DETECTION OF AFFECTED AREA, PESTS AND CLASSIFICATION OF PESTS USING CONVOLUTIONAL NEURAL NETWORKS FROM THE LEAF IMAGES

Mr. Alagiah Suthakaran

169336L

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka, for the partial fulfillment of the requirements of Degree of Master of Science in Information Technology

March 2019

DECLARATION

I, Alagiah Suthakaran declare that the report entitled “Detection of affected area, pests and classification of pests using convolutional neural networks from the leaf images” and the work presented in it are done by myself. I confirm that:

- This work was done wholly or mainly while candidate for a master degree at this University;
- Where any part of this report has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated;
- Where I have quoted from the work of others, the source is always given;
- I have acknowledged all main sources of help.

Signature: .... Date : ....

Supervised by:

Mr. Saminda Premarathne,

Senior Lecturer,

Faculty of Information Technology,

University of Moratuwa.

Signature: .... Date : ....

ACKNOWLEDGMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

First and foremost, I would like to thank to my supervisor of this project, Mr.S.C.Premaratne, Senior Lecturer, Department of Information Technology, Faculty of Information Technology, University of Moratuwa for the valuable guidance and advice. He inspired me greatly to work in this project. His willingness to motivate me contributed tremendously to this project. In addition, I would like to thank all the faculty members of the Faculty of Information and Technology who shared their vast knowledge during these two years by providing me with a good environment that greatly influenced the achievement of this goal.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends for their help and wishes for the successful completion of this project.

Thanks,

A.Suthakaran

ABSTRACT

Pest infection is the most important problem on vegetable plants. One way to control the pest infection is to use proper pesticides. Early detection of the pest or the initial presence of pests is a key element for crop protection. The identification of the pest was done manually at the beginning. This takes time and also requires ongoing monitoring of experts. An automatic pest detection system is needed to examine the infestation and classify the type of pest. Today, there are many techniques and methods for identifying pests and detecting plant diseases. In these techniques, image processing techniques are very efficient and reliable. First, the proposed model detects whether the leaf is affected or not and calculates the affected area in the image. Next, the region of the detected pest and classification were performed using convolutional neural networks. The severity of the infection can be observed by calculating the percentage of the affected area, which leads to taking the appropriate measures.

Keywords: segmentation, region of interest, pest detection, convolution neural networks

Table of Contents

List of Figures/Tables

CHAPTER 01

Introduction

1.1 Background

Agriculture is one of the main economic activities of Batticaloa District in the eastern province of Sri Lanka. More than 60% of the population depends mainly on agriculture. It is therefore important to increase crop productivity. Different vegetable crops are grown in the Batticaloa region. Such as ladies' fingers, tomato, long beans, brinjal, squash, chilli and onion. But one of the most important problems at the moment is “pest infection” on vegetable plants. The identification of the leaves of harmful or affected vegetable plants, the symptoms of the pests play a key role in the successful cultivation of crops. Pesticides are the only way to stop the effects of these pests. Pesticides will remove some species of pests. And more pesticide use is very destructive for crops, air, soil, water resources and animals that come in contact with pesticides. The use of pesticides decreases the overall biodiversity of the soil. In our research, we focus on early detection of pests. The images are acquired using a camera and also obtained from imagenet.org and other Internet resources. Then, the images must be preprocessed in order to retrieve the image content by the image processing methods.

This study describes the automatic detection of leaf affectedness and determines the type of presence of the pest on various plant leaf images. Then comes a simple and effective method based on image processing using the convolutional neural network for the classification of pests.

In this respect, it is difficult to recognize the vegetable field pests due to the strong articulation of pests, the size and color of the pests, and the visual difficulty of some pests, despite the structure of the pests. The task becomes more difficult when pests are to be recognized from still images using an automated system. Images of an insect pest can be taken from different viewpoints, on a congested background, or can undergo transformations such as rotation, noise, etc. It is therefore likely that two images of the same pest are different and that intra-pest variations make it difficult to detect the type of pest [2]. To meet these challenges, we must propose an effective segmentation method. Image segmentation is the most important work in many image processing systems such as pattern recognition, image retrieval and reduced surveillance. The result of segmentation is mainly used for the understanding of image content and the recognition of visual entities via the identification of the region of interest [13]. In this study, a color transformation technique used for image pre-processing and an adaptive thresholding method is applied to separate the leaf region in the presence of a pest. And the region properties considered to identify the affected area and the pest region. The method is implemented in MATLAB 2014a.

1.2 Research Problem Statement

Pest detection is a major role of vegetable cultivation in agriculture. In order to increase the rate of production, it is necessary to detect the pests earlier. As a result, the development of an automated model for identifying pests in the vegetable garden is of great importance. Image processing techniques are of great importance for the automatic identification of pest images.

1.3 Aim and Objectives

1.3.1 Aim

The purpose of this research is to interpret the image for the detection of pests, then to classify pests using the convolutional neural network among five types of pests. In addition, this study will suggest the appropriate pesticides for the pests detected from the image.

1.3.2 Objectives

The main objectives of this research are:

- To acquire the images from vegetable gardens as data set.
- To identify the number of classes of insect pests from the dataset.
- To develop a model to detect the leaf is affected or not by the pests, considering the texture properties.
- To identify the suitable segmentation method for the region of interest in pest detection.
- To build a model for classification algorithm and configure that for the recognition.

- To evaluate and analyse the results from the result of classification.

1.4 Overview of the report

This chapter contains the background and the introduction for the problem statement. The next chapter describes about similar works done by others related to the problem domain. Chapter three describes, how Convolutional neural networks works and the structure of the internal architecture. Then, chapter four elaborates the proposed methodology and the outcome of the stepwise processes. On the next chapter, the experimental setup for the classification part is explained and also the results displayed based on the experiments. Finally, chapter six presents the discussion and conclusion based on the classification results and includes the future works based on our research work.

1.5 Summary

This research work has been conducted to propose a methodology for affected region detection, pest detection and the classification of pest. In this chapter, we explained the need for pest detection in agriculture field and about the aim and objectives for the study.

CHAPTER 02

Review of literature related to pest detection and pest classification

2.1 Introduction

This chapter describes the types of segmentation methods to detect the objects given in the leaf images, the feature extraction methods for various classification methods to classify the pests and the experimental setup for the classification experiments. Furthermore, this chapter summarizes the drawbacks of the methods followed by the authors on the problem domain.

2.2 Related works on the problem domain

In [2], the authors developed a framework for classifying insect pest images for paddy fields using gradient-based features using the bag-of-words method. Images of twenty classes of pests for paddy fields were obtained from Google Images and photographs taken by the Faculty of Agriculture, Jaffna University, Sri Lanka. The images were then classified via the system involving the identification of regions of interest and the representation of these regions in the form of scaled invariant feature transform descriptors (SIFTs) or speeded-up robust feature descriptors (SURFs), constructing codebooks that allow descriptors to be a fixed-length vector in the histogram space, and multi-class classification of feature histograms using support vector machines (SVMs). In addition, the histograms of the oriented gradient descriptors (HOG) were applied in the classification. As a basic classifier, the nearest neighbor approach was used and compared to SVM-based classifiers. Their experimental results in three categories. In the first category, the performance of the bag-of-words approach using the SIFT and SURF descriptors were compared when classifying paddy field insect pests. For this approach, the SIFT or SURF features were extracted from the local patches on each image of the same class. These features were then quantified in a visual codebook in which the descriptors of the same class were represented as a bag-of-words.

In the second category, the performance of the HOG features was tested during the pest classification. For this approach, they maintained that all HOG parameters were consistent for different classes of pests. The detection window of 128 × 128 pixels was divided into 15 blocks horizontally and 15 blocks vertically, for a total of 225 blocks. Each block contains 4 cells with a histogram of 8-bin for each cell, for a total of 32 values per block. This brings the final size of the vector to 15 blocks on 15 blocks vertically × 4 cells per block × 8 cells per histogram = 7200 values. Finally, the concatenation performances of SIFT and SURF features with HOG were compared.

Their test results indicate that the HOG descriptors significantly outperform the existing local invariant features: SIFT and SURF in the pest classification for paddy field insect pests. HOG descriptors, when combined with SURF features, give a classification accuracy of about 90%. For simplicity and speed, the linear SVM was used as a classifier throughout the study.

In [10], the authors proposed to review various segmentation techniques used in image processing to detect plant diseases. There are thresholds such as the surface threshold, the Otsu threshold, the SVM (support vector machine), the nearest neighbor K classifiers, the fuzzy logic, the artificial neural network. The different techniques compared through accuracy and time.

In [11], the authors proposed to calculate the percentage of paddy leaf pixels affected by the disease. After k-means segmentation, the images are converted to a binary image and then calculated as a percentage of the affected pixels. Then calculate the affected pixels accurately from the infected rice leaf.

In [3], the authors proposed a new type of early detection system for pests. Fragmented leave images were processed to obtain a gray color image and resized to suit system requirements. They used the bi-cubic interpolation method to resize the image. They applied the smoothing filter to minimize the amount of noise and refine the visual quality of the image. The average filter used for the simplest smoothing. The Support Vector Machine classifier was used to classify pest types. The different types of properties include region properties, gray covariance matrix properties, and so on. The properties standard deviation, entropy, contrast and etc. In their research, the affected and unaffected images are compared using the dataset provided in the SVM. If it is an affected image again it is compared using the second dataset provided in the SVM.

The disparate properties which determine that they are whiteflies or aphids were listed in their results. Based on that the SVM is trained and input image of affected leaf is given to the second support vector machine which will generate the output as 1 or 0 based on the parameters of the input image. 1 is for aphids and 0 for white flies.

In [1] the authors proposed a software solution for early pest detection on the affected crops in greenhouse. Whiteflies are chosen for the study. The samples collected by using the pan tilt camera with 20X zoom maintaining equal illumination to the object in greenhouse. The images were converted into gray scale. Then images resized by using bi-cubic interpolation as it generates more accurate results than any other methods like Nearest-neighbor interpolation and bilinear interpolation. The average filter applied to reduce the noise and improve the visual quality of the image. The properties such as entropy, mean, standard deviation, eccentricity, Euler number, filled area, solidity, gray co-occurrence matrix, contrast, and energy are calculated from the filtered images in order to apply the classification. For classification the SVM classifier is used. SVM classifier helps to detect the pests and in the classification of pest based on their features. Support vector machine generates the output based on the comparison with the parameters of database. A special type of mask is used for identification of pests followed by the average filtering. Then the filtered image is convolved with the mask. Then extracting the region properties and gray co-occurrence matrix properties the classification is done in two types, whiteflies and aphids. They have considered region properties like standard deviation and contrast for the identification part. They again used SVM classifier for determine the category.

The results indicate more precision in identifying the presence of the pest at an early stage. They divided the images into two categories affected and unaffected. The affected category is also divided into two classes: aphids and whiteflies. Their result shows that the SVM training was performed with 100% accuracy.

In [4], the authors developed an algorithm for detecting and classifying the pests. Their suggested system provides a simple, effective and fast solution for detecting pests. They explain a simple and effective method based on image processing using a neural network for the classification of pests. The acquired images are resized and converted to grayscale. There are different properties such as region properties, gray covariance matrix properties. Among them; properties such as entropy, mean, standard deviation, contrast are extracted from the image. These properties are used as inputs for the Neural Networks classifier. That the characteristics of the input image are extracted and given as input to the network. The classification of diseases is carried out on the feed forward network. Affected images and unaffected images used to form the network. Their results indicate that there is a difference in standard deviation, contrast, variance, entropy, and so on. If a leaf is affected by the pest, the next step is to look for the type of pest that is whitefly or aphid. They represent derived features as inputs into a feedback neural network, resulting zero or one. Zero considered unaffected and one represents as affected image.

The authors claim that their method is as profitable and simple. And we can also conclude that when parasites are detected at an early stage, the use of pesticides is reduced by up to 80%.

In [15], a method proposed by the authors is to examine crops for pest infestation and to classify the type of pest on crops. The proposed algorithm counts the pests on the leaves and then estimates the number of white flies per leaf. This document presents a method for automatically identifying white flies from leaves. Color images require large-scale storage and processing time during pre-processing. For this reason, images are converted to grayscale so that they can be easily manipulated and require less storage space during processing. The grayscale images obtained by considering the following equation on the images.

$$I(x, y) = 0.2989*B + 0.5870*G + 0.1140*B$$

Bi-cubic interpolation was used to resize images, which gives better results than other resizing techniques, such as bilinear interpolation and nearest neighbor. The average filter used for filtering. The background of the image is calculated using morphological operators, which is essential once this image is subtracted from the original image, in order to detect the pests of the images. Thus, the resulting image will only have the 0 background pixel values and the objects with the pixel 1 values. They mentioned that their methodology gives better results than the watershed and Gaussian mixture segmentation models. . These algorithms use a large number of complex arithmetic operations such as division, multiplication and mean, which exponentially increase the execution time. The proposed method therefore improves performance and gives better results.

The erosion algorithm was used to eliminate the noise. After the noise was removed, the next objective was to improve the pest detected after the segmentation performed using the dilation algorithm. The gray level co-occurrence matrix (GLCM) and the regional properties of the images are calculated for the feature extraction process. These properties were used to train the support vector machine for image classification. Moore's neighborhood tracing algorithm and Jacob's stopping criterion were used to count pests on the 1% of plants per square of the field.

In [12] a model developed for the detection of plant diseases. First, the input RGB image has been converted to HSI, because RGB is used for color generation and this is for the color descriptor. Then, the green pixels are masked and deleted using a specific threshold value, then the image is segmented and the useful segments are extracted. Finally, the texture statistics are calculated from the SGDM matrices. Finally, the presence of diseases on the leaf of the plant is evaluated.

In [8], the authors proposed an automatic diagnostic system for the classification of tea pests using correlation-based feature selection (CFS) and the incremental back propagated learning network (IBPLN). The authors created a database focusing on eight major pests from separate tea garden records in the districts of northern Bengal in India. The database consists of 609 instances belonging to eight classes described by eleven attributes (signs and symptoms); which are all nominal. The classification was performed using artificial neural networks. The results of the classification were compared with the original feature set and with a reduced set of features. Their research indicates that CFS can be used to reduce the feature vector and that the CFS + IBPLN combination can be used for other classification problems.

In [9], the authors proposed a solution based on computer vision for automatic detection and classification of leaf diseases. Five diseases were tested using their solution, which has an effect on the plants. The suggested model begins by creating a structure for transforming the color space independent of the device. Therefore, they created the color transformation structure that defines the color space conversion. Then, they applied the device-independent color space transformation, which converts the color values of the image to the color space specified in the color transformation structure. Finally, the K-means grouping was used to partition the image of the leaf into four groups in which one or more groups constitute the disease in case the leaf is infected with several diseases. The image analysis technique used for this study was the co-occurrence methodology. The network structure used in this research was given as follows. They chose the number of hidden layers in the neural network. The number of entries is equal to the number of texture elements of the neural network mentioned in this search. Feedback propagation networks used as a neural network. The performance function was the mean squared error and the number of iterations was 10,000 and the maximum allowed error was 10-5. Based on their experiences, the neural network classifier developed based on statistical classification works well and can successfully detect and classify the diseases tested with an accuracy of about 93%.

In [16], a method proposed by the authors to demonstrate the combination of the k-means classification algorithm and the correspondence filter to detect and recognize pests. They tested their algorithm on more than ten crops or plant pests. The RGB images in the dataset have been transformed into L * a * b color space. The k-means classification technique was then used to segment the images of plant pests. The process of segmenting objects or pests begins by identifying pixels of predominantly green color using a variable threshold estimate. All red, green, and blue color components of the pixel below the defined threshold are set to zero, so that all pixels with intensities below the threshold value are rejected. This technique is applied to the extent that they assume that these pixels are not useful for the identification of the pest.

All pixels on the object group boundaries and all colour pigments, designated zeroes, have been removed and segmented. The second step of the algorithm begins by angle multiplexing the pest detected from the k-means algorithm in the 5 'increment correspondence filter. The recognition process was then initiated using the constructed correspondence filter. Correspondence filters are normalized as a 3D plot and displayed. The peak of the matrix occurs where the pest images are best correlated, giving a maximum correlation height, equivalent to +1.0 and above. The input images are processed again by the k-means clustering algorithm and the correspondence filtering training images. The result confirmed the presence of homogeneous and similar objects.

The correspondence filter has been precisely tested for detections and recognition accuracy, particularly for its distortion invariance capability. To optimize correspondence filter recognition performance, appropriate values for the alpha, beta, and gamma adjustment parameters had to be selected. For these correspondence filters, the chosen adjustment parameters were $\alpha = 0.0000009$; beta = 0.45 and gama = 0.1.

In [6], the proposed methodology involves reduced computer complexity and aims to detect pests not only in a greenhouse environment but also in a farm environment. The acquired images are converted from RGB to luminance and Chroma components (YCbCr). Mixture models and watershed segmentation algorithms are used to extract whiteflies from the background of the converted image. These algorithms involve a large number of arithmetic calculations in the form of divisions, multiplications and calculation averages, which leads to an exponential increase in execution time. To obtain only whiteflies, the erosion algorithm used. The image enhancement in the form of dilation is also performed to improve the clarity of the segmented whiteflies. The counting algorithm is implemented using Moore Neighbor tracing and Jacob stopping criteria to count the number of white flies on each affected leaf. Their illustrated experience, the RDI algorithm uses only two subtractions and a pixel comparison operation to separate the whiteflies. They state that many factors, such as the variation in light intensity illuminating the leaf, the presence of insects or pests other than those of interest to us, and the whiteflies in the vicinity have been taken into account in order to verify the precision algorithm.

2.3 Summary

This chapter elaborates the findings of the literature study and summary of the previous work on the problem domain. From this chapter, it is noted that most of the researchers considered the gray images than RGB images due to the computational complexity while processing the images. And, the region properties mostly considered for the classification process especially when they applying the various neural network architectures for the classification task. From the literature study we decided to select convolution neural network as the classification algorithm and thresholding method for the segmentation task.

CHAPTER 03

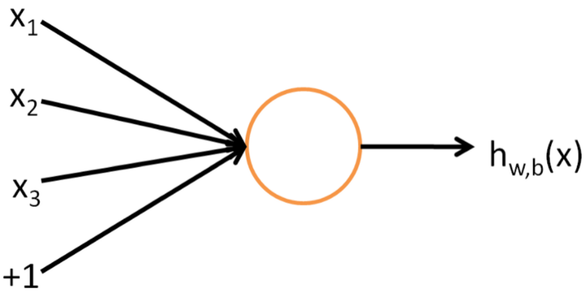
Convolutional Neural Networks

3.1 Introduction

This chapter presents the working structure of the convolution neural network. Moreover, the chapter elaborates the internal structure and the logic of each structure in convolution neural network.

3.2 Neural Networks

Neural Networks is one of the most important machine learning algorithms at present. As the name suggests, neural networks were inspired by the neural architecture of a human brain, and like in a human brain the basic building block is called a Neuron. Its functionality is similar to a human neuron, i.e. it takes in some inputs and fires an output. In purely mathematical terms, a neuron in the machine learning world is a placeholder for a mathematical function, and its only job is to provide an output by applying the function on the inputs provided.



The function used in a neuron is generally termed as an activation function. There have been 05 major activation functions tried to date, step, sigmoid, tanh, and ReLU. Neuron and activation functions which together for the basic building blocks of any neural network.

Neurons are organized into a series of groups called layers. In each layer, the neurons are connected to the neurons of the next layer. The data comes from the input layer to the output layer along these compounds. Each individual node performs a simple mathematical calculation. Then, it transmits its data to all the nodes to which it is connected.

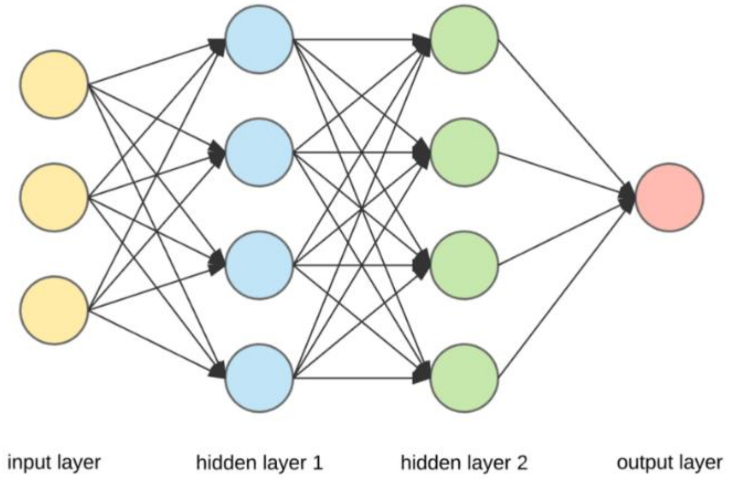


Figure 01: Layer structure of the neural network

The latest wave of neural networks is related to the increase in computing power and the accumulation of experience. This has led to in-depth learning, where the technological structures of neural networks have become more complex and able to solve a wide range of tasks that could not be solved effectively before. The classification of images is a striking example.

A neural network with more than one hidden layer is usually called a deep neural network.

3.3 Architecture of Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the variants of neural networks widely used in the field of computer vision. It takes its name from the type of hidden layers of which it is composed. The masked layers of a CNN typically contain convolutional layers, clustering layers, fully connected layers, and normalization layers. Here, it simply means that instead of using the normal activation functions, the convolution and pooling functions are used as activation functions.

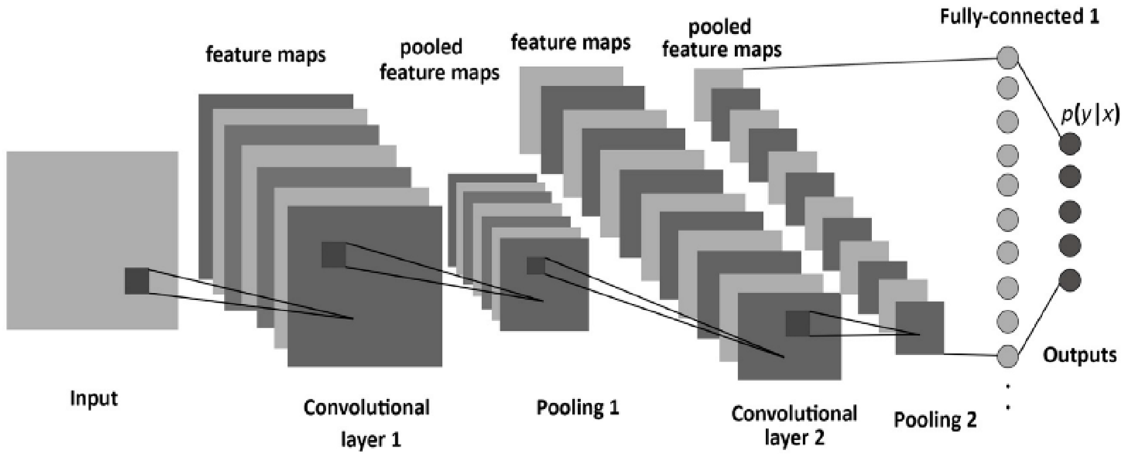


Figure 02: Layer structure in convolution neural network

To understand it in detail one needs to understand what convolution and pooling are. Both of these concepts are borrowed from the field of Computer Vision and are defined below.

3.3.1 Convolution:

Convolution uses two images (in 2D): one as an input image and the other called the kernel as a filter on the input image, producing an output image (the convolution takes two input images and generates an output as the third image).

In simple terms, it takes an input signal and applies a filter on it, essentially multiplying the input signal with the kernel to get the modified signal. Mathematically, a convolution of two functions f and g is defined by:

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2)$$

This is nothing other than the scalar product of the input function and a function of the kernel. In the case of image processing, it is easier to visualize a kernel that slides over an entire image and thus modifies the value of each pixel of the process.

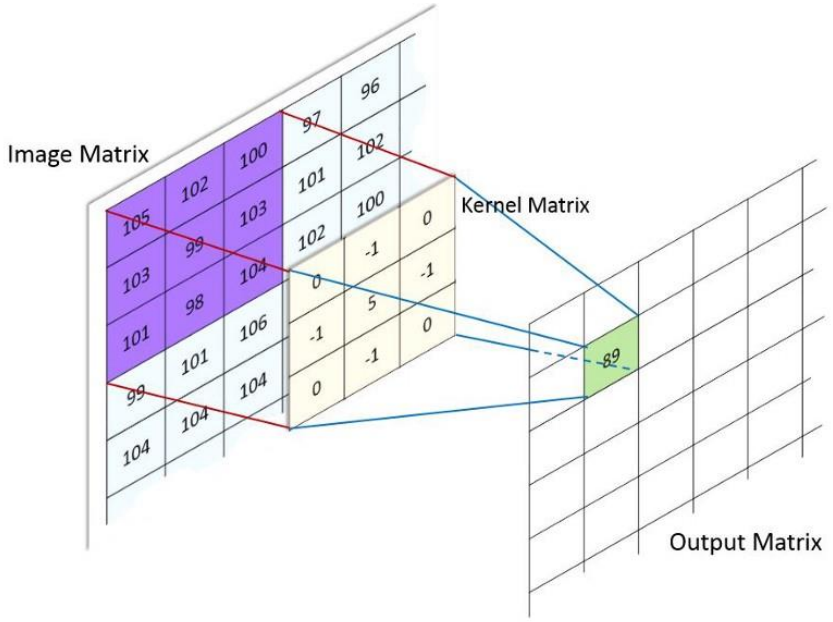
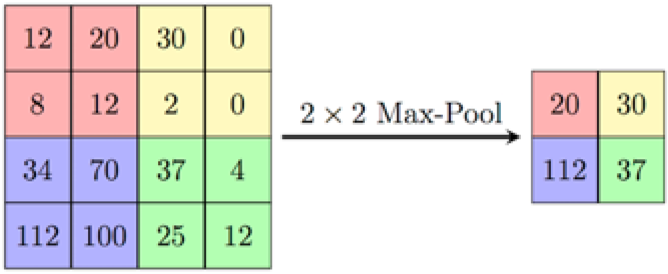


Figure 03: convolution process

3.3.2 Pooling:

Pooling is a discretization process based on a sample. The objective is to subsample an input representation (image, hidden layer output matrix, etc.), reducing its dimensionality and allowing assumptions to be made about the entities contained in the aggregated sub-regions.

There are 2 main types of pooling commonly known as max and min pooling. As the name suggests max pooling is based on picking up the maximum value from the selected region and min pooling is based on picking up the minimum value from the selected region.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Figure 04: Pooling process

Thus, as can be seen, a convolutional neuron network or CNN is basically a deep neural network consisting of hidden layers having convolution and pooling functions in addition to the activation function to introduce non-linearity. .

3.3.3 Complete CNN architecture:

Technically, CNN models to learn in depth to train and test, each input image pass through a series of convolutional layers with filters (Kernels), pooling layers, fully connected (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The figure below is a complete stream of CNN for processing an input image and classifying objects based on values.

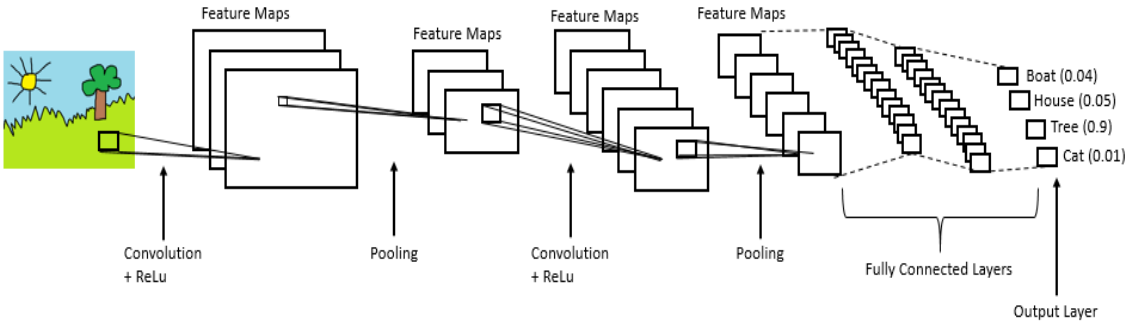


Figure 05: Complete CNN architecture

3.4 Summary

This chapter describes convolution neural network as the classification algorithm which contains the complex architecture and perform well based on the complex calculations. Also explains how an image taken as input and produce output as the predicted label from the fully connected layers.

CHAPTER 04

Proposed methodology

4.1 Introduction

In this research work, the images are acquired using a camera in Kaluthavalai region, Batticaloa and from Google images.

In a complex field environment, many factors influenced the accuracy of segmentation of the target leaf. Factors such as soil, sand, membranes, rods, water pipes, etc. Although the background may vary depending on the environment, due to the different colour, the pixels of the leaf pictures of the plant can be classified as background (non-leaf regions) and leaf regions (foreground). Background regions are usually non-green (for example, yellow), while leaf regions are usually green (sometimes, background regions are also green in some images). Thus, considering the colour, the background can be eliminated with precision. Therefore, we have used green channel levels in the RGB colour space to remove pixels that are not considered to be a part of the leaf regions.

Figure 06 describes the basic procedure of the methodology proposed in our research.

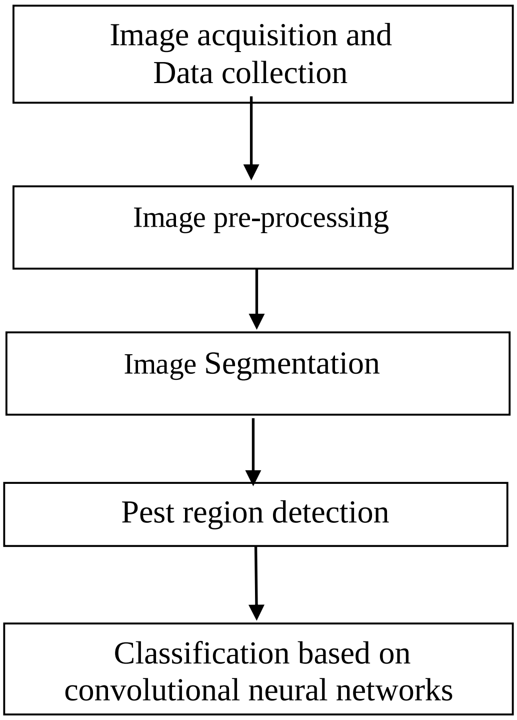


Image acquisition and Data collection
Image pre-processing
Image Segmentation
Pest region detection
Classification based on convolutional neural networks

Figure 06: The basic procedures of the proposed model

4.2 The step-by-step procedure of the proposed methodology:

- RGB image acquisition
- Converting input images from RGB to HSV.
- Apply a threshold to hide green pixels.
- Remove hidden green pixels.
- Segment the leaf portion and form edges for the region of interest.
- The model detects which leaf component is affected or not based on the characteristics of the texture.
- Detect and remove holes other than the pest region using morphological operations.
- Calculate the properties of the region.
- Detect the affected region using the selection frame among the holes using the area value calculated from the region properties.
- Count the number of pests.
- Create a classification model to identify the type of pest detected from the affected leaf using the convolutional neuron network.
- Identification of the type of pest.

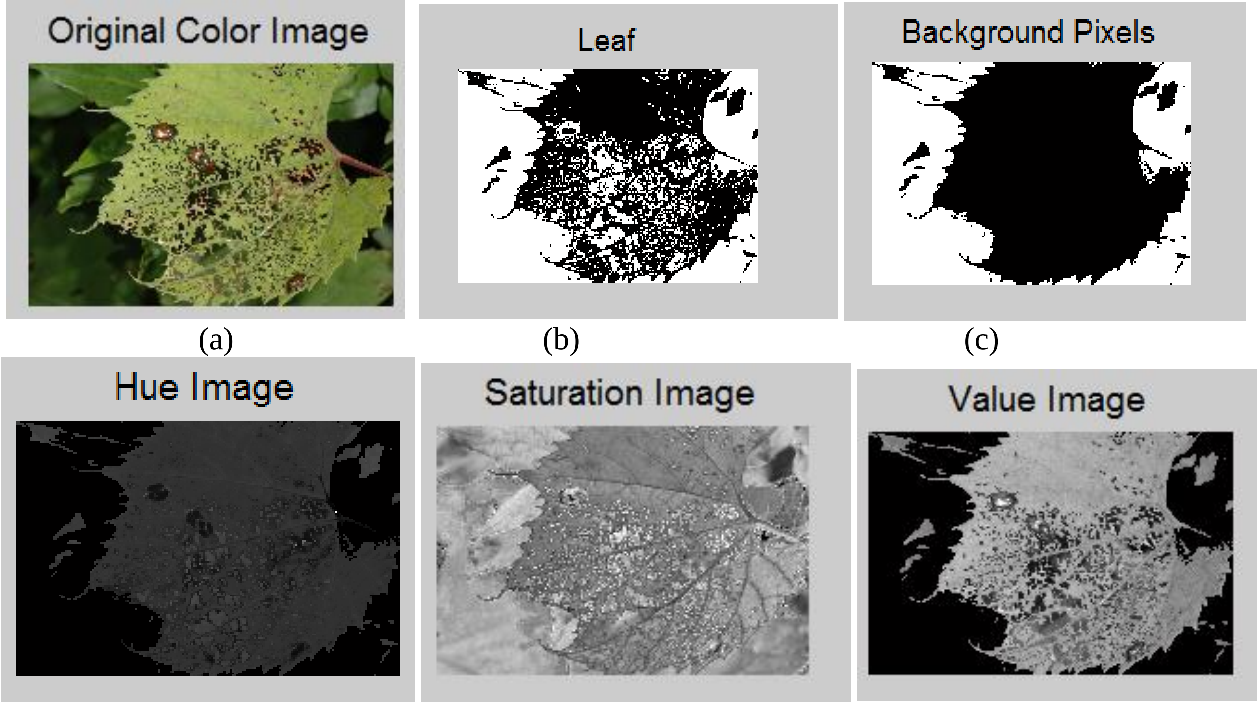
4.2.1 Colour Transformation Structure*:*

First, the images (RGB) of the leaves are taken by the camera from the garden and some images are downloaded from imagenet.org. The input images are then converted from the RGB colour space to a representation of the HSV colour space (saturation hue value). The RGB format is ideal for colour generation. But the HSV model is an ideal tool for colour perception. Hue is a colour attribute that describes the pure colour perceived by an observer. Saturation refers to the relative purity or amount of white light added to the hue and the value means the amplitude of the light. The Hue component is used for further analysis followed by the color transformation structure. Then, the Saturation and Value components are removed because they do not provide additional information.

4.2.2 Masking the green-pixels by thresholding:

Set the pixel value in an image to zero or some other background value called the masking process. Here, we focus on identifying pixels of predominantly green color. This is accomplished according to the specified threshold value calculated for these pixels. The green components of the pixel intensities are set to zero if they are less than the pre-calculated threshold value.

Figure 07 shows the region of interest and the HSV components after removing the background. To remove the background, threshold value obtained from the value level in the image of the value component. Images taken in different environmental conditions as well as lighting affect the threshold value. For these reasons, the threshold value is changed for each image.



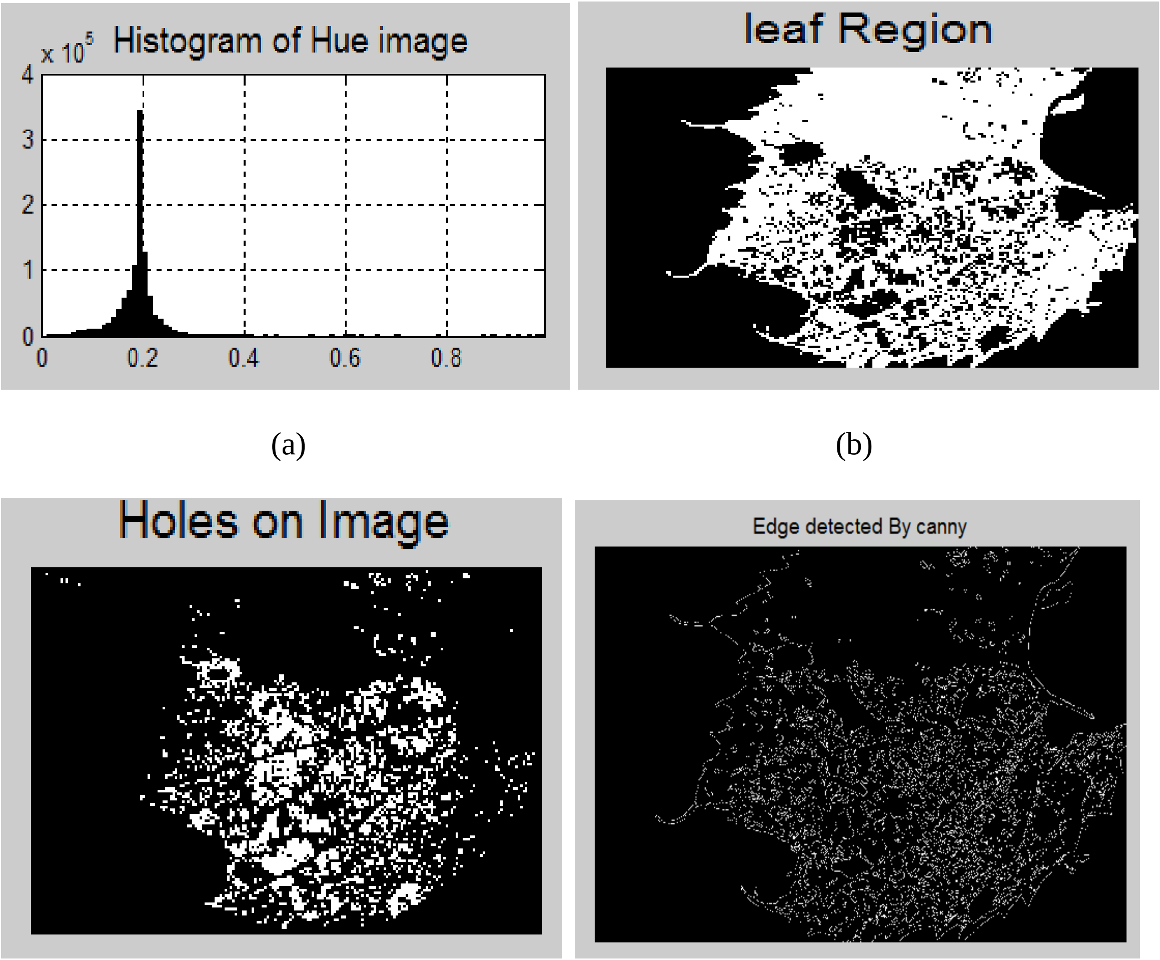
(a) (b) (c)

(d) (e) (f)

Figure 07: (a) Input image (b) Green regions (c) Background removed
(d) Hue components (e) Saturation components
(f) Value components

4.2.3 Obtaining leaf Segments:

In this step, the leaf segments are separated. A histogram obtained from the image of the Hue component. On the basis of the histogram, the regions of the leaves are separated from the binary image. And the non-interested areas outside the leaf considered as noise and which have been suppressed by the properties of the regions and the related components. Then, the leaf region obtained with the holes using & logical operation in the image. Figure 08 shows the hue histogram and leaf area with holes and pests.



(a) (b)

(c) (d)

Figure 08: (a) histogram of hue image (b) Leaf region (c) Holes in Leaf region (d) Edge detection for leaf region with holes.

4.2.4 Compute the affected area from leaf potion with the pest region:

In this step, the percentage of the affected region is calculated taking into account the holes in Figure 08 (c) and the total number of pixels available in the leaf region of the image. It should be noted that the affected area included harmful areas in the image. So that the area of the pest in the image is detected.

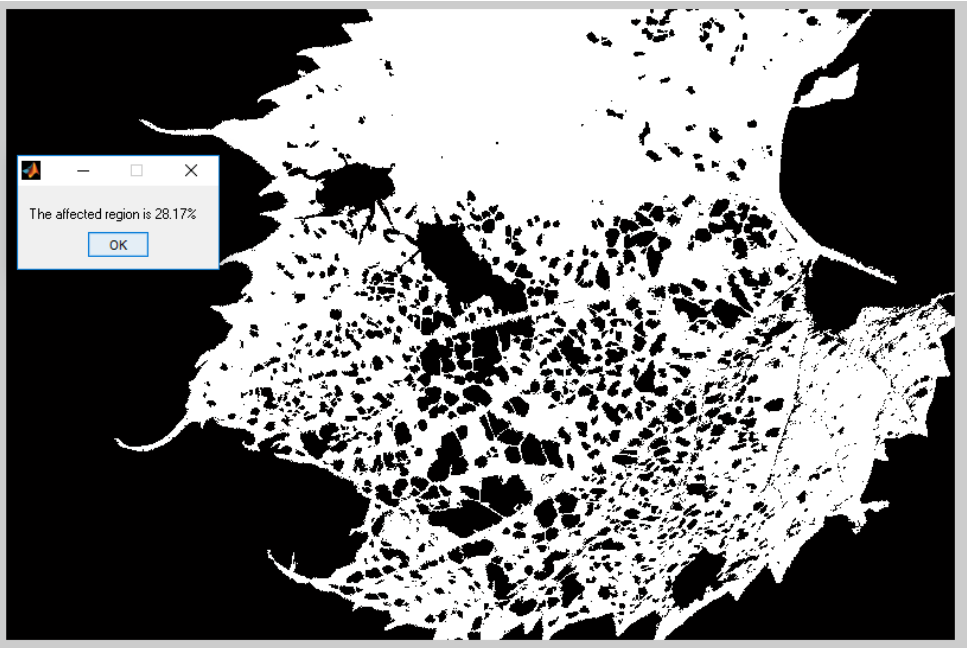

The affected region is 28.17%

Figure 09: Affected region calculated from the leaf image (with pest region).

4.2.5 Morphological operations to detect the pest region in the leaf image:

Morphological image processing is suitable for removing small holes in the image. The bwareaopen operation removes all connected components (objects) having less than 08 pixels from a binary image. Then, the morphological closing operation is applied, followed by a filling operation to detect the region of the pests from the image of the leaf. The distances between the centers of mass with all the pixels along the limit for each object were calculated. Then, some holes are removed from the image based on the appropriate range selection. The result of the process given in Figure 10 below.

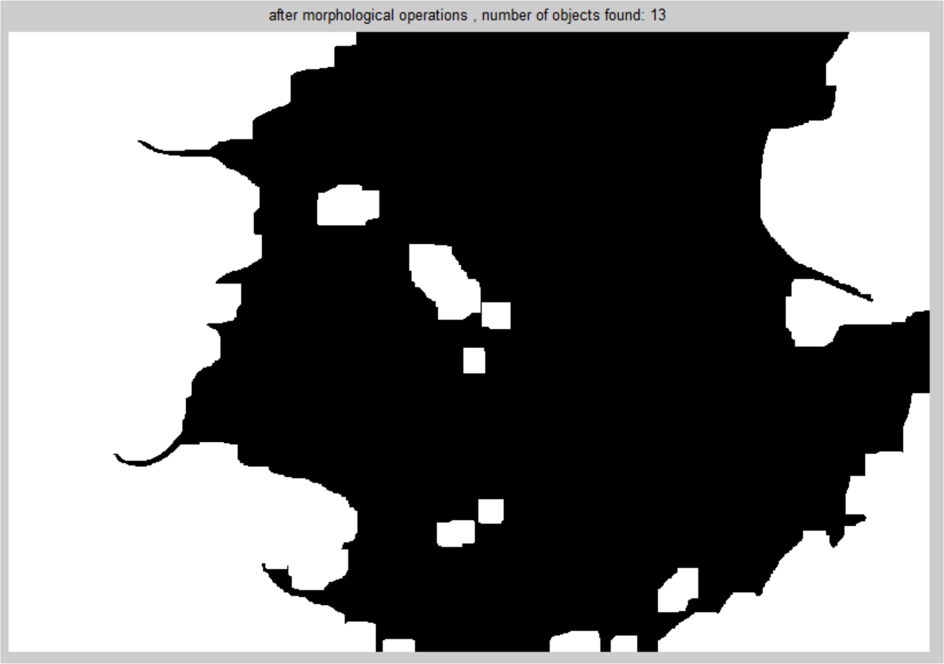
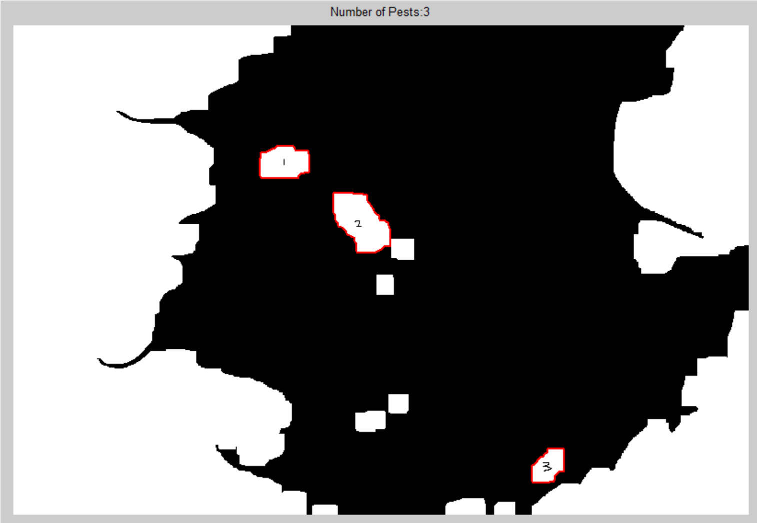

after morphological operations , number of objects found: 13

Figure 10: Output image after the morphological operations

4.2.6 Pest regions detected by considering the region properties:

In this step the region properties such as area, eccentricity, Euler number and etc. were calculated. Then the pest regions selected by the tracing the boundaries using *bwboundaries* and the proper selection of the area (calculated under the region properties). The output obtained from this step given below.

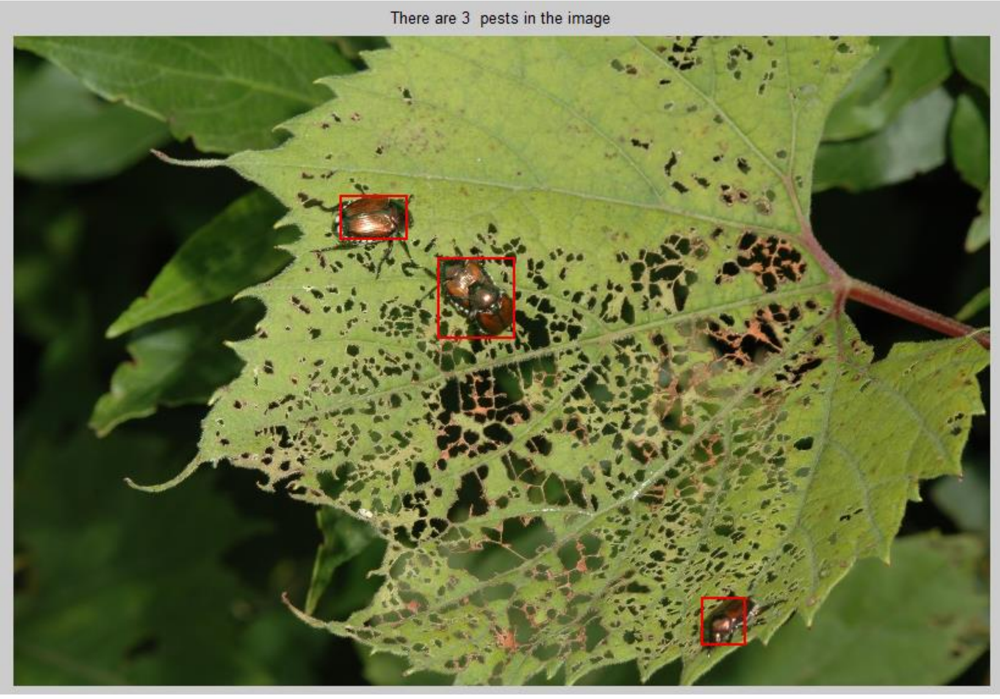


Number of Pests:3

Figure 11: Pest regions detected from the binary image

4.2.*7* Pest regions detected by considering the region properties:

Depending on the region properties and the bounding box option, the regions of the pest detected in the affected leaf images. Counting pests on the leaves is one of the objective of the research, so that it can give an idea of the amount of harmful attack on a leaf. In Figure 12, the number of harmful organisms counted and that displayed at the top of the image.



There are 3 pests in the image

Figure 12: Pest regions detected from the binary image

4.3 Construct the Classification model using convolution neural network

To detect the type of pest from the leaf image, a classification model must be constructed. In order to build the classification model, a Matlab "MatConvNet" toolox was used in the experiment. MatConvNet is an implementation of the convolutional neural networks (CNN) for MATLAB. The toolbox is designed with an emphasis on simplicity and flexibility. It introduces CNN building blocks as easy-to-use MATLAB functions, providing routines for calculating linear convolutions with filter banks, pooling capabilities, and more. In this way, MatConvNet allows rapid prototyping of new CNN architectures; at the same time, it supports efficient computing of processors and GPUs to form complex models on large data sets such as ImageNet ILSVRC. In our experience, micro-processor (CPU) based computing architecture has been used to train and test the dataset model. The version of Matconvnet-1.0-beta-17 and Matlab R2014a was also used for implementations.

4.4 Summary

In this chapter a methodology explained for the whole process of detecting affected region and counting the pest from the image. The methodology not fully supported for the detection of pest region, hence the work semi-automated for the pest detection then the classification process continued. The outcome of all the steps clearly mentioned under the steps. Finally the need of toolbox for the implementation of convolutional network considered.

CHAPTER 05

Experimental setup & Results

5.1 Introduction

This chapter includes the experimental setup for the classification model and the configuration of MatConvNet for the classification.

5.2 Dataset setup for the classification model

There were five classes of pest images considered in the research work that listed given below.

Type of pest (Class label)	Number of pests
Flea beetle	679
June beetle	922
Lady bug	948
Squash bug	541
Tarnish plant bug	328

Table 1: Types of pests and number of images

For the classification model, data set divided into training, testing and validation sets. In each class 70% of the images selected for training purpose and remaining 30% of images selected for testing purpose. Then 70% of images selected to build the training model from the training images and the remaining 30% of the training images used to build the validation model. And the classification performances calculated based on the training model applied on the testing images.

The figure below depicts the dataset preparation for the classification model in a pest class.

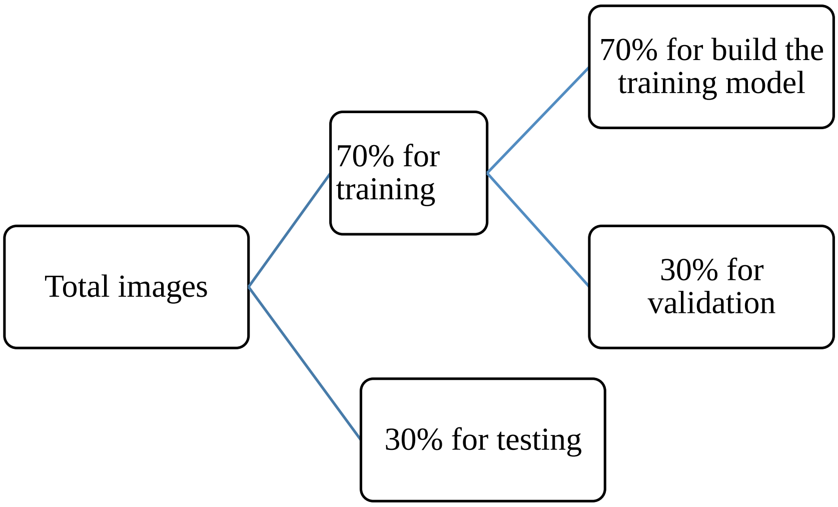
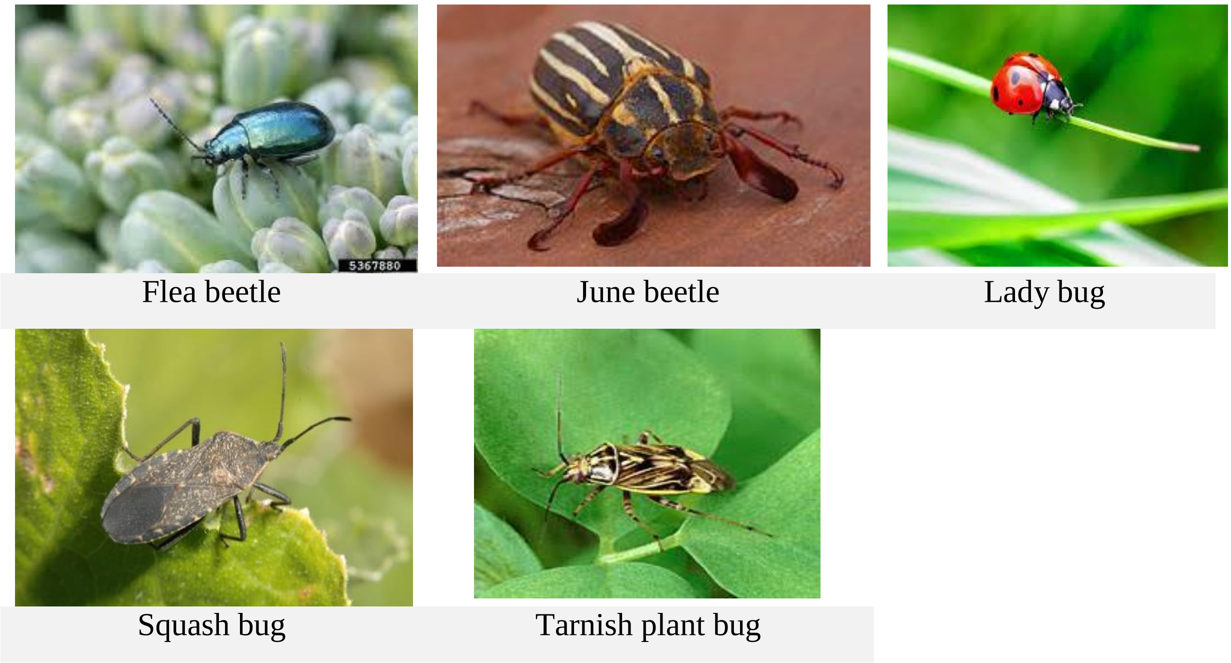


Figure 13: Data set model for experiments

Figure 14 shows some examples of selected five pest species that mainly found in the Batticaloa area.


Flea beetle

June beetle

Lady bug

Squash bug

Tarnish plant bug

Figure 14: Types of pest considered for the recognition process

5.3 MatConvNet configuration to build the training model on the dataset

The two commonly used approaches to deep learning are the formation of a model from scratch and transfer learning. Both approaches have advantages and can be used for different in-depth learning tasks. Developing and training a model from scratch works best for very specific tasks for which pre-existing templates cannot be used. The disadvantage is that this approach usually requires a large amount of data to produce accurate results. In our research the dataset provide less number of images it was not enough to build the training model from scratch. So that the training model has built from the dataset by using the pre-trained networks. There are lot of pre-trained networks available to build the training network models such as alexnet, vgg16, resnet18 and etc. [17].

In our research "imagenet-vgg-m" pre-trained networks model have used to build the training model. The classification procedure using the MatConvNet explained in the following diagram.

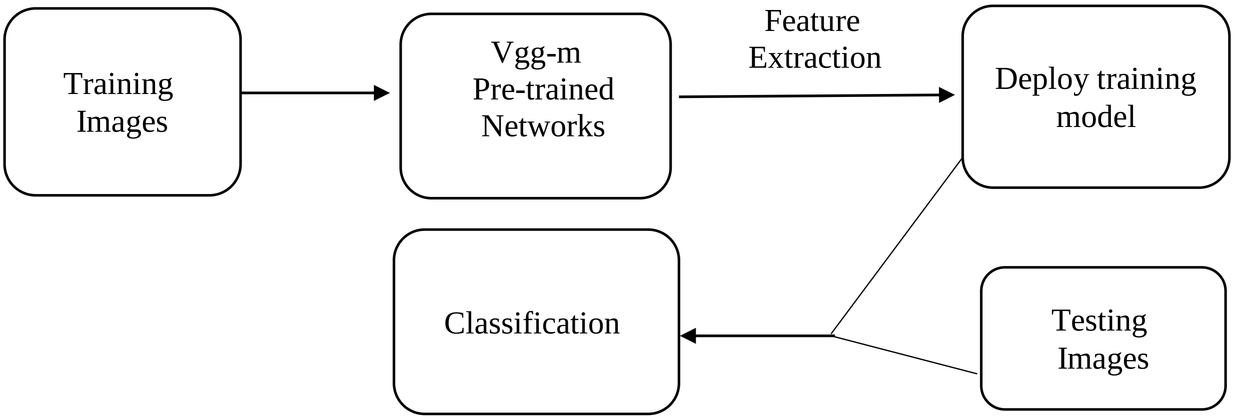


Figure 15: MatConvNet procedure

5.4 Classification results (Testing results)

The classification rate calculated on the testing images by using the following equation.

$$\text{Rate}= \frac{\text{Number of images which are correctly classified}}{\text{Total number of testing images}} \times 100$$

The deployed training model have used to calculate the classification rate. The rate calculated for each classes listed below.

Type of pest	**Total number of testing images**	**Correctly classified images**	**Classification rate**
Flea beetle	197	151	76.65 %
June beetle	266	221	83.08 %
Lady bug	283	203	71.73 %
Squash bugs	156	104	66.67 %
Tarnish plant bugs	89	58	65.17 %

Table 2: classification rate for the testing image set

From the above table we can observe the tarnish plant bugs has the low classification rate. Because the squash bugs and the tarnish plant bugs has the small variation which is only the surface color. So most of the testing images under the tarnish plant bugs misclassified into the squash bugs. Also the number of images taken from the type of pest also less amount. Due to this reason the classification rate may differ from others.

5.5 Summary

In this chapter, explains the experimental setup for the classification process and displayed the classification rate based on the experiments.

CHAPTER 06

Discussion and Conclusion

In our research a method suggested to find the affected area, detect the number of pests and determine the type of insect pest, a classification model built using the convolutional neuron network. Implementation of the system of identification of pests to using image processing techniques has caused many problems. Major difficulties are due to minor changes. Due to the change of light, the characteristics of the image such as color, intensity, contrast, etc. may vary. Also the methodology not fully supported for the detection of pest region due to these reasons, hence the work semi-automated for the pest detection then the classification process continued

Another difficulty is, we need more number of pests in each classes to get higher classification rate by using convolution neural network. Due to this reason pre-trained networks model used to build a training model. Also different class pest have more similarity that made effects to decide the pest type.

Here we have considered the Convolution neural network implementation by using the MatConvNet (matconvnet-1.0-beta17) for the classification process but there are latest versions with the extended versions of the convolution neural networks such as R-CNN, Fast R-CNN and Faster R-RCNN for the object detections. In order to accomplish the latest feature advantages we need more computational power like GPU based computer architecture than CPU based architecture. Also CUDA - a parallel computing platform and other image processing toolboxes need to be considered to gain the higher classification rate. In future we will focus on this type of work with the large amount of data.

Finally, this research makes it possible to identify pests at an early stage. So we can reduce pesticides. Accuracy can be improved by using different methods using image processing techniques compared to manual systems. And, to suggest the appropriate pesticides based on the affected pattern and the recognition of the pest type we need more research on this domain.

References

[1] Rupesh G. Mundada and Dr.V. Gohokar, "Detection and Classification of Pests in Greenhouse Using Image Processing",IOSR Journal of Electronics and Communication Engineering, Vol 5, Issue 6 (Mar - Apr) 2013, pp 57-63.

[2] Venugoban K. and M. A. Ramanan "Classification of rice field images insect pests using function gradients". International journal of Machine learning and computing, 1-5, February 2014..

[3] Sreelakshmi M and Padmanayana " Early Detection and Classification of Pests Using Image Processing". International Journal of innovative research in electrical, electronics, instrumentation and control, Vol. 3, special Issue 1,April 2015, ISSN 2321-5526.

[4] Monika Wadhai, V. V. Gohokar and Arti Khaparde " Agricultural pest control using computer vision technique" International Journal of Advanced Research (2015), Volume 3, Issue 8, 309-314

[5] Preetha Rajan Radhakrishnan. B and L. Padma Suresh " Detection and Classification of Pests from Crop Images Using Support Vector Machine". 2016 International Conference on new technology trends [ICETT], 21-22 October. 2016.

[6] Sushma R. Huddar, Swarna Gowri, Keerthana K., Vasanthi S. and Sudhir Rao Rupanagudi, " Novel Algorithm for Segmentation and Automatic Identification of Pests on Plants using Image Processing", Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on 26-28 July 2012, IEEE-20180.

[7] Prathibha G P, T g Goutham, Tejaswini M V, Rajas P R, Mrs Kamalam Balasubramani, "Early Pest Detection in Tomato Plantation using Image Processing", International Journal of Computer Applications (0975 – 8887), Volume 96– No 12, june 2014.

[8] R. K. Samanta and I. Ghosh, "Tea Insect Pests Classification Based on Artificial Neural Networks" , International Journal of Computer Engineering Science (IJCES), vol. 2, Issue 6, 2012.

[9] Dheeb Al Basish, Malik Braik et Sulieman Bani-Ahmed " Detection and classification of leaf diseases Using K-means-based segmentation and Neural-based Classification". Information Technology Journal 10 (2):267-275, 2011, ISSN 1812-5638 / DOI:10.3923/itj.20110267.275.

[10] Rabia Masood, S.A. Khan, M. N. A. Khan, "Plants disease Segmentation using image Processing" I. J. Modern Education and Computer Science, 2016, 1, 24-32.

[11] Rashedul L'Islam, Md. Rafiqul Islam, "An Image Processing Technique to Calculate Percentage of Disease Affected Pixels of Paddy Leaf", International Journal of Computer Applications, Volume 123 – No.12, August 2015.

[12] Sanjay B. Dhaygude, Nitin P. Kummbhar, "Agricultural plant Leaf Disease Detection Using Image Processing", International Journal of advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 1, January 2013.

[13] S.S. Lomte, A. P. Janwale, “Plant leaves image segmentation Techniques: a Review”, JCSE International Journal of Computer Sciences and Engineering, Volume-5, Issue-5, E-ISSN:2347-2693, may 2017.

[14] Al-hiary, S. Bani-Ahmed, M. Reyalat, M. Braik et Z. ALRahamneh, “Fast and Accurate Detection and classification of plant diseases”, International Journals of computer application (0975-8887), Volume 17-No, 1, march 2011.

[15] Danish Gondal , Yasir Niaz “Early detection of pests in crops using image processing and computational intelligence”, FAST-NU Research journal(FRJ), Volume 1, Issue 1, January 2015.

[16] Faithpraise Fina, Philip Birch, Rupert / Young, J. Obu, Bassey Faithpraise and Chris Chatwin, “Automatic plant pest detection recognition using k-means clustering algorithm and correspondence filters”, International Journal of Advanced Biotechnology and research,ISSN 0976-2612,volume 04, Issue 2,2013, pp 189-199.

[17] http://www.vlfeat.org/matconvnet/

Appendix A

Implementation for affected region and pest detection

```
clc;                % clear command window.
close all figures;  % close all figures (except imtool.)
clear;            % delete all existing variables. Or learvars if you want.
fontSize = 14;

%=read image from folder
folder = pwd;
baseFileName = 'D:\MSc research\leafs as affected\sample_1.jpg';
% Get full filename, with Path added.
fullFileName = fullfile );
if ~ exists (fullFileName,'file ')
    % Don't find it there.  Check the search path.
    fullFileName = filename; % no path this time.
    if ~ exists (fullFileName,'file ')
        % Hasn't found it yet.  Alert the user.
        errorMessage = sprintf ('Error: %s does not exist.", fullFileName);
        uiwait(warndlg(errorMessage));
        return;
    end
end
I = imread (fullFileName);
% Get the dimensions of the image.  numberOfColorBands should be = 3.
[lines, columns, number of colorbands] = size (I);
% Display the original image in color.
subplot(3, 3, 1);
imshow (I);

% Convert to color space HSV
hsv = rgb2hsv (I);
h = hsv (:,:, 1);
s = hsv (:,:, 2);
v = hsv (:,:, 3);

greenpixels = v < 0.465;
% Display black pixels.

subplot(3, 3, 2);
imshow (greenpixels);
title ("Leaf", "FontSize", fontSize);

background = xor (greenpixels, imclearborder(greenpixels));
% Display the background pixel.
subplot (3, 3, 3);
imshow(background);
title('background pixel', 'FontSize', fontSize);

% Mask the H, S and V images.
h(background) = 0;
s(background) = 0;
v(background) = 0;
% Display
subplot(3, 3, 4);
imshow(h);
title('Hue Image', 'FontSize', fontSize);
```

```
subplot(3, 3, 5);
imshow(s, []);
title ('Saturation Image', 'FontSize', fontSize);
subplot(3, 3, 6);
imshow(v, []);
title ("Image value", "FontSize", fontSize);
drawnow;

LLL=hsv2rgb(v);

 [pixelCount, graylevels] = hist(h(:), 100);
% Remove the large peak at zero gray levels due to background.
pixelCount (1) = 0;
subplot (3, 3, 7);
bar(graylevels, pixelCount));
grid on;
title ("histogram of the hue image"," FontSize", "fontSize""));
xlim ([0 grey level (end)]); % scale X axis manually.


inRegion = (h > 0,18) & (h < 0,50);
% Call anything else (not background) "affected."
leafRegion = ~ greenpixels & ~background;
%{
subplot (3, 3, 8);
imshow(inRegion, []);
title ("leafRegion", "FontSize", "fontSize")"));
%}

CC = bwconncomp (leafRegion,8); % determine connected components:
S = regionalprops (CC, "Area"); % calculate area of each component:
L = labelmatrix (CC); %remove small objects:
% idx = (find ([S. Area] >= 50000) / /(find ([S. Area] <= 125000))));
BW2 = ismember(L, find ([S. Area] > = 115000));

subplot(3, 3, 8);
imshow (BW2, []); %of leaf area image is BW2
title (leaf regional ', 'FontSize', fontSize);

% Find holes to perform the following operations
holes=~leafRegion & ~background;
subplot (3, 3, 9);
imshow ('holes, []);
title ("holes in the Image', 'FontSize', fontSize);


figure, imshow(BW2);
title ('region of interest', 'FontSize', fontSize);

figure,imshow(I);
title('Original Image', 'FontSize', fontSize);
figure,imshow(BW2);
title('Leaf Region', 'FontSize', fontSize);
figure, imshow (holes);
title (‘holes on the leaves’, 'FontSize', fontSize);

%============Canny Edge detector application
Edges = edge(BW2,'canny'));
figure,imshow(Edges);
title ('edge detected by canny operator, 'FontSize', fontSize);
```

```
%{
Edges = edge (I,'Prewitt');
figure,imshow(Edges);
title ("edge detected by prewitt operator', 'FontSize', fontSize);
%}

%=============Calculate the area affected as a fraction of the Leaf potion
entireLeafPixels = sum (BW2(:));
areaFraction = (sum (holes (:)) / entireLeafPixels);
message = sprintf ('the area is %.2f%%%', areaFraction * 100);
uwait (msgbox (message));

% imBW = im2bw (BW2, 0.3); % / / specify manual threshold of 0.3
%smallestacceptablesholesize = 2500; % in pixels.
% imBW = ~bwareaopen (imBW, smallestacceptable holesize);

%{
CC1 = bwconncomp(BW2, 8);
S1 = regional crops (CC1, area");
L1 = labelmatrix (CC1);
BW3 = ismember(L1, find([S1.The area] >= 50000)); %%%%%%%%%not worked
figure,imshow(BW3);
%}

openInput=bwareaopen(BW2,4);

ClosedInput = imclose(openInput,ones(36));
imshow(ClosedInput);
title('Closed Image', 'FontSize', fontSize);
hold on;
imBWFilled = imfill(~ClosedInput, 'holes');

 [L,num] = bwlabel(imBWFilled);

imshow(imBWFilled);
title(['after morphological operations , number of objects found: '
num2str(num)]);

centres = zeros(num,2);
for n = 1 : num
    bmap = L == n;
    [rows,cols] = find(bmap == 1);
    centres(n,:) = [mean(cols) mean(rows)];
end

bwBound = bwperim(imBWFilled, 8);


% For each object, find the distances between the center of mass with all %
of the pixels along the limit for each object.
%Find the range (max - % min).
ranges = zeros(num,1);
for n = 1 : num
    bmap = L == n;
    boundPix = bwBound & bmap;

    [rows,cols] = find(boundPix == 1);
    dists = sqrt((cols - centres(n,1)).^2 + (rows - centres(n,2)).^2);
    ranges(n) = max(dists(:)) - min(dists(:));
end
%

% Find those object IDs that have less than a range of 100.

indPests = find(ranges > 100);

% Total number of pests
numPests = numel(indPests);
regionp = regionprops (imBWFilled,'centroid'));
[labeled, numObjects]=bwlabel (imBWFilled,8);
stats=regionprops(labeled,'Eccentricity','Area','Orientation','Centroid',
'PixelList','EulerNumber','BoundingBox','PixelIdxList');
areas=[stats.Area];
the eccentricities=[stats.Eccentricity];
Orientation=[stats.Orientation];
The center of gravity=[stats.Centroid];
%PixelList=[stats.PixelList];
EulerNumber=[stats.EulerNumber];
BoundingBox=[stats.BoundingBox];
%PixelIdxList=[stats.PixelIdxList];
%%

figure;
imshow (imBWFilled);
title ['number of parasites:' num2str (numPests))]);
hold on;
%% Moving from larger borders
[maxValue, index] = max ([stats.Area]);
[rw,col]=size(stats);
for i=1: rw    if (i==index)
       imBWFilled(stats(i).PixelIdxList)=0;
% Remove all small regions except large area index
    end
end
idx = find((4000 <= areas) & (areas <= 15000));

imBWFilled = ismember(L, idx);

%%
axis image;

boundaries = bwboundaries(imBWFilled);
numberOfBoundaries = size(boundaries, 1);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'r', 'LineWidth', 2);
end

allowableAreaIndexes = (areas> 4000) & (areas <20000);
id = find (allowableAreaIndexes);
statsdefects = stats (id);

figure, imshow (I);
hold on;
for id = 1: length (id)

h = rectangle ('position', statsdefects (id) .BoundingBox, 'EdgeColor',
'r', 'LineWidth', 2);
```

```
set (h, 'EdgeColor', [0.95 0 0]);

hold on;
end
title (['There are', num2str (numPests), 'parasites in the image']);

holdoff
```