution for enhancing methods for many classes, labels, and tasks[53].	http://www.multiboost.org/
MLC++	Supervised machine learning methods and functions in a C++ ecosystem [52].	http://www.sgi.com/tech/mlc/source.html
Accord	Fully C#-written machine learning platform with audio and picture analysis libraries [54].	http://accord-framework.net/

A most straightforward example of an image segmentation technique is thresholding [55]. Threshold segmentation is a process of converting a colour or grey scale image into a binary image with the sole purpose of making feature classification easier [55, 56]. The output binary images consist of black and white coloured pixels which correspond to the background and foreground respectively or vice versa[26, 55, 56].

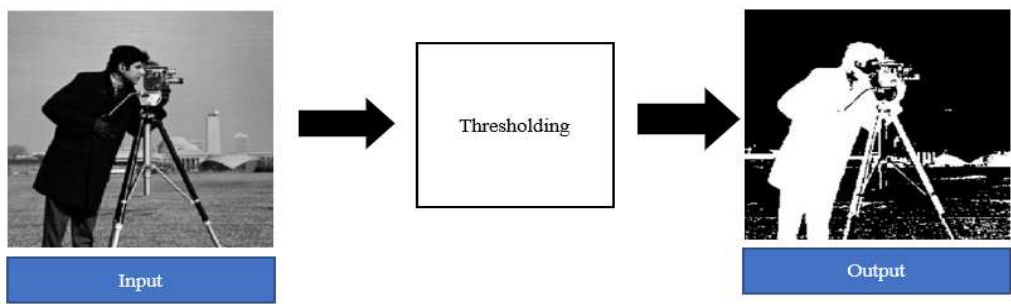


Figure 8. Example of thresholding image segmentation [26].

Threshold segmentation is mathematically defined as follows, where T refers to a certain threshold intensity, g is the black or white pixel of a binary image and f is the grey level of the input picture [56]:

$$g(x,y)=\begin{cases} 0, & \text{if } f(x,y) < T \\ 1 & \text{if } f(x,y) > T \end{cases} \tag{1}$$

Threshold segmentation is subdivided into three, global, local, and adaptive thresholding [15, 57]. Global thresholding is applied in scenarios where there is enough distribution between the intensity distribution of the foreground compared to the background [15]. Hence a single threshold value is selected and used to distinguish between the features of significance and the background [15, 55]. Local thresholding is applied in cases where there is no distinct difference in intensity distribution between the background and the foreground and hence not conducive to selecting a single threshold value [55]. In such a case, an image is partitioned into smaller images and select different threshold values for each partitioned picture [15]. Adaptive thresholding is also appropriate for images with uneven intensity distribution where a threshold value is calculated for each pixel [57]. The Otsu thresholding method is another thresholding technique used for image segmentation [15]. In this technique, a measure of spread for the pixel intensity levels on either side of the threshold is listed by looping through all the reasonable threshold values [58]. The intent is to decide the threshold value where the summation of foreground and background escalates is at its minimum [15, 58]. The fundamental characteristic of the Otsu thresholding method is the fact that it implements the threshold values automatically instead of it being preselected by the user [58]. (2) the mathematical definition for the thresholding in the Otsu method.

$$g(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & 0 \text{ otherwise} \end{cases} \quad (2)$$

Another segmentation method application in image processing is watershed transformation [59]. A grayscale image undergoes a transition called a watershed [59, 60]. In a metaphorical sense, the name alludes to a geologic catchment or drainage split that divides parallel catchments [59]. The watershed conversion locates the lines that follow the tops of ridges by treating the image it operates upon as a topographic map, with the luminosity of each pixel denoting its elevation [60]. Figure 5 is an example of a watershed-segmented image where the black pixels denote the background, the grey pixels denote the features to be extracted and the white pixels correspond to watershed lines [61].



Figure 9. Watershed image segmentation example [61].

On the other hand, Grabcut is a very popular and innovative segmentation technique that takes into consideration the textural and boundary conditions of an image [62]. This segmentation method is based on the iterative graph-cut method where a mathematical function is derived to implement the background as well as the foreground [63]. Each pixel in an image is then assessed to decide whether it falls in the background or the foreground [62, 63]. The Grab-cut segmentation method is preferred in most applications because of minimal user interference in the operation of this technique however it is not without its drawbacks [62]. The Grab-cut sequence cycles take a long to implement because of the complexity of the thresholding equation [63]. The segmentation is also poor in scenarios where the background is complex and minimal distinction between the features of interest and the background [64]. Several distinct segmentation methods and algorithms exist in the literature. The suitability of one method is based on a particular application and hence this study is not in a position to rule out certain segmentation methods or merit the ones outperforming the others.

2.1.4. . Feature Extraction

One of the foundational elements of computer vision-based image recognition is the extraction of features [65]. A feature is a data that is utilized to solve a particular computer vision problem and is a constituting part of a raw image [64]. The feature vectors include the features that have been retrieved from an image [66]. An extensive range of techniques is used to identify the items in an image while creating feature vectors [62]. Edges, image pixel intensity, geometry, texture, image modifications like Fourier, Wavelet, or permutations of pixels from various colour images are the primary features [46, 66]. A set of classifiers and machine learning algorithms are what feature extraction is ultimately used for [66]. The feature extraction in plant leaf disease monitoring systems is subdivided into three spheres which include texture, colour and shape [20, 21, 46, 65].

2.1.4.1. Shape Features

The shape is a basic characteristic of a leaf used in feature extraction of leaf images during image processing [66]. The primary shape parameters include the length (L) which is the displacement between the two points in the longest axis, the width (W) which denotes the displacement between

the shortest axis, the diameter (D) denoting the maximum distance between the points, the area (A) which denotes the surface area of all the pixels found within the margin of a leaf picture and the perimeter (P) which denotes the accumulative length of the pixels around the margin of a leaf picture [55, 58, 62, 64]. From the 5 defined primary characteristics of shape features, 11 distinct secondary features are formed by mathematical definitions involving two or more primary variables [59]. These 11 features are called morphological features of a plant. The morphological features are as follows:

- Circularity (C) – A feature defining the degree to which a leaf conforms to a perfect circle. It is defined as [60]:

$$C = \frac{4\pi A}{P} \quad (3)$$

- Rectangularity (R) – A feature defining the degree to which a leaf conforms to a rectangle. It is defined as [55]:

$$R = \frac{LW}{A} \quad (4)$$

- Aspect Ratio (AS) – Ratio of width to length of a leaf. It is defined as [55]:

$$AS = \frac{W}{L} \quad (5)$$

- Smooth factor (SF) – Ratio of leaf picture area when a 5x5 and 2x2 regular smoothing filters have been used [58].
- Perimeter to diameter ratio (PDr) – Ratio of the perimeter to the diameter of a leaf. It is defined as [64]:

$$PDr = \frac{P}{D} \quad (6)$$

- Perimeter to length plus width ratio (PLWr) – Ratio of the perimeter to length plus width of a leaf. It is defined as [64]:

$$PLWr = \frac{P}{L + W} \quad (7)$$

- Narrow factor (NFr) – Ratio of diameter to length of a leaf [60]:
- Area convexity (ACr) – Area ratio between the area of a leaf and the area of its convex hull [59].
- Perimeter convexity (ACr) – The ratio between the perimeter of a leaf to that of its convex hull [60].
- Eccentricity (Ar) – The degree to which a leaf shape is a centroid [64].
- Irregularity (Ir) – Ratio of the diameters of an inscribed to the circumscribed circles on the image of a leaf [59].

2.1.4.2. Colour Features

Other researchers and scholars chose to implement the colour features as the pivotal features during the extraction process [67]. The colour features normally cited in the literature on leaves feature extraction include the following:

$$NFr = \frac{D}{L} \quad (8)$$

- Colour standard deviation (σ) – A major of how much the different colours found in an image match one another or are rather different from one another [60]. Say an image is differentiated into an array of its basic building blocks, the pixels, then i is a pointer moving across the rows of pixels in an array from the origin to the very last row M while j is a pointer moving across the columns of pixels in an array from the origin to the very last column N. At any point, a pixel colour intensity is defined by p(i, j) where i and j denote the coordinate position of a pixel in an image array. Therefore, the colour standard deviation is mathematically defined as follows:

$$\sigma = \frac{1}{MN} \sqrt{\sum_{i=1}^M \sum_{j=1}^N p(i, j)} \quad (9)$$

- Colour mean (μ) – A major to identify a dominant colour in a leaf image. This feature is normally used to identify the leaf type [63]. It is mathematically defined as follows:

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \quad (10)$$

- Colour skewness (φ) – A major to identify a colour symmetry in a leaf image [21, 46]:

$$\varphi = x = \frac{\sum_{i=1}^M \sum_{j=1}^N [p(i, j) - \mu]^3}{MN\sigma^3} \quad (11)$$

- Colour Kurtosis (φ) – A major to identify a colour shape dispersion in a leaf image [65]:

$$\varphi = x = \frac{\sum_{i=1}^M \sum_{j=1}^N [p(i, j) - \mu]^4}{MN\sigma^4} \quad (12)$$

2.1.4.3. Texture Features

There are also several textural features referenced by authors such as Singh [68], Martsepp[69] and Ponce[70]. Using the same assumption of an image partitioned into pixels in section 2.1.4.2, the following are the textural features used for feature extraction in plant leaves:

- Entropy (Entr) – This is a measure of how complex and uniforms a texture of a leaf image [68]:

$$\text{Entr} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p(i, j) \log_2 p(i, j) \quad (13)$$

- Contrast (Con) – This is a measure of how clear the features are in a leaf image, it is also referred to as the moment of inertia[69, 70]:

$$\text{Cont} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (i - j)^2 p(i, j) \quad (14)$$

- Energy (En) – This is a major of the degree of how the uniform is a grey image. It is also called the second moment [69]:

$$\text{En} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p^2(i, j) \quad (15)$$

- Correlation (Cor) – This is a major of whether there is a similar element in a sample picture, which corresponds to the re-occurrence of a similar matrix within a large array of pixels [68].

$$\text{Cor} = \frac{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N ijp(i, j) - a_1 a_2}{b_1^2 b_2^2} \quad (16)$$

Where:

$$a_1 = \sum_{i=1}^N ip(i, j)$$

$$a_2 = \sum_{j=1}^N jp(i, j)$$

$$b_1^2 = \sum_{i=1}^N (i - a_1)^2 \sum_{j=1}^N p(i, j)$$

$$b_2^2 = \sum_{j=1}^N (j - a_2)^2 \sum_{i=1}^N p(i, j)$$

- Difference moment inverse (DMI) – This is a major of the degree of how homogenous an image [69]:

$$\text{DMI} = \sum_{i=1}^M \sum_{j=1}^N \frac{p(i, j)}{1 + (i - j)^2} \quad (17)$$

Other textural features include the maximum probability, which is the highest response to correlation, the standard deviation and/or variance which is the aggregate texture observed in a leaf picture and the average illuminance which is the average light distribution across the leaf when an image was captured, to name a few [66, 68-70]. The selection of whether to use which colour, shape or textural feature strictly depends on the application of the system being designed.

2.1.5. Feature Classification Through Machine Learning Algorithms

The classification techniques are the machine learning algorithms that are used to categorize input sample data into different classes or groups of belonging or membership [3, 5, 11, 56]. These classifiers may employ supervised learning, unsupervised learning and reinforcement learning methods during their training [39]. Supervised learning occurs when a person is a trainer of the model and may use pre-formed data sets to do the training [39, 53]. Unsupervised learning occurs when there is no training data available hence the algorithm must train itself and improve its classification efficiency by iteratively adjusting itself [5, 39, 53]. Reinforcement learning occurs when the algorithm makes classification rulings based on the feedback applied by the environment to it [12, 39]. In the case of vision-based plant disease monitoring systems, the most cited classification algorithms include Space Vector Machines (SVM), Artificial Neural Networks, k-Nearest Neighbor machines and Fuzzy machines. The coming subsections discuss these classification techniques.

2.1.5.1. SVM Classifier

Support Vector Machine, sometimes known as SVM, is a predictive model used to solve both regression and classification tasks [3]. It is a supervised learning model that works well for numerous practical problems and can solve both linear and non-linear tasks [3, 71]. The SVM concept is straightforward; a vector or a hyperplane that splits the data into groups is generated by this technique [72].

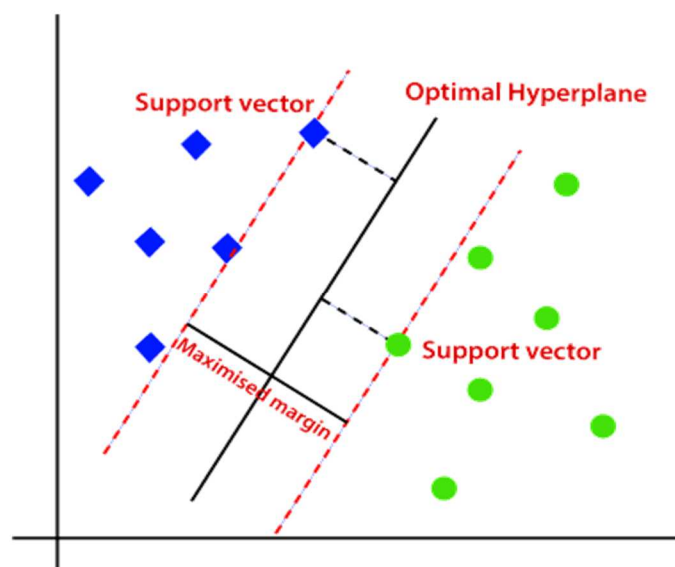


Figure 10. SVM classification algorithm [72].

In figure 6, the optimal hyperplane is used to separate the two classes of data (the blue squares and green circles). The two planes (dashed lines) parallel to the optimal hyperplane are called the positive and negative imaginary planes which are the planes passing through the closest data points to either side of an optimal hyperplane [72]. These closest points to the optimal hyperplane are called the support vectors and are used to determine the exact position of an optimal hyperplane [73]. There might be several possible hyperplanes but the optimal hyperplane is the one with the maximum marginal distance, which is the distance between the two marginal planes [72, 73]. The maximized margin results in a more generalized solution compared to smaller margins and should the training data change, the algorithm with a smaller margin will have accuracy challenges [73]. In some cases, data classes are not always easily separable with a straight line or plane as in the case of figure 6. Therefore when data classes show a property of non-linearity, transforming a space in which these data classes occur from a low dimension (often 2-dimensional) into a high-dimension space (often 3-dimensional) space using the Kernel method. The Kernel method is a computation of a dot product of the dimensions in the new high-dimension space [72-74]. (18) gives the general solution of a hyperplane where \vec{x} is any data point or support vector, \vec{w} is the weight vector that applies the bias of the support vectors and w_0 is the constant [74].

$$g(\vec{x}) = \vec{\omega}\vec{x} + \omega_0 \quad (18)$$

$$\text{and } g(\vec{x}) = \begin{cases} 1, & \text{for Class 1 vectors} \\ -1, & \text{for Class 2 vectors} \end{cases}$$

2.1.5.2. ANN Classifier

An ANN is a supervised learning model that is a collection of interlinked input and output nodes in which each link has an associated bias value called a weight [75]. A single input layer, one or perhaps more intermediate layers which are normally called hidden layers, and one or more output layers make up the structure of an ANN [75, 76]. The weight of each connection is modulated as the network operates to facilitate neural network learning [76]. The performance of the network is enhanced by adjusting the weight continuously [75]. ANN can be divided into two groups based on connection types: feed-forward networks and recurrent networks [33]. In contrast to recurrent neural networks, feed-forward neural networks do not have cycle-forming connections between units [76]. The architecture, transfer function, and learning rule all have an impact on how a neural network behaves [49, 76]. The weighted total of input triggers the activation of neural network neurons [75]. Figure 7 shows a generalized model of an ANN model with the input layer, the hidden intermediate layer (purple layer) and the output layer.

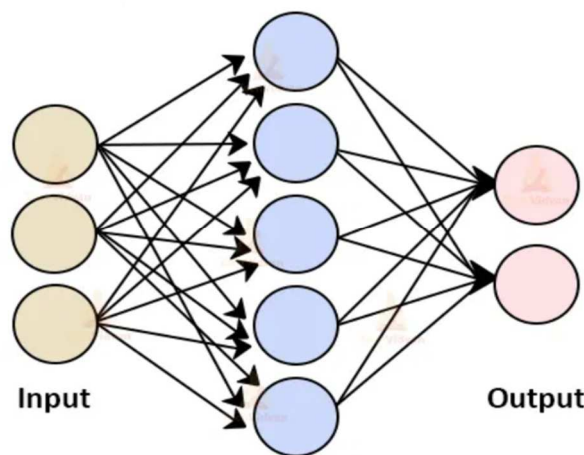


Figure 11. ANN model architecture [77].

2.1.5.3. kNN Classifier

The k Nearest Neighbors algorithm, sometimes known as kNN, is the most straightforward machine learning technique [77]. It is a non-parametric technique used for problems involving regression and classification [74, 77]. Non-parametric implies that no dataset for initial training is necessary [77]. Therefore, kNN does not require the use of any presumptions [78]. The k-closest training examples in the feature space provide the input for classification and regression tasks, respectively [77]. Whether kNN is applied for classification or regression determines the results [78]. The outcome of the kNN classifier is a class of belonging [74, 77, 78]. Based on the predominant kind of its neighbourhood, the given data point is classed [78]. The input point is awarded to the category that has the highest frequency among its k closest neighbours [77]. In most cases, k is a little positive integer such as 1. The result of a kNN regression is just a value of the property for the attribute. The aggregate of the variables of the k closest neighbours constitutes this number [78].

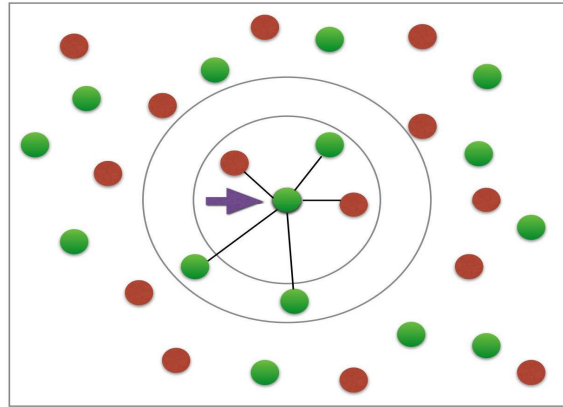


Figure 12: Classification principle of a kNN model [79].

Figure 10 shows a space with numerous data points or vectors that can be classified into two classes: the red class and the green class. Now, assume there exists a data point at any location in the space shown in figure 8 that is unknown whether it belongs to either the red or green class. The kNN will then go through the following computational steps to assign that point a class of belonging:

- Take the uncategorized data point as input to a model
- Measure the spatial distance between this unclassified point to all the other already classified points. The distance can be computed via Euclidean, Minkowski or Manhattan formulae [79].
- Check the points with the shortest displacement from the unknown data point to be classified for a certain K value (K is defined by the supervisor of the algorithm) and separate these points by class of belonging [79].
- Select the correct class of membership as the one with the most frequent vectors as the neighbours of the unknown data point [79].

The most cited method of computing the spatial distance between the data point p to be classified and its neighbours q_n is the Euclidean formulae (19) [74, 79]:

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (18)$$

2.1.5.4. FUZZY Classifier

The fuzzy classifier system is a supervised learning model that enables computational variables, outputs, and inputs to assume a spectrum of values over predetermined bands [80]. By developing fuzzy rules that connect the values of the input variables to internal or output variables, the fuzzy classifier system is trained [81]. It has mechanisms for credit assignment and conflict resolution that combine elements of typical fuzzy classifier systems [80]. A genetic algorithm is used by the fuzzy classifier system to develop suitable fuzzy rules [82].

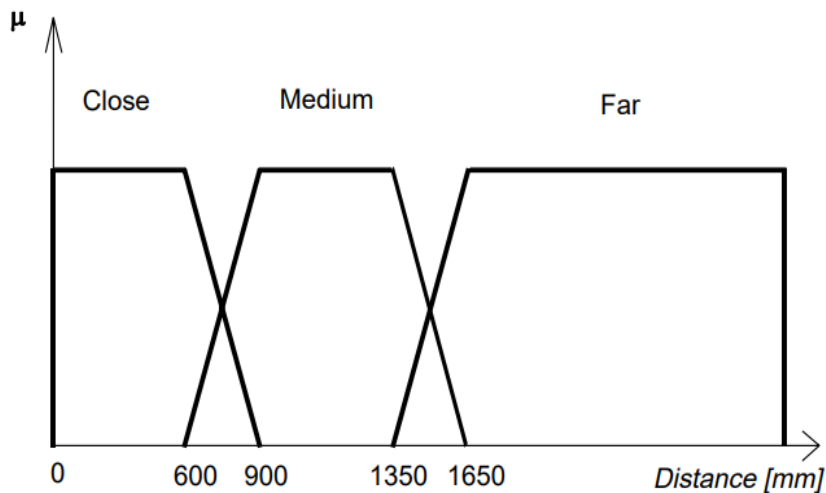


Figure 13. Example of FUZZY sets for classification [82].

As shown in figure 12, fuzzy sets display a continuous membership, and a data point membership classification can be ruled as the extent (μ) to which it belongs to a certain fuzzy set. For example, 690 mm in figure 9 has a degree of membership $\mu(960)$ on the close fuzzy set that is 0.7. It can also be seen from figure 9 that a data point can belong to multiple fuzzy sets, and the degrees of membership to each set may or may not (in the intersection points) differ, since some fuzzy sets overlap with each other. Table 3 summarizes the advantages and disadvantages of all the classification techniques discussed in this section.

Table 3. Table showing PRO & CONS of different classification methods

PROS & CONS OF DIFFERENT CLASSIFICATION METHODS MOST USED IN PLANT PHENOMICS AND DISEASE MONITORING	
1. Support Vector Machine (SVM)	
Advantages	Disadvantages
Works very accurately when there is a clear formation of a hyperplane [74].	Accuracy difficulties with a large amount of training data [71].
Works more accurately on high-dimension spaces like 3-D and 4-D [51].	Susceptibility to noise and overlapping data classes [75].
Saves memory space [71].	The number of characteristics for a single data set must not exceed the number of data points in the training set [74].
2. Artificial Neural Network (ANN)	
Capable of multitasking [76].	Complex programming algorithms [75].
The machine is learning continuously, and the accuracy is improving iterable [50].	Accuracy is data dependent; more training data translate to more accurate classification and the opposite is true [75].
Have many applications (e.g., mining, agriculture, medicine, and engineering) [59].	Hardware reliance (cost, complexity and maintenance) [33].
3. k-Nearest Neighbor (kNN)	
No initial training period [74].	Accuracy difficulties with a large amount of training data [78].
Simple to add new data to the model to extend its scope [79].	Not suitable for high-dimensional space [79].
Relatively easy to implement with only the two parameters to work out, the k value and the geometric distance between the points [77].	Susceptibility to noise and outliers [74].
4. Fuzzy classifier	

Unclear, distorted, degraded or vague input data is accommodated by the model [80].	Depending on people's experience and expertise [81].
More flexibility and ease to change the rules [82].	Require excessive supervision in a form of testing and validation [81].
Robust in applications with no exact input format [81].	The is no universal approach to implementing fuzzy classification models which adds to their inaccuracy [82].

2.2. Literature Survey: Plant Disease/Nutrient Deficiency Monitoring Systems

Many authors in the literature have proposed plant disease/pest/weed detection systems that employ the above-described general format. Table 4 summarizes a literature survey on these systems.

2.2.1. Tabulated summary of Plant Disease/Nutrient Deficiency Monitoring Systems publications

Table 4. Table summarizing a literature survey on the plant disease/pest/weed detection systems

Classification Method	Plant/Crop	Reference	Number of diseases	Disease	Results
SVM Classification	Maize	[83]	1	Not Specified	79% accuracy
	Grapefruit, Lemon, lime	[84]	2	Canker And Anthracnose Diseases	95% accuracy for both
	Grape	[85]	2	Downy Mildew And Powdery Mildew	88.89% accuracy for both
	Oil palm	[3]	2	Chimaera And Anthracnose	97% and 95% accuracy respectively
	Potato	[86]	4	Late Blight And Early Blight	95% for both
	Grape	[10]	3	Black Rot, Esca And Leaf Blight	Not specified
	Tea	[87]	3	Not Specified	90% accuracy
	Soybean	[84]	3	Downy Mildew, Frog Eye, And Septories Leaf	90% accuracy average
	Tomato	[88]	6	Not Specified	96% accuracy
	Rice	[89]	Not specified	Pests Diseases	92% accuracy
	Soybean	[90]	1	Charcoal Rot	90% accuracy
	Cucumber	[91]	1	Downy Mildew	Not specified
	Rice	[92]	1	Rice Blast	93% accuracy
	Rice	[93]	1	Rice Blight	80% accuracy
	Tea	[94]	1	Not Specified	90% accuracy
ANN Classification	zucchini	[95]	1	Soft-Rot	Not specified
	Not specified	[96]	4	Alternaria Alternata, Anthracnose, Bacterial Blight, Cercospora Leaf Spot	96% accuracy average
	Grapefruit	[97]	3	Grape-Black Rot, Powdery Mildew, Downy Mildew	94% accuracy average
	Apple	[98]	3	Apple Scab, Apple Rot, Apple Blotch	81% accuracy average
	Pomegranate	[99]	3	Bacterial Blight, Aspergillus Fruit Rot, Gray Mold	99% accuracy average

	Not specified	[100]	4	Early Scorch, Cottony Mould, Late Scorch, Tiny Whiteness	93% accuracy average
	Cucumber	[101]	2	Downy Mildew, Powdery Mildew	99% accuracy average
	Pomegranate	[102]	4	Leaf Spot, Bacterial Blight, Fruit Spot, Fruit Rot	90% accuracy average
	Groundnut	[103]	1	Cercospora	97% accuracy
	Pomegranate	[104]	1	Not Specified	90% accuracy
	Cucumber	[105]	1	Downy Mildew	80% accuracy
	Rice	[106]	3	Bacterial Leaf Blight, Brown Spot, Leaf Smut	96% accuracy average
	Citrus	[107]	5	Anthracnose, Black Spot, Canker, Citrus Scab, Melanose	90% accuracy average
	Wheat	[108]	4	Powdery Mildew, Rust Puccinia Triticina, Leaf Blight, Puccinia Striiformis	Not specified
	Not specified	[109]	5	(YS) the yellow spotted, (WS) white spotted, (RS) red spotted, (N) Normal and (D) discoloured spotted	86% Accuracy
k-NN Classification	Groundnut	[77]	5	Early leaf spot, Late leaf spot, Rust, early and late spot Bud Necrosis	96% Accuracy
	Tomato, Corn, Potato	[110]	Not Specified	No disease: Leaf Recognition	94% Accuracy (Corn) 86% Accuracy (Potato) 80% Accuracy
	Tomato	[111]	3	Rust, early and late spot Bud Necrosis	95% Accuracy
	Banana	[112]	2	bunchy top, sigatoka	99% Accuracy
	Tomato	[113]	3	Rust, early and late spot Bud Necrosis	97% Accuracy
	Rice		4	Bacterial Blight of rice, Rice Blast disease, Rice Tungro, False smut	88% Accuracy Average
	Mango	[82]	3	Powdery Mildew, Phoma blight, Bacterial canker	90% Accuracy Average
Fuzzy Classification	strawberry	[114]	1	Iron deficiency	97% Accuracy
	Cotton, Wheat	[115]	18	Bacterial blight, Leaf Curl, Root Rot, Verticillium wilt, Anthracnose, Seed rot, Tobacco streak virus, Tropical rust, Fusarium wilt, Black stem rust, leaf rust, stripe rust, Loose smut, Flag smut,	99% Accuracy Average

		complete bunt, partial bunt, Ear_cockle, Tundo		
Soybean	[19]	1	Foliar	96% Accuracy
Cotton		3	Bacteria blight, Foliar, Alternaria	95% Accuracy Average

2.2.2. Research Opportunities Identified

During the literature survey presented in the earlier presented sections, the following opportunities which the author of this paper believes has seen little or no interest from the researchers are as follows:

- Little or no literature discussed the real-time monitoring of the onset signs of diseases before they spread throughout the whole plant part.
- Few papers discussed real-time monitoring and real-time mitigation measures such as actuation operations, spraying pesticides, and spraying fertilizers, to name a few examples.
- Very little research discusses the combination of these monitoring and phenotyping tasks into 1 system to reduce costs and improve technology availability to farmers and add convenience.

3. Conclusion

This paper has presented the background for the research in precision agriculture. The most pressing challenge in precision agriculture was identified as the monitoring of disease/pest/weed/nutrient deficiency in crops, backed by the literature trends. The basing principles of a disease/pest/weed/nutrient deficiency monitoring system were discussed. This covered the image processing steps viz., image acquisition, image pre-processing, image segmentation and feature extraction; and the different classification algorithms most cited in literature viz., the SVM, the ANN, k-NN and Fuzzy classifiers. The summary of a literature survey regarding the disease/pest/weed/nutrient deficiency monitoring systems was presented. The main opportunities observed during a literature survey were then presented.

A literature survey reveals that this field of precision agriculture is a relatively new research field and that considerable research focus and hence progress has been observed over the past two decades. It is also apparent that there still exists an array of opportunities that still require to be filled with further research such as those presented in 2.2.2. Much more can still be done to further improve the accuracy levels of some monitoring systems presented in table 4 such as increasing the amount of training data. This study is already serving as a foundation for a Doctor of Philosophy research which seeks to explore one of the research opportunities presented in this paper.

References

1. A. Badage, "Crop disease detection using machine learning: Indian agriculture," *Int. Res. J. Eng. Technol.*, vol. 5, no. 9, pp. 866-869, 2018.

2. U. Ukaegbu, L. Tartibu, T. Laseinde, M. Okwu, and I. Olayode, "A deep learning algorithm for detection of potassium deficiency in a red grapevine and spraying actuation using a raspberry pi3." pp. 1-6.

3. U. Shruthi, V. Nagaveni, and B. Raghavendra, "A review on machine learning classification techniques for plant disease detection." pp. 281-284.

4. S. Adekar, and A. Raul, "Detection of Plant Leaf Diseases using Machine Learning," *2019 International Research Journal of Engineering and Technology (IRJET)*, 2019.

5. R. S. Pachade, "A REVIEW ON COMPARATIVE STUDY OF MODERN TECHNIQUES TO ENHANCE INDIAN FARMING USING AI AND MACHINE LEARNING."

6. M.-L. du Preez, "4IR and Water Smart Agriculture in Southern Africa: A Watch List of Key Technological Advances," 2020.

7. M. S. Hoosain, B. S. Paul, and S. Ramakrishna, "The impact of 4IR digital technologies and circular thinking on the United Nations sustainable development goals," *Sustainability*, vol. 12, no. 23, pp. 10143, 2020.

8. B. Swaminathan, "Identification of Plant Disease and Wetness/Dryness Detection."

9. M. Islam, K. A. Wahid, A. V. Dinh, and P. Bhowmik, "Model of dehydration and assessment of moisture content on onion using EIS," *Journal of food science and technology*, vol. 56, no. 6, pp. 2814-2824, 2019.

10. S. Anju, B. Chaitra, C. Roopashree, K. Lathashree, and S. Gowtham, "Various Approaches for Plant Disease Detection," 2021.

11. S. Swain, S. K. Nayak, and S. S. Barik, "A review on plant leaf diseases detection and classification based on machine learning models," *Mukt shabd*, vol. 9, no. 6, pp. 5195-5205, 2020.
12. N. Prashar, "A Review on Plant Disease Detection Techniques." pp. 501-506.
13. N. Agrawal, J. Singhai, and D. K. Agarwal, "Grape leaf disease detection and classification using multi-class support vector machine." pp. 238-244.
14. Z. A. Dar, S. A. Dar, J. A. Khan, A. A. Lone, S. Langyan, B. Lone, R. Kanth, A. Iqbal, J. Rane, and S. H. Wani, "Identification for surrogate drought tolerance in maize inbred lines utilizing high-throughput phenomics approach," *Plos one*, vol. 16, no. 7, pp. e0254318, 2021.
15. F. Perez-Sanz, P. J. Navarro, and M. Egea-Cortines, "Plant phenomics: An overview of image acquisition technologies and image data analysis algorithms," *GigaScience*, vol. 6, no. 11, pp. gix092, 2017.
16. K. Padmavathi, and K. Thangadurai, "Implementation of RGB and grayscale images in plant leaves disease detection-comparative study," *Indian Journal of Science and Technology*, vol. 9, no. 6, pp. 1-6, 2016.
17. A. Kern, "CCD and CMOS detectors," 2019.
18. S. S. Magazov, "Image recovery on defective pixels of a CMOS and CCD arrays," *Informatsionnye Tekhnologii i Vychislitel'nye Sistemy*, no. 3, pp. 25-40, 2019.
19. D. Defrianto, M. Shiddiq, U. Malik, V. Asyana, and Y. Soerbakti, "Fluorescence spectrum analysis on leaf and fruit using the ImageJ software application," *Science, Technology & Communication Journal*, vol. 3, no. 1, pp. 1-6, 2022.
20. A. F. A. Netto, R. N. Martins, G. S. A. De Souza, F. F. L. Dos Santos, and J. T. F. Rosas, "Evaluation of a low-cost camera for agricultural applications," *J. Exp. Agric. Int*, vol. 32, pp. 1-9, 2019.
21. P. K. R. Maddikunta, S. Hakak, M. Alazab, S. Bhattacharya, T. R. Gadekallu, W. Z. Khan, and Q.-V. Pham, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17608-17619, 2021.
22. J. Trivedi, Y. Shamnani, and R. Gajjar, "Plant leaf disease detection using machine learning." pp. 267-276.
23. H. Wang, S. Shang, D. Wang, X. He, K. Feng, and H. Zhu, "Plant disease detection and classification method based on the optimized lightweight YOLOv5 model," *Agriculture*, vol. 12, no. 7, pp. 931, 2022.
24. P. B. Wakhare, S. Neduncheliyan, and K. R. Thakur, "Study of Disease Identification in Pomegranate Using Leaf Detection Technique." pp. 1-6.
25. B. K. Ekka, and B. S. Behera, "Disease Detection in Plant Leaf Using Image Processing Technique," *International Journal of Progressive Research in Science and Engineering*, vol. 1, no. 4, pp. 151-155, 2020.
26. N. R. Kolhalkar, and V. Krishnan, "Mechatronics system for diagnosis and treatment of major diseases in grape vineyards based on image processing," *Materials Today: Proceedings*, vol. 23, pp. 549-556, 2020.
27. K. Takada, "A Study on Learning Algorithms of Value and Policy Functions in Hex," 北海道大学, 2019.
28. X. Contreras, N. Amberg, A. Davaatseren, A. H. Hansen, J. Sonntag, L. Andersen, T. Bernthaler, C. Streicher, A. Heger, and R. L. Johnson, "A genome-wide library of MADM mice for single-cell genetic mosaic analysis," *Cell reports*, vol. 35, no. 12, pp. 109274, 2021.
29. C. Mazur, J. Ayers, J. Humphrey, G. Hains, and Y. Khmelevsky, "Machine Learning Prediction of Gamer's Private Networks (GPN@ S)." pp. 107-123.
30. V. Vijayalakshmi, and K. Venkatachalapathy, "Comparison of predicting student's performance using machine learning algorithms," *International Journal of Intelligent Systems and Applications*, vol. 11, no. 12, pp. 34, 2019.
31. K. S. Adewole, A. G. Akintola, S. A. Salihu, N. Faruk, and R. G. Jimoh, "Hybrid rule-based model for phishing URLs detection." pp. 119-135.
32. D. Krivoguz, "Validation of landslide susceptibility map using ROC package in R." pp. 188-192.
33. M. Sieber, S. Klar, M. F. Vassiliou, and I. Anastasopoulos, "Robustness of simplified analysis methods for rocking structures on compliant soil," *Earthquake Engineering & Structural Dynamics*, vol. 49, no. 14, pp. 1388-1405, 2020.
34. C. Aybar, Q. Wu, L. Bautista, R. Yali, and A. Barja, "rgee: An R package for interacting with Google Earth Engine," *Journal of Open Source Software*, vol. 5, no. 51, pp. 2272, 2020.
35. M. Schweinberger, "Tree-based models in R," The University of Queensland. Retrieved from <https://slcla.github.io> ..., 2021.
36. S. Pölsterl, "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn," *J. Mach. Learn. Res.*, vol. 21, no. 212, pp. 1-6, 2020.
37. N. Melnykova, R. Kulievych, Y. Vycluk, K. Melnykova, and V. Melnykov, "Anomalies Detecting in Medical Metrics Using Machine Learning Tools," *Procedia Computer Science*, vol. 198, pp. 718-723, 2022.
38. E. J. Gómez-Hernández, P. A. Martínez, B. Peccerillo, S. Bartolini, J. M. García, and G. Bernabé, "Using PHAST to port Caffe library: First experiences and lessons learned," *arXiv preprint arXiv:2005.13076*, 2020.
39. M. N. Gevorkyan, A. V. Demidova, T. S. Demidova, and A. A. Sobolev, "Review and comparative analysis of machine learning libraries for machine learning," *Discrete and Continuous Models and Applied Computational Science*, vol. 27, no. 4, pp. 305-315, 2019.

40. M. Weber, H. Wang, S. Qiao, J. Xie, M. D. Collins, Y. Zhu, L. Yuan, D. Kim, Q. Yu, and D. Cremers, "Deeplab2: A tensorflow library for deep labeling," *arXiv preprint arXiv:2106.09748*, 2021.
41. G. DESHPANDE, "Deceptive security using Python."
42. V. Patel, "Open Source Frameworks for Deep Learning and Machine Learning."
43. A. Pocock, "Tribuo: Machine Learning with Provenance in Java," *arXiv preprint arXiv:2110.03022*, 2021.
44. E. Schubert, and A. Zimek, "ELKI: A large open-source library for data analysis-ELKI Release 0.7. 5" Heidelberg", *arXiv preprint arXiv:1902.03616*, 2019.
45. Y. Liu, X. S. Shan, W. Kan, B. Kong, H. Li, C. Wang, T. Yang, C. Li, Y. Tan, and H. Qu, "JSAT."
46. A. Bhatia, and B. Kaluza, *Machine Learning in Java: Helpful techniques to design, build, and deploy powerful machine learning applications in Java*: Packt Publishing Ltd, 2018.
47. H. Luu, *Beginning Apache Spark 2: with resilient distributed datasets, Spark SQL, structured streaming and Spark machine learning library*: Apress, 2018.
48. M. K. Vanam, B. A. Jiwani, A. Swathi, and V. Madhavi, "High performance machine learning and data science based implementation using Weka," *Materials Today: Proceedings*, 2021.
49. T. Saha, N. Aaraj, N. Ajarapu, and N. K. Jha, "SHARKS: Smart Hacking Approaches for Risk Scanning in Internet-of-Things and cyber-physical systems based on machine learning," *IEEE Transactions on Emerging Topics in Computing*, 2021.
50. R. R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, and S. Zhang, "mlpack 3: a fast, flexible machine learning library," *Journal of Open Source Software*, vol. 3, no. 26, pp. 726, 2018.
51. Z. Wen, J. Shi, Q. Li, B. He, and J. Chen, "ThunderSVM: A fast SVM library on GPUs and CPUs," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 797-801, 2018.
52. K. Kolodiazhnyi, *Hands-On Machine Learning with C++: Build, train, and deploy end-to-end machine learning and deep learning pipelines*: Packt Publishing Ltd, 2020.
53. A. Mohan, A. K. Singh, B. Kumar, and R. Dwivedi, "Review on remote sensing methods for landslide detection using machine and deep learning," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, pp. e3998, 2021.
54. R. Prasad, and V. Rohokale, "Artificial intelligence and machine learning in cyber security," *Cyber Security: The Lifeline of Information and Communication Technology*, pp. 231-247: Springer, 2020.
55. F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, "Segmentation of images by color features: A survey," *Neurocomputing*, vol. 292, pp. 1-27, 2018.
56. A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Computers and electronics in agriculture*, vol. 158, pp. 226-240, 2019.
57. J. Ker, S. P. Singh, Y. Bai, J. Rao, T. Lim, and L. Wang, "Image thresholding improves 3-dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans," *Sensors*, vol. 19, no. 9, pp. 2167, 2019.
58. A. Kumar, and A. Tiwari, "A comparative study of otsu thresholding and k-means algorithm of image segmentation," *Int. J. Eng. Technol. Res.*, vol. 9, pp. 2454-4698, 2019.
59. L. Zhang, L. Zou, C. Wu, J. Jia, and J. Chen, "Method of famous tea sprout identification and segmentation based on improved watershed algorithm," *Computers and Electronics in Agriculture*, vol. 184, pp. 106108, 2021.
60. L. Xie, J. Qi, L. Pan, and S. Wali, "Integrating deep convolutional neural networks with marker-controlled watershed for overlapping nuclei segmentation in histopathology images," *Neurocomputing*, vol. 376, pp. 166-179, 2020.
61. P. M. Anger, L. Prechtel, M. Elsner, R. Niessner, and N. P. Ivleva, "Implementation of an open source algorithm for particle recognition and morphological characterisation for microplastic analysis by means of Raman microspectroscopy," *Analytical Methods*, vol. 11, no. 27, pp. 3483-3489, 2019.
62. S. Jadhav, and B. Garg, "Comparative Analysis of Image Segmentation Techniques for Real Field Crop Images." pp. 1-17.
63. C. Li, X. Zhao, and H. Ru, "GrabCut Algorithm Fusion of Extreme Point Features." pp. 33-38.
64. J. F. Randrianasoa, C. Kurtz, E. Desjardin, and N. Passat, "AGAT: Building and evaluating binary partition trees for image segmentation," *SoftwareX*, vol. 16, pp. 100855, 2021.
65. N. Zhu, X. Liu, Z. Liu, K. Hu, Y. Wang, J. Tan, M. Huang, Q. Zhu, X. Ji, and Y. Jiang, "Deep learning for smart agriculture: Concepts, tools, applications, and opportunities," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 32-44, 2018.
66. Q. Zhang, Y. Liu, C. Gong, Y. Chen, and H. Yu, "Applications of deep learning for dense scenes analysis in agriculture: A review," *Sensors*, vol. 20, no. 5, pp. 1520, 2020.
67. D. Ileri, E. Belal, C. Okinda, N. Makange, and C. Ji, "A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing," *Artificial Intelligence in Agriculture*, vol. 2, pp. 28-37, 2019.
68. S. Singh¹, and P. P. Kaur, "A study of geometric features extraction from plant leaves," 2019.

69. M. Martsepp, T. Laas, K. Laas, J. Priimets, S. Tökke, and V. Mikli, "Dependence of multifractal analysis parameters on the darkness of a processed image," *Chaos, Solitons & Fractals*, vol. 156, pp. 111811, 2022.
70. J. M. Ponce, A. Aquino, and J. M. Andújar, "Olive-fruit variety classification by means of image processing and convolutional neural networks," *IEEE Access*, vol. 7, pp. 147629-147641, 2019.
71. N. R. Bhimte, and V. Thool, "Diseases detection of cotton leaf spot using image processing and SVM classifier." pp. 340-344.
72. G. K. Sandhu, and R. Kaur, "Plant disease detection techniques: a review." pp. 34-38.
73. P. Alagumariappan, N. J. Dewan, G. N. Muthukrishnan, B. K. B. Raju, R. A. A. Bilal, and V. Sankaran, "Intelligent plant disease identification system using Machine Learning," *Engineering Proceedings*, vol. 2, no. 1, pp. 49, 2020.
74. A. A. Bharate, and M. Shirdhonkar, "Classification of grape leaves using KNN and SVM classifiers." pp. 745-749.
75. S. Sivasakthi, and M. MCA, "Plant leaf disease identification using image processing and svm, ann classifier methods." pp. 30-31.
76. C. U. Kumari, S. J. Prasad, and G. Mounika, "Leaf disease detection: feature extraction with K-means clustering and classification with ANN." pp. 1095-1098.
77. M. Vaishnave, K. S. Devi, P. Srinivasan, and G. A. P. Jothi, "Detection and classification of groundnut leaf diseases using KNN classifier." pp. 1-5.
78. E. Hossain, M. F. Hossain, and M. A. Rahaman, "A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier." pp. 1-6.
79. J. Singh, and H. Kaur, "Plant disease detection based on region-based segmentation and KNN classifier." pp. 1667-1675.
80. A. Bakhshipour, and H. Zareiforoush, "Development of a fuzzy model for differentiating peanut plant from broadleaf weeds using image features," *Plant methods*, vol. 16, no. 1, pp. 1-16, 2020.
81. H. Sabrol, and S. Kumar, "Plant leaf disease detection using adaptive neuro-fuzzy classification." pp. 434-443.
82. P. Sutha, A. Nandhu Kishore, V. Jayanthi, A. Periyanan, and P. Vahima, "Plant Disease Detection Using Fuzzy Classification," *Annals of the Romanian Society for Cell Biology*, pp. 9430-9441, 2021.
83. K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize leaf disease detection and classification using machine learning algorithms," *Progress in Computing, Analytics and Networking*, pp. 659-669: Springer, 2020.
84. Y. Kashyap, and S. Shrivastava, "ROLE OF VARIOUS SEGMENTATION AND CLASSIFICATION ALGORITHMS IN PLANT DISEASE DETECTION."
85. A. Dwari, A. Tarasia, A. Jena, S. Sarkar, S. K. Jena, and S. Sahoo, "Smart Solution for Leaf Disease and Crop Health Detection," *Advances in Intelligent Computing and Communication*, pp. 231-241: Springer, 2021.
86. N. K. KUMAR, and A. PRATHYUSHA, "SOLANUM TUBEROSUM LEAF DISEASES DETECTION USING DEEP LEARNING."
87. S. Hossain, R. M. Mou, M. M. Hasan, S. Chakraborty, and M. A. Razzak, "Recognition and detection of tea leaf's diseases using support vector machine." pp. 150-154.
88. S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Computers in Industry*, vol. 108, pp. 115-120, 2019.
89. N. Cherukuri, G. R. Kumar, O. Gandhi, V. S. K. Thotakura, D. NagaMani, and C. Z. Basha, "Automated Classification of rice leaf disease using Deep Learning Approach." pp. 1206-1210.
90. E. Khalili, S. Kouchaki, S. Ramazi, and F. Ghanati, "Machine learning techniques for soybean charcoal rot disease prediction," *Frontiers in plant science*, vol. 11, pp. 590529, 2020.
91. M. P. Patil, "Nearest neighborhood approach for identification of cucumber mosaic virus infection and yellow sigatoka disease on banana plants," 2022.
92. H. Orchi, M. Sadik, and M. Khaldoun, "On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey," *Agriculture*, vol. 12, no. 1, pp. 9, 2021.
93. D. Zhang, X. Zhou, J. Zhang, Y. Lan, C. Xu, and D. Liang, "Detection of rice sheath blight using an unmanned aerial system with high-resolution color and multispectral imaging," *PloS one*, vol. 13, no. 5, pp. e0187470, 2018.
94. G. Yashodha, and D. Shalini, "An integrated approach for predicting and broadcasting tea leaf disease at early stage using IoT with machine learning—a review," *Materials Today: Proceedings*, vol. 37, pp. 484-488, 2021.
95. A. V. Zubler, and J.-Y. Yoon, "Proximal methods for plant stress detection using optical sensors and machine learning," *Biosensors*, vol. 10, no. 12, pp. 193, 2020.
96. B. L. Thushara, and T. M. Rasool, "Analysis of plant diseases using expectation maximization detection with BP-ANN classification," *XIII (VIII)*, 2020.
97. P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Expert Systems with Applications*, pp. 118117, 2022.

98. A. I. Khan, S. Quadri, and S. Banday, "Deep learning for apple diseases: classification and identification," *International Journal of Computational Intelligence Studies*, vol. 10, no. 1, pp. 1-12, 2021.
99. A. J. Das, R. Ravinath, T. Usha, B. S. Rohith, H. Ekambaram, M. K. Prasannakumar, N. Ramesh, and S. K. Middha, "Microbiome Analysis of the Rhizosphere from Wilt Infected Pomegranate Reveals Complex Adaptations in Fusarium—A Preliminary Study," *Agriculture*, vol. 11, no. 9, pp. 831, 2021.
100. S. S. Gaikwad, "Fungi classification using convolution neural network," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 4563-4569, 2021.
101. D. Priya, "Cotton leaf disease detection using Faster R-CNN with Region Proposal Network," *International Journal of Biology and Biomedicine*, vol. 6, 2021.
102. B. M. Joshi, and H. Bhavsar, "Plant leaf disease detection and control: A survey," *Journal of Information and Optimization Sciences*, vol. 41, no. 2, pp. 475-487, 2020.
103. G. Gangadevi, and C. Jayakumar, "Review of Classifiers Used for Identification and Classification of Plant Leaf Diseases." pp. 445-459.
104. V. Vučić, M. Grabež, A. Trchounian, and A. Arsić, "Composition and potential health benefits of pomegranate: a review," *Current pharmaceutical design*, vol. 25, no. 16, pp. 1817-1827, 2019.
105. V. Sahni, S. Srivastava, and R. Khan, "Modelling techniques to improve the quality of food using artificial intelligence," *Journal of Food Quality*, vol. 2021, 2021.
106. S. Patidar, A. Pandey, B. A. Shirish, and A. Sriram, "Rice plant disease detection and classification using deep residual learning." pp. 278-293.
107. M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Computers and electronics in agriculture*, vol. 150, pp. 220-234, 2018.
108. T. Hayit, H. Erbay, F. Varçın, F. Hayit, and N. Akci, "Determination of the severity level of yellow rust disease in wheat by using convolutional neural networks," *Journal of Plant Pathology*, vol. 103, no. 3, pp. 923-934, 2021.
109. S. S. Jasim, and A. A. M. Al-Taei, "A Comparison Between SVM and K-NN for classification of Plant Diseases," *Diyala Journal for Pure Science*, vol. 14, no. 2, pp. 94-105, 2018.
110. P. DAYANG, and A. S. K. MELI, "AS: Evaluation of image segmentation algorithms for plant disease detection," *Int. J. Image Graph. Signal Process.(IJIGSP)*, vol. 13, no. 5, pp. 14-26, 2021.
111. M. Agarwal, S. K. Gupta, and K. Biswas, "Development of Efficient CNN model for Tomato crop disease identification," *Sustainable Computing: Informatics and Systems*, vol. 28, pp. 100407, 2020.
112. R. D. Devi, S. A. Nandhini, R. Hemalatha, and S. Radha, "IoT enabled efficient detection and classification of plant diseases for agricultural applications." pp. 447-451.
113. S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant Leaf Disease Detection using Computer Vision and Machine Learning Algorithms," *Global Transitions Proceedings*, 2022.
114. H. Altıparmak, M. Al Shahadat, E. Kiani, and K. Dimililer, "Fuzzy classification for strawberry diseases-infection using machine vision and soft-computing techniques." pp. 429-436.
115. M. Toseef, and M. J. Khan, "An intelligent mobile application for diagnosis of crop diseases in Pakistan using fuzzy inference system," *Computers and Electronics in Agriculture*, vol. 153, pp. 1-11, 2018.

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