

Institute of Technology, Carlow

Computing

Course: BSc (Hons) Software Development Year 4

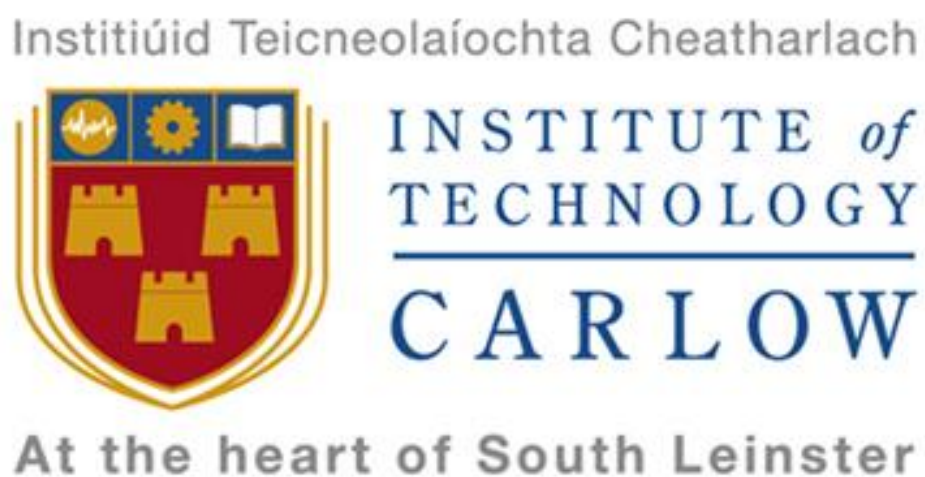
Author: Eamonn Gaynor

Student ID: C00197458

Tutor: Mr. Nigel Whyte

Document: Research Document

Plant Disease Identification Application



Abstract

Plant diseases and pests cause the loss of up to 40 % of the annual harvests, a challenge for a growing world population. (The Crop Site, 2017). Dependable, accurate and precise examinations of plant disease is crucial in predicting yield loss, monitoring disease resistance and preventing epidemic outbreak. Assessments of plant disease which are inaccurate may lead to a misleading diagnosis being drawn from the examination, thus leading to poor disease management and/or loss of the plant itself. (Bock et al., 2010). Smart farming, which is the application of modern Information and Communication Technologies into agriculture, now called the *Third Green Revolution* can have a huge impact on this area of farming, thus reducing the burden for farmers globally. (Smart-AKIS, 2017)

Scientists at EnviroCORE research innovative environmental technologies along with biotechnologies which enhance the economic and social development in an environmentally friendly manner. (Envirocore, 2017). Currently at IT Carlow, scientists at EnviroCORE carry out visual inspections of plants leaves and make subjective opinions on the overall health of the plant. The aim of this project is to create an application for smart devices which will allow users to identify plant disease through the means of artificial intelligence using computer vision techniques. The goal is to automate visual inspection and eliminate the ambiguity associated with a diagnosis, in turn reducing the subjectivity of the process. By using JAVA as the core programming language and OpenCV as an open source library for vision techniques the application will be produced.

Contents

Abstract.....	2
Literature Review	4
Title: Guidelines for Identification and Management of Plant Disease	4
Title: Image Processing, Analysis, and Machine Vision	4
Title: Using Deep Learning for Image-Based Plant Disease Detection.....	5
Technologies	6
Computer Vision and OpenCV	6
1. Canny Edge Detection.....	7
2. Hough Transform.....	7
3. Thresholding	8
Present Applications.....	9
Plantix – PEAT.....	9
Crop Diagnosis – Ergobyte	10
Plant disease web application – MATLAB.....	11
References.....	12

Literature Review

Title: Guidelines for Identification and Management of Plant Disease

By: Ken Pernezny, Monica Elliott, Aaron Palmateer, and Nikol Havranek

The article “Signs and symptoms of plant disease” written by Ken Pernezny, Monica Elliott, Aaron Palmateer, and Nikol Havranek explores the diagnosis methods behind plant disease created by microorganisms including bacteria, viruses, and primarily fungi.

Of all pathogens concerned with the biology of microorganisms, 85% of the plant disease is caused by fungi, of which can be seen without the use of a microscope. Fungi include spore producing organisms such as mushrooms, moulds and yeast. Fungi can disperse from host to host very quickly, as wind can carry fungi miles. Water, rainfall and irrigation can also carry spores from one plant to the next. Fungi being highly contagious, may also travel through the means of entangled roots, where disease can attach itself and travel upwards through the soil. (Ken Pernezny, 2017)

Symptoms, described as abnormal features in the plant, where it may indicate there some sort of disease. Obvious features included spots, the technical term being “lesion”, which indicates a localized diseased area or wound. As these lesions cultivate, the symptom can then be described as blight, whereby a much larger area of the plant leaf/tissue is affected by the disease. More obvious and severe symptoms can be found on the stems and roots, where masses of blights, much like a tumour found in humans can be found.

The information found in this publication can be used throughout this project, as the signs and symptoms associated with discovering plant disease, can be used in conjunction with computer vision and transformed into a mobile application to improve accuracy and accountability.

Title: Image Processing, Analysis, and Machine Vision

By: Milan Sonka, Vaclav Hlavac, Roger Boyle

Milan Sonka et al. describes computer vision, as a high-level computer replicating the human eye, and obtaining the ability to decide based on the information provided from the image. For human observers, a lot of prior knowledge is used in interpreting images, machines only begin with an array of numbers in which it begins to make identifications with and draws conclusions based on the data it retrieves. They point out that, with the analyzation of big data becoming more and more apparent in today’s world, the ability for machine to perform computer vision techniques on large data sets will be crucial.

There are two types of image processing, low-level processing and high-level computer vision. Low-level comprises of data which is comprised of original images by matrices composed of brightness values, whereas high-level data represents knowledge about the image, e.g. the object size, shape, and relations between the different objects in the image.

The first step is described as an image capture, such as a camera or an image upload. From there the computer suppresses the noise, known as pre-processing, whereby some features maybe enhanced which are relevant to understanding the image. In this projects case for example, the edge of the leaf would be determined using edge extraction. The next step being image segmentation, where the computer will separate various objects from the image. There are two types of image segmentation, total and partial, total being where dark non-touching objects can be completely removed from a white background, commonly used where they user might want to remove printed text. In general, partial segmentation is used for more complicated problems. (Sonka, Hlaváč and Boyle, n.d.)

Title: *Using Deep Learning for Image-Based Plant Disease Detection*

By: Sharada Prasanna Mohanty, David Hughes and Marcel Salathé

‘Using Deep Learning for Image-Based Plant Disease Detection’ written by Sharada Prasanna Mohanty, David Hughes and Marcel Salathé explain how working with modern technology, in particular computer vision with mobile applications can improve food security in relation to crop and plant disease.

By using a public dataset of 50,000+ images of diseased and healthy plants under collection by controlled conditions, they trained a deep neural network which can identify up to 14 different crop species and over 26 diseases. The model has achieved a 99.35% accuracy with its diagnosis, it is there understanding that with these deep learning models created on an ever-increasing image dataset, it presents a clear future for the use of smartphone-assisted plant disease diagnosis, but on a much larger scale.

The extract explains how food security remains a high threat by several factors, including climate change, a decline in pollinators, but most notably plant disease. Not only is food crucial on a global scale to meet the demand of over 7 billion people, but for the livelihood of small farmer. Small farms rely heavily on the production of healthy crops, where plant disease could have a major impact on their main source of income. In the developing world, 80% of all agricultural production is produced by these small farms. The article describes how mobile phones have now taken advantage of the latest technology available, and use its incredible computing power with high definition displays and cameras for the use in smart farming. They predict that by 2020, there will be 5 billion plus mobiles on the planet, with over 69% of users. Sharada Prasanna Mohanty (2016)

Technologies

This project will include various technologies to produce the mobile application. The main body of the application will be written predominantly in the coding language, JAVA. The main reasoning behind this, is for the ability to incorporate OpenCV. OpenCV is an open source computer vision library for programming functions, it is used in conjunction C++, JAVA and Python. As this application is essentially being used for visual inspections, it is imperative that the artificial intelligence for image processing techniques is of high quality. Computer vision are methods in which an image is captured, transformed to digital format in which functional operations can be performed to retrieve useful data or to produce an enhanced format of the image. Image processing comprises usually of three steps, importation of the image via upload or digital capture, analyzation and manipulation of the data consisting of data compression, pattern spotting and enhancements. The image can then be outputted in an altered format or to show data captured from the image processing in a reported format. There are many advantages to using image processing, the main purposes being; image visualization, where hidden objects are made visible to the naked eye. The sharpening of a blurred image, image retrieval, pattern measurement and image recognition. Throughout this project, the application will use image recognition to distinguish the patterns associated with plant disease, and pattern measurement to decipher the plants edges and gather measurement.

The functions which will be included are edge detection technology, binary image processing of both the original image and of clustered rendition, segmentations and fuzzy logic techniques.

The application will also allow the user to open, capture and send an image. The functions for the camera technology will also be written in JAVA. The captured images can then be sent to an SQL cloud storage, where the image processing can be executed. The images stored in the cloud may also be retrieved from the user, for both monitoring use and to build reports from previous data captures. To allow for this, a login system will be designed, where the user can store data and images via a secure account setup.

Computer Vision and OpenCV

Computer Vision is defined as being an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos, it designed to interpret what the human visual system does. It works in a way that it extracts high dimensional data from real world pictures, and transforms this data to produce numerical or symbolic information, which can be used to determine decisions. Computer Vision is used in so many different formats, the replication of the input from human retina means that it can translate these images into descriptions which can ultimately process thoughts and produce appropriate actions. It was Larry Roberts, who while writing his PHD, described the possibilities of extracting 3D geometrical information from simple 2D blocks, many researchers of artificial intelligence followed this theory and studied computer vision.

One of the main computer vision libraries used today is called OpenCV, which is a cross platform, open-source library. OpenCV is written predominately in C++, but can also be used with JAVA, Python, and C. It was originally developed by Intel, and is not being maintained by Willow Garage and ItSeez. Well known functions being used in today's world by OpenCV, comprise of facial recognition, identifying objects, and augmented reality, such as VR headsets. OpenCV can handle many different demonstrations, which include derivation, contours, transforms, histograms, segmentations, image processing, cascade detectors and 3D processing. Below are some functions within these demonstrations that are of interest.

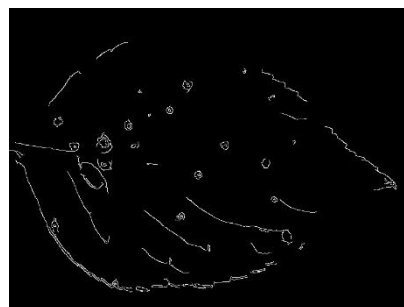
1. Canny Edge Detection

Canny edge detection is an edge detection algorithm, used widely in OpenCV, and developed by John F. Canny. The algorithm is multi-stage, in which it goes through four key stages.

The first of these stages is noise reduction, whereby the noise in the image is removed using a 5x5 Gaussian filter. A Gaussian filter, also known as Gaussian smoothing is blurring an image to reduce both image noise and detail. The second stage is to find the intensity gradient of the image. The smoothed image prior is filtered with a Sobel kernel in both the horizontal and vertical direction to get both derivatives, from there the edge gradient can be found for each pixel. The third step is non-maximum suppression, whereby a full scan of the image is done to remove any unwanted pixels, which do not make up that of the edge. The last stage is Hysteresis Thresholding, this decides what is an edge and what is not. They do so by comparing the gradients found in the prior steps, and compare them using a minVal and maxVal. Any gradients found above or below those values, are defined as not being an edge.



Canny Edge Detection Leaf Original



Canny Edge Detection Leaf

2. Hough Transform

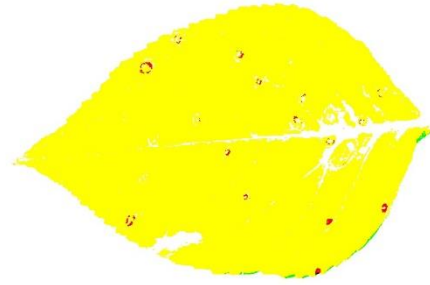
The Hough Transform is used to isolate features shapes found in image. It is commonly used in determining regular shapes in an image, such as circles, ellipses lines, rectangles etc. A particularly useful feature whereby the user wishes to identify a number of instances of a particular shape or occurrence in an image.

3. Thresholding

Thresholding, one of the simplest forms of image segmentation, is whereby an image in greyscale is transformed into a binary image. The function creates a fixed constant of image intensity, where the function replaces the pixel with a black pixel if it is set to be either above or below. Thresholding can be categorized into six different groups, from which the computer will automatically decide on which to use. These groups are histogram shape, clustering, entropy, object attribute, spatial and local.



Thresholding Leaf Original



Thresholding Leaf

Present Applications

There are applications both on the market and off the market which are not unlike the scope of this project. Some applications held more interesting features than others, with some applications lacking key components. With regards to design, the applications are similar, with most features and colours resembling that of one another, with a view on simplicity and ease of use.

Aside from mobile applications, web applications have also been created with a comparable objective, but without the use of image capturing via camera. Explored below are two applications in which both have very useful features which relate to this project, and a web application which incorporates various functions which will be implemented in the mobile application.

Plantix – PEAT

Plantix, an application produced by German company PEAT, who work with latest cutting edging technology to support and secure harvests, and pride themselves to ensure a safe food production worldwide.

The application produced features various functions which this project aims to replicate in a similar manner. The design is smooth, with uncomplicated operations during use. As seen below in (fig 1.1 Landing Page), the applications home page is populated with a ‘latest image’ section, where a user may return to the last disease recorded, and from which plant the diagnosis was captured. Similar to the ‘latest image’ function, there is also a ‘Library – latest visits’ bookmark, where the user may retrieve the list of diagnosed diseases, while also recording the timestamps from which the disease materialized for monitoring purposes.

Also located on the home screen, and throughout the application, is a local weather feature. The function holds two main purposes in this application, which are to determine weather conditions, and to determine location. The weather report is used in the diagnosis process, where variable conditions such as precipitation and temperature are considered when delivering the final result. The location is used to decipher whether the resulted disease is common in that area of the world. Vital information, as such results could lead to the discovery of a migrated disease from another country. The product allows the user to capture an image from the built-in camera, or to upload a photo, in which the second screen is produced as seen in (fig 1.2 Diagnosis). All images are stored in the cloud, where a user may log into their account via Facebook, Pinterest etc. to retrieve the uploads.

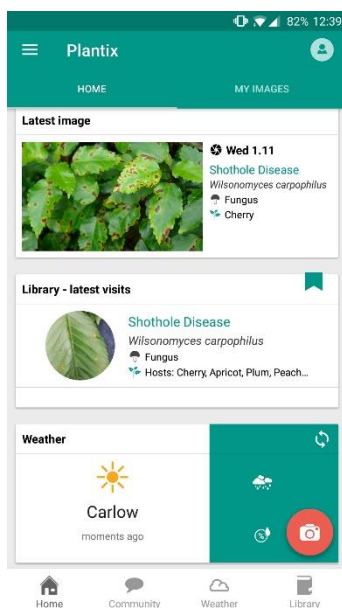


Fig. 1.1 Landing Page

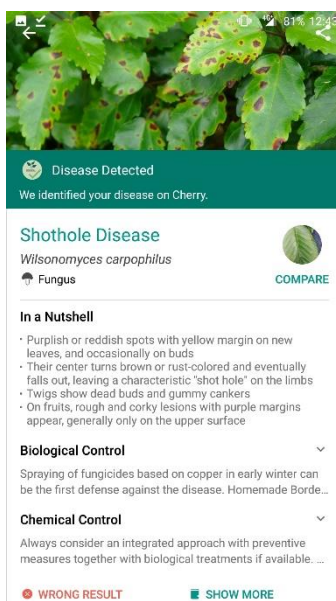


Fig.1.2 Diagnosis

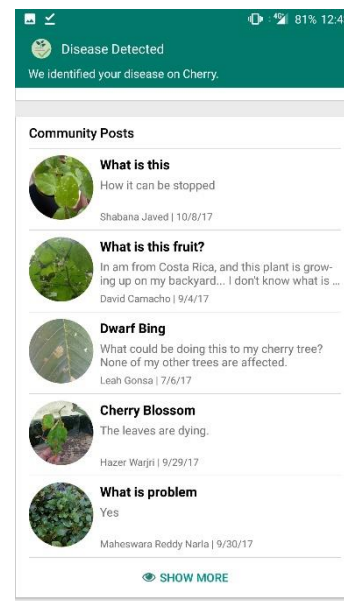


Fig.1.3 Mitigation

In (fig.1.2 Diagnosis), the user is presented with the diagnosis from the upload. An extremely detailed description of the disease is presented, with accompanying mitigation plans under biological and chemical control. The user may also choose to select ‘Wrong result’ where the information is stored for future diagnostics, and to improve the applications accuracy over time. An interesting feature presented on this screen, is the compare function. Where the user can compare their infected plant to similar images posted from different users who encountered the same disease. Logged in users may post their findings, mitigations and thoughts on a community forum as seen in (fig.1.3 Mitigation). A unique feature which has not been seen in other plant disease applications.

What has been found to be missing from this application, is the output of the scientific reasoning behind the diagnosis. Key information such as plant/leaf coverage, where the user is presented with a percentage of how much of the leaf is infected. Important data which may be used in the monitoring of the disease, as the researcher could track the disease, to determine whether it is spreading or is responding well to treatment.

Crop Diagnosis – Ergobyte

Crop diagnosis is an application developed by Ergobyte, a Greek software company who develop and commercialize software solutions for health professionals and clinics. This application deciphers plant disease not through image processing, but through the means of a multiple questions. The design of the application compared to Plantix, is far less complex, with very little extra features. However, the application can still achieve its objective in a different manner, one in which this project aims to incorporate as a tool to improve accuracy for the overall result.

As seen in (fig 2.1 Landing page), the user is presented with the home page, with multiple pages located on a tab to the left. These pages include a log in, a diagnosis library, history, and an about. To diagnose the disease, the user is first off asked to select the base design of the plant, as seen varying from Tomato, Olive, Pepper etc. From there, as example seen in (fig 2.2 Map) and (fig 2.3 Questions), the user is presented with multiple questions varying from Yes/No to tick the box. As also seen in Plantix, location (fig 2.2 Map) plays a key role in determining what the outcome, where the application has a built-in geo location function. Other questions varying from farming method, stem discoloration, and under which environmental control the plant has been kept. Accompanying these questions, are image examples which give the user a good indication as to what they are to look out for, an interesting feature. Upon completing the form, the user is then presented with the diagnosis as seen in (fig. 2.4 Diagnosis), with a percentage of how accurate they believe the result to be.

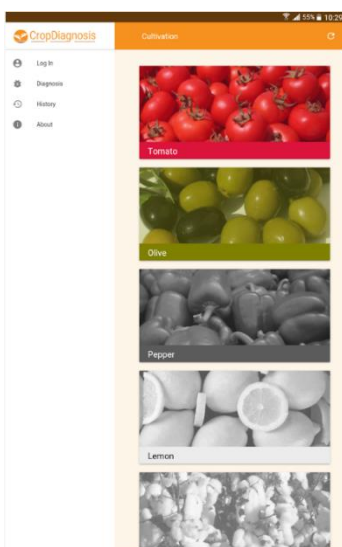


Fig.2.1 Landing Page

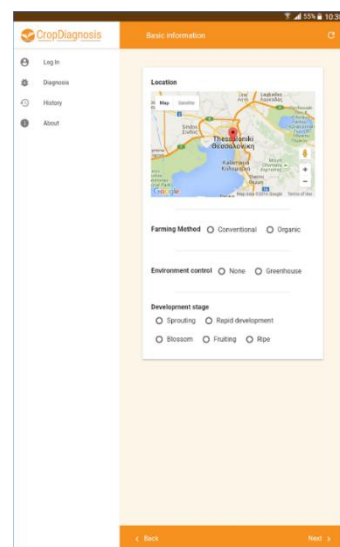


Fig.2.2 Map

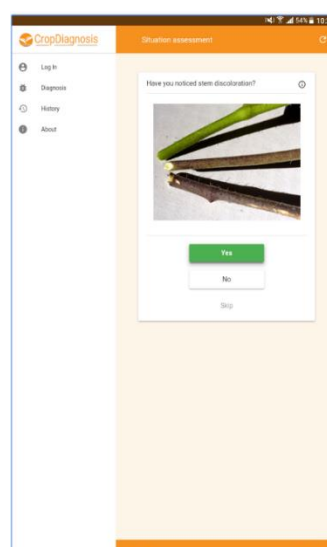


Fig.2.3 Questions

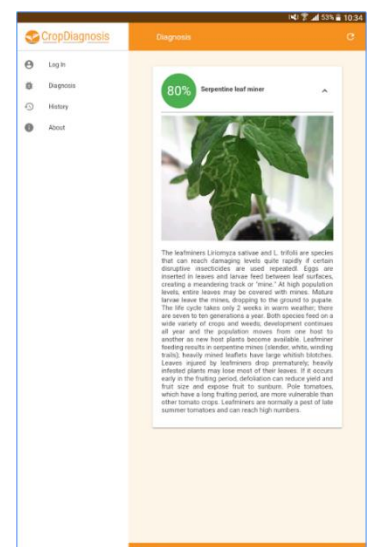


Fig.2.4 Diagnosis

Although this application achieves its intended purpose via the means of cancellation, it could be much improved regarding accuracy with image processing as support. This project will use this cancellation method in conjunction with image processing, which will ultimately improve reliability and accuracy.

Plant disease web application – MATLAB

MATLAB is a high-performance language for technical computing. Mat-lab integrates visualization for programming in a user-friendly format with an emphasis on mathematical notation. The project below, as seen in (fig. 3.1 Bacterial Spot) and (fig 3.2 Percentage), although it is in a web application form, it takes on many functions in which this project will envisage to implement in mobile format. The MATLAB project was written by a MATLAB user, upon researching technologies used, image processing was completed using algorithms such as canny edge detection, fuzzy logic, and K-Means segmentations. In this project, algorithms such as these will be utilized in JAVA for the mobile application. As seen below in (fig 3.1 Bacterial Spot), the user uploads an image, where image processing is performed in various formats. The application initially creates a binary copy of the image, via binarization also known as Thresholding, is a method found in OpenCV. Binarization is the process of converting a pixel image to a binary image, where there are only two possible coloured pixels. Binarization was initially invented for the use of sending faxes, but today is more prominently used in image segmentations.

Once a binary image is captured, the user can then perform K-means segmentation on the image. K-means clustering is used when you have unlabelled data (data without defined categories or sets). The algorithms objective is to assign unlabelled data to K, into which conclusions or results can be drawn. (Trevino, 2017). In this example, the algorithm is used to label the blotches of the plant created by disease.

The application uses fuzzy logic to determine as what percentage of the image is actual plant, and to which is part of the background. Fuzzy logic is whereby there are only two possible Boolean values, true or false (0 or 1). By the combination of these functions, the program can output (fig. 3.2 Percentage) to the user, a percentage of the plant in which by it is infected. This type of output is what this project will be incorporating in the mobile application, as well as describing the type of disease in which it has discovered.

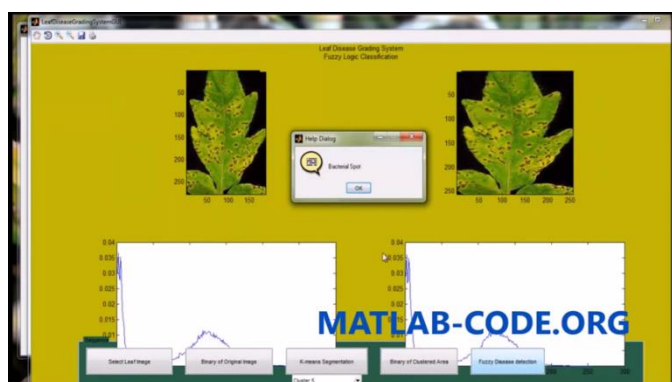


Fig.3.1 Bacterial Spot

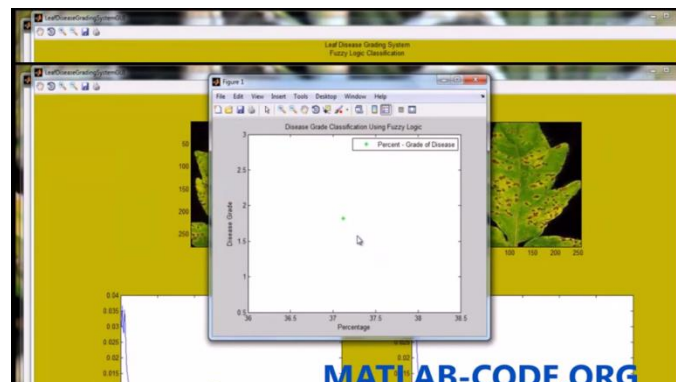


Fig.3.2 Percentage

References

- **(Bock et al., 2010).** Bock, C., Poole, G., Parker, P. and Gottwald, T. (2010). Plant Disease Severity Estimated Visually, by Digital Photography and Image Analysis, and by Hyperspectral Imaging. *Critical Reviews in Plant Sciences*, 29(2), pp.59-107.
- **(Envirocore., 2017).** Research - Envirocore. [online] Available at: <http://www.envirocore.ie/research/> [Accessed 4 Nov. 2017].
- **Smart-AKIS. (2017).** *What is Smart Farming? - Smart-AKIS.* [online] Available at: <https://www.smart-akis.com/index.php/network/what-is-smart-farming/> [Accessed 5 Nov. 2017].
- **The Crop Site. (2017).** *Almost 40 Per Cent of Worldwide Crops Lost to Diseases.* [online] Available at: <http://www.thecropsite.com/articles/1202/almost-40-per-cent-of-worldwide-crops-lost-to-diseases/> [Accessed 5 Nov. 2017].
- **Trevino, A. (2017).** *Introduction to K-means Clustering.* [online] Datascience.com. Available at: <https://www.datascience.com/blog/k-means-clustering> [Accessed 6 Nov. 2017].
- **Ken Pernezny, a. (2017).** *PP249/MG442: Guidelines for Identification and Management of Plant Disease Problems: Part II. Diagnosing Plant Diseases Caused by Fungi, Bacteria and Viruses.* [online] Edis.ifas.ufl.edu. Available at: <http://edis.ifas.ufl.edu/mg442> [Accessed 8 Nov. 2017].
- **Sonka, M., Hlaváč, V. and Boyle, R. (n.d.).** Sonka, M., Hlaváč, V. and Boyle, R. *Image processing, Analysis, and Machine Vision.* 4th ed. Timothy L. Anderson, pp.1-9.
- **Sharada Prasanna Mohanty (2016)** Sharada Prasanna Mohanty, David Hughes, and Marcel Salathé. *Using Deep Learning for Image-Based Plant Disease Detection.*