

December 2013

Computer Vision Algorithms For An Automated Harvester

Alireza Masoudian

The University of Western Ontario

Supervisor

Dr. Kenneth A. McIsaac

The University of Western Ontario

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Alireza Masoudian 2013

Follow this and additional works at: <http://ir.lib.uwo.ca/etd>

Recommended Citation

Masoudian, Alireza, "Computer Vision Algorithms For An Automated Harvester" (2013). *Electronic Thesis and Dissertation Repository*. 1804.
<http://ir.lib.uwo.ca/etd/1804>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.

COMPUTER VISION ALGORITHMS FOR AN AUTOMATED HARVESTER

(Thesis format: Monograph)

by

Alireza Masoudian

Graduate Program in Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Graduate Program in Engineering Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Alireza Masoudian 2013

Abstract

Image classification and segmentation are the two main important parts in the 3D vision system of a harvesting robot. Regarding the first part, the vision system aids in the real time identification of contaminated areas of the farm based on the damage identified using the robot's camera. To solve the problem of identification, a fast and non-destructive method, Support Vector Machine (SVM), is applied to improve the recognition accuracy and efficiency of the robot. Initially, a median filter is applied to remove the inherent noise in the colored image. SIFT features of the image are then extracted and computed forming a vector, which is then quantized into visual words. Finally, the histogram of the frequency of each element in the visual vocabulary is created and fed into an SVM classifier, which categorizes the mushrooms as either class one or class two. Our preliminary results for image classification were promising and the experiments carried out on the data set highlight fast computation time and a high rate of accuracy, reaching over 90% using this method, which can be employed in real life scenario.

As pertains to image Segmentation on the other hand, the vision system aids in real time identification of mushrooms but a stiff challenge is encountered in robot vision as the irregularly spaced mushrooms of uneven sizes often occlude each other due to the nature of mushroom growth in the growing environment. We address the issue of mushroom segmentation by following a multi-step process; the images are first segmented in HSV color space to locate the area of interest and then both the image gradient information from the area of interest and Hough transform methods are used to locate the center position and perimeter of each individual mushroom in XY plane. Afterwards, the depth map information given by Microsoft Kinect is employed to estimate the Z- depth of each individual mushroom, which is then being used to measure the distance between the robot end effector and center coordinate of each individual mushroom. We tested this algorithm under various environmental conditions and our segmentation results indicate this method provides sufficient computational speed and accuracy.

Keywords: Harvesting robot, Vision system, Image classification, Hough transform

Acknowledgments

I would like to offer my sincerest gratitude to my supervisor, Dr. Kenneth A. McIsaac, not only for enthusiastically discussing my work with me, answering my questions and giving me very valuable advice, but also for his patience and brilliant personality. One simply could not wish for a better and friendlier supervisor.

I would like also to thank my parents, M. Masoudian and F. Haji because without them I would not be here, my sisters (Maryam, Mahboubeh and Marjan) for all of their support. I cannot find a proper word to express my gratitude to them.

Table of Contents

| | |
|--|------|
| Abstract | ii |
| Acknowledgments..... | iii |
| Table of Contents | iv |
| List of Tables | vii |
| List of Figures | viii |
| Nomenclature | xi |
| Chapter 1 | 1 |
| 1 Introduction | 1 |
| 1.1 General Introduction | 1 |
| 1.1.1 Digital Image Processing | 1 |
| 1.1.2 Vision Guided Robotic Systems | 2 |
| 1.1.3 Machine Learning | 3 |
| 1.2 Automated Mushroom Harvesting..... | 4 |
| 1.3 Literature Review..... | 6 |
| 1.3.1 Fruit Harvesting Robots | 7 |
| 1.3.2 Mushroom Harvesting Robots | 13 |
| 1.3.3 Mushroom Microbial Spoilage Detection..... | 15 |
| 1.3.4 Cell Segmentation | 16 |
| 1.4 Thesis Contributions | 17 |
| 1.5 Thesis Outline | 18 |
| Chapter 2 | 19 |
| 2 Image Feature Extraction Using SIFT and BOW Algorithms | 19 |
| 2.1 Introduction..... | 19 |
| 2.2 Image Pre-Processing..... | 20 |

| | |
|--|----|
| 2.3 Image Feature Extraction | 21 |
| 2.3.1 SIFT Parameter Selection | 24 |
| 2.4 Feature Vector Quantization | 25 |
| 2.5 Conclusion | 28 |
| Chapter 3 | 29 |
| 3 Application of Support Vector Machine to Detect Microbial Spoilage | 29 |
| 3.1 Introduction | 29 |
| 3.2 SVMs Based Classification Approach | 29 |
| 3.2.1 SVMs Parameters Selection | 31 |
| 3.3 Experimental Results | 35 |
| 3.4 Conclusion | 41 |
| Chapter 4 | 42 |
| 4 Segmentation Algorithm for Mushroom Identification | 42 |
| 4.1 Introduction | 42 |
| 4.2 Segmentation Based on HSV Color Space and K-Means | 43 |
| 4.3 Mushroom Size Detection | 52 |
| 4.4 Conclusion | 62 |
| Chapter 5 | 63 |
| 5 Application of Infrared and Stereo Cameras in Mushroom Harvesting | 63 |
| 5.1 Introduction | 63 |
| 5.2 Image Depth Extraction | 63 |
| 5.3 Temperature Information | 66 |
| 5.4 Conclusion | 71 |
| Chapter 6 | 72 |
| 6 Conclusion | 72 |
| 6.1 Introduction | 72 |

| | |
|------------------------------------|----|
| 6.2 Summary | 72 |
| 6.3 Conclusions..... | 73 |
| 6.4 Contributions..... | 74 |
| 6.5 Proposal for Future Works..... | 75 |
| References..... | 76 |
| Curriculum Vitae | 81 |

List of Tables

| | |
|---|----|
| Table 2.1: The parameters of the SIFT algorithm used in this experiment | 24 |
| Table 3.1: Different kernels' hyperparameters | 34 |
| Table 3.2: Confusion matrix | 35 |

List of Figures

| | |
|--|----|
| Figure 1.1: Mushroom growing bed | 6 |
| Figure 1.2: Fruit tracking system overview in [12] | 10 |
| Figure 1.3: Mushroom harvesting vision system in [25] | 14 |
| Figure 1.4: Curve searching method in [38] | 16 |
| Figure 2.1: Overview of the classification approach | 19 |
| Figure 2.2: Median filter operation | 21 |
| Figure 2.3: Keypoints detection on DoG Images..... | 23 |
| Figure 2.4: SIFT descriptor construction in [13] | 24 |
| Figure 2.5: Histogram of visual words construction..... | 26 |
| Figure 2.6: SVM accuracy versus BOW..... | 27 |
| Figure 3.1: SVMs approach overview | 29 |
| Figure 3.2: 10-Folds Cross Validation..... | 32 |
| Figure 3.3: Parameters for polynomial Kernel (big scale)..... | 33 |
| Figure 3.4: Parameters for polynomial Kernel (medium scale)..... | 33 |
| Figure 3.5: Parameters for polynomial Kernel (small scale) | 34 |
| Figure 3.6: Classification Result..... | 36 |
| Figure 3.7: Classification running time | 36 |
| Figure 3.8: Linear classifier result | 37 |
| Figure 3.9: Polynomial classifier result | 38 |

| | |
|---|----|
| Figure 3.10: RBF classifier result | 38 |
| Figure 3.11: Sigmoid classifier result | 39 |
| Figure 3.12: ROC curve for polynomial kernel | 40 |
| Figure 3.13: ROC curve for RBF kernel..... | 40 |
| Figure 3.14: ROC curve for sigmoid kernel | 41 |
| Figure 4.1: RGB channels of sample one | 43 |
| Figure 4.2: RGB channels of sample two | 43 |
| Figure 4.3: HSV color space in [45] | 44 |
| Figure 4.4: HSV sample image of mushroom | 44 |
| Figure 4.5: HSV channels of sample one..... | 45 |
| Figure 4.6: HSV channels of sample two | 45 |
| Figure 4.7: Cluster index for image one | 47 |
| Figure 4.8: Cluster index for image two | 47 |
| Figure 4.9: Intensity histogram of the image | 48 |
| Figure 4.10: Image thresholding | 48 |
| Figure 4.11: Mathematical morphology | 49 |
| Figure 4.12: The resulted image (sample one) after morphological operations | 49 |
| Figure 4.13: The resulted image (sample two) after morphological operations | 50 |
| Figure 4.14: Contour extraction for sample one | 51 |
| Figure 4.15: Contour extraction for sample two | 51 |

| | |
|--|----|
| Figure 4.16: Binary image converted from grayscale image | 52 |
| Figure 4.17: Mushroom size detection overview | 54 |
| Figure 4.18: Construction of the accumulation array in [49]..... | 56 |
| Figure 4.19: Segmentation result for first image | 59 |
| Figure 4.20: Accumulation array for first image | 59 |
| Figure 4.21: 3D view of the accumulation array for first image | 59 |
| Figure 4.22: Segmentation result for second image..... | 60 |
| Figure 4.23: Accumulation array for second image..... | 60 |
| Figure 4.24: 3D view of the accumulation array for second image..... | 60 |
| Figure 4.25: Segmentation result for third image | 61 |
| Figure 4.26: Accumulation array for third image | 61 |
| Figure 4.27: 3D view of the accumulation array for third image | 61 |
| Figure 5.1: Original Image for depth extraction | 64 |
| Figure 5.2: Depth of the mushroom image | 64 |
| Figure 5.3: Simulink function to extract the XYZ position | 65 |
| Figure 5.4: Thermal image taken by infrared camera (1) | 66 |
| Figure 5.5: Thermal image taken by infrared camera (2) | 67 |
| Figure 5.6: Thermal image taken by infrared camera (3) | 68 |
| Figure 5.7: Thermal image taken by infrared camera (4) | 69 |
| Figure 5.8: Thermal image taken by infrared camera (5) | 70 |

Nomenclature

| | |
|------|------------------------------------|
| SVM | Support Vector Machine |
| DOG | Difference of Gaussian |
| BOW | Bag of Words |
| MDS | Multi-Dimensional Scaling |
| HT | Hough Transform |
| ROC | Receiver Operating Characteristics |
| LOG | Laplacian of Gaussian |
| HSV | Hue, Saturation, Value |
| SIFT | Scale Invariant Feature Transform |
| EER | Equal Error Rate |

Chapter 1

1 Introduction

1.1 General Introduction

1.1.1 Digital Image Processing

Digital image processing [1] could be explained as a series of features and attribute-extraction processes from images in order to detect and analyze individual objects in images. Digital image processing can be categorized into two main groups: Image analysis and computer vision. Image analysis mostly aims at utilizing different image processing techniques such as noise filters to enhance the contrast or quality of the image while on the other hand, computer vision concentrates on methods such as learning algorithms to make the detected objects in the image understandable or sensible or easier for human intelligence to assimilate. There are three levels of computerized processes in the field of computer vision and depending on the application, each of them could be employed. In this thesis, we mainly concentrate on low and mid-level processes. Low-level process in general includes fundamental operations such as image noise filtering and enhancements to make the image ready for further processing while mid-level process mostly involves image segmentation and classification techniques. In the mid-level process, the outputs are features and information extracted from input images unlike that obtained from the low level process. This extracted data can be widely used to find the characteristics of objects in image such as an object's dimension.

1.1.2 Vision Guided Robotic Systems

Computer vision [2] could be defined as the science of deploying different methods and algorithms to take, process and analyze images in order to generate appropriate numerical data and information; and there are various mathematical techniques and methods, which could be employed appropriately to extract high dimensional data and information from images. Computer vision has many applications in industries such as mobile robot navigation, controlling processes with industrial robots and quality control to check dimension, color and angles of objects. The main concept of vision-guided robotics is to design a vision system and control software capable of recognizing and analyzing different objects in the image in order to give the robot the accurate coordinate of the component. This information can be used by the robot's controller to guide the robot's mechanical arm to move to the object precisely.

Vision-guided robots [2] have many advantages over the traditional automation systems. One of which is the flexibility and efficiency, which means that many additional operations can be integrated to the current task of the robot without the need of adjusting the mechanical parts. A vision-guided robot has two main components; Vision system and robotic system. Camera selection is an important task as pertains to the vision system. Different applications need different vision systems or different cameras so to say, so choosing a camera which can capture images containing useful and discriminant features has a high impact on the whole system performance. Depending on the application being used, there are five major vision systems: Digital camera, stereo camera, Hyperspectral camera, Infrared and Laser scanner. Emphasis will not be put on infrared and laser scanner in this thesis. The stereo vision system and digital camera aid in real time identification and classification of mushrooms.

1.1.2.1 Stereo Vision System

Stereo vision [2] is a technique which aims at determining the depth from two or more cameras. In other words, using two or more images of the same scene or objects, stereo vision can compute the three dimensional information of each pixel in the image. Using

two cameras, it is possible to find the corresponding points in the two images by triangulation and then finally extract the depth map of the scene.

The depth map [2] is equal to the distance from the camera's sensor to every pixel in the image. Stereo vision has a wide range of applications such as robotics, where it is used to extract the three dimensional coordinates of each object in the surrounding area of the robot to avoid it crashing with obstacles and to guide the robot to the correct path.

In this thesis, we investigated one of its major applications which is to extract the depth of the object in the image and therefore find the X, Y, Z real world coordinates of each pixel in millimeters. As the objects in image have different depths, stereo vision information could be used to separate overlapping image objects which the robot may otherwise not be able to distinguish as a separate object. The robot's controller is able to guide the end effector to reach its object in the environment by finding the Z- depth of the object.

1.1.2.2 Hyperspectral Imaging

Hyperspectral imaging [3] is a non-destructive technique also known as spectroscopic imaging. It deals with the collection of image data in hundreds of contiguous spectral bands. In other words, Hyperspectral imaging divides the spectrum into more bands, which make it possible to extract a continuous spectrum for each image cell. The number of spectral bands imaged simultaneously produces a vast amount of data, which when interpreted appropriately can be used in several applications such as quality control. For example, in quality control application, to determine the sub-pixel concentrations of materials, unknown spectra can be compared against spectral libraries. In this thesis, we have proposed a new method that can be used in quality control without the use of Hyperspectral camera with the details being expanded in Chapter two and three.

1.1.3 Machine Learning

Machine learning [4] is a set of methods for the automated analysis of structure in data. In other words, with a given set of large data, machine learning algorithms can be trained on the data to create a predictive model which can be employed for a specific task. The

learning step can be categorized into two groups; supervised learning and unsupervised learning. In supervised learning, the desired output is known while the algorithm is trained on labeled data with a class or value. The algorithm then constructs a mapping function from the input to the output such that prediction of output labels is made possible from unseen inputs. In unsupervised learning on the other hand, the desired output is unknown and the data is unlabelled, or put in another way, the values are unknown. The objective of unsupervised learning is to construct a structure in the data by implementing methods like K-means [39], genetic algorithms and clustering approaches. Machine learning algorithms are very widespread and are widely used in vision guided robotics as seen in this thesis. In this thesis, machine learning algorithms are used to categorize the mushrooms into healthy and unhealthy groups as explained in Chapter three.

1.2 Automated Mushroom Harvesting

The need to produce higher quality products and reduce production cost is fast becoming a concern in the agricultural industry, and a way to solve this problem is to replace humans with robots and automated systems because of the inherent advantages like better performance, higher accuracy and production efficiency. Harvesting mushrooms has always been a major challenge in the agricultural industry because of the sensitivity required to pick them. To solve this problem, automation can be used because harvesting robots can decrease production cost while increasing efficiency. Robots of this kind have existed for over the past few years and they come in a wide variety of mechanisms and technologies. Using these robots which are easily adaptable, in that they are capable of performing different plans and strategies in harvesting and working under any severe environmental conditions and for 24 hours cycle time, has many advantages such as increasing production efficiency and market supply of mushrooms while lowering labor cost simultaneously [24]. Also the yield directly depends on picker skill and picker training takes very long. It should also be mentioned that mushrooms grow 10% by mass per hour so 24 hour picks is required to minimize spoilage.

Designing an accurate vision system capable of identifying and tracking mushrooms is a major challenge because of reasons such as irregularities in growth pattern and bad

illumination in the growing environment. As such, different agricultural harvesting robots need different sensors based on the problem that they might face. For example, a mushroom harvesting robot should be able to identify the ripeness and dimension of mushrooms as well as the location of each mushroom. This requires the color information of each mushroom or its boundary information and a 3D vision system that capable of navigating the robot and tracking mushroom locations. In order to successfully navigate and track under different environmental conditions, a robust vision based system combined with different sensors needs to be employed.

Picking fresh mushrooms from the environment entails a series of consecutive tasks either done by hand or by robot. To design and implement a robust automatic system, attention should be paid to three key factors. The first and most important is to create an algorithm to locate and identify mushrooms in any size and shape and under any environmental condition such as in darkness or when illumination changes. A three-dimensional mushroom center location should be calculated to get the position because a mushroom harvesting robot should be able to guide its mechanical arm towards each individual mushroom in the growing bed, in addition to other features such as radius and ripeness. Secondly, defining an accurate path plan to guide the robot through the mushroom growing beds is another key factor. Finally, the design of a gripper capable of not only grasping the mushroom in a manner such that it does not damage or contaminate it or its neighbours but eventually carefully places it into a container is required.

The mushroom (*Agaricus bisporus*) crop develops in a series of “flushes” almost weekly inside growing beds as shown in the Figure 1.1 Each flush of mushrooms is harvested over a period of three to four days where up to four flushes are harvested in each growing cycle and the yield of the crop depends on the skill of the picker. As can be seen in the figure, since mushrooms grow at irregular rates and intervals, segmentation becomes challenging because the mushrooms of different sizes, levels of ripeness and at irregular spacing from each other. There is a clear market for the development of a high-speed picking machine that can optimize yields from mushroom growing beds using a recognition system to identify the right mushroom to pick and a mechatronic system able

to pick mushrooms at high-speed while limiting damage, all while working in situ so as not to affect the growing environment required for high-density mushroom cultivation.



Figure 1.1: Mushroom growing bed

There are a number of cues that can be used to identify mushrooms for optimal picking, including cap colour, size, texture, shape and temperature. It will also be necessary for the picking robot to have the ability to autonomously identify diseased mushrooms that need to be quarantined before the entire crop is contaminated. More emphasis is put on identifying ripe mushrooms, mushroom recognition and classification in this thesis.

1.3 Literature Review

Studies have been carried out over the past years in the areas of fruit segmentation and vision systems of automatic harvesters, which are similar to the method used for accurately segmenting and recognizing each mushroom and determining the mushrooms to be picked by the robotic harvester.

In this section, methods employed to automate the location and identification of fruits, mushroom microbial spoilage detection and mushroom harvesting robots using computer vision algorithms will be discussed, with highlights on sensor selection, image processing algorithms, characteristics of each method, as well as strategies to identify the fruits.

1.3.1 Fruit Harvesting Robots

Over the past few years, considerable attention has been devoted to the development of an advanced robotic system that enables autonomous fruit harvesting. The application of an automated system to pick fruits was first proposed and implemented by Parrish and Goksel [5]. They used a B/W camera and a filter in their experiment in order to increase the contrast between the apple and background. The image was separated using a threshold process into bright and dark pixels, and a morphological operation was then carried out on the bright pixels to remove the inherent noise and inappropriate details from the image. Finally the center point and radius of the apple were taken by calculating the difference between the length of each region's horizontal and vertical extremes. In order to measure the dimension of each apple or in general to measure its density, a circular window with size computed as the extreme mean value was placed over the segmented apple. The region was accepted as a real apple if the user's defined threshold value was smaller than region intensity.

Another study to detect apples was carried out by Grand Esnon [6] who used a colored camera mounted on a robot arm. The color information from the apple's color was used to separate the image's background from the apples, however his method was not robust enough under illumination changes: This was the major drawback of the algorithm as false detections caused by illumination differences were possible. A similar study was proposed by Rabatel [7], which was based on the analysis of three spectral bands taken from three-colored cameras with the help of three different optical filters, which were used to create three complementary colored images. Spectral color band information from the three spectral bands was taken at 95 nm because it was discovered that apple leaves had the same reflectivity, thereby making the system immune to illumination changes. However the reported successful recognition rate was only 50%.

Whittaker [8] further proceeded in the research into automated fruit harvesting using a vision system by proposing a new method to identify and locate tomatoes on vines. This method was not sensitive to illumination changes and therefore insensitive to the color of the tomatoes since it only used the shape information of tomatoes to recognize it. An image gradient map and contours of the tomatoes were computed by using Sobel filter on

the images taken by a CCD camera. To detect the circular shape of the tomatoes, Hough transform was applied and then optimized. This method was highly dependent on user parameter selection as users specify the threshold value in the Hough transform method. This algorithm was not sensitive to environmental conditions, however the correct detection rate was only 60%. The problem responsible for the 40% false detection was because of the user's error in specifying the threshold value and therefore, detecting contours of leaves as fruits. Apart from that, this algorithm was also reported to be time consuming and not feasible for real time application. As no depth detection sensor was used, implying that the distance between the robot arm and the tomato was not known, the robot arm had to keep moving forth and back till it sensed or touched tomatoes.

Slaughter and Harrell [9] developed a citrus harvesting robot capable of locating and recognizing citrus. In their approach, the color information of the RGB image taken from the citrus was used to separate it from the background. They used a Bayesian classifier to categorize different pixels using the image color information. One important drawback of their method is the fact that RGB color space can only be applied on certain types of fruits, such as oranges, that are easily distinguishable from background. Their reported performance was quite satisfactory, obtaining a 75% correct detection over the total visible fruits.

Grasso and Recce [10] improved on the fruit detection system. They developed an orange harvesting robot capable of picking up oranges using a stereo vision system. Two different cameras were mounted on the robot end effector to get two views from different angles. These two images were used to perform a stereo vision algorithm in order to get the location of each fruit. After preprocessing the images, a Hough transform method, similar to that used by Whittaker, was performed to find the center coordinate of each orange. The authors reported two major problems; first, false detection occurred in cases where the oranges were partially occluding each other, and second, a leaf was sometimes detected as an orange due to the similarity in colors. A similar robot was developed by Ceres [11] to locate oranges but contrary to what Grasso and Recce did, a laser range finder was employed to get depth images. Data gotten by the laser combined with fruit shape information were used to perform a selective harvesting strategy such that only

fruits from a certain dimension (shape information) or certain maturity level (reflectivity information) could be harvested.

Jednipat Moonrinta and his team [12] employed a combination of the feature extraction method and the classification algorithm in order to detect fruits and reconstruct a three dimensional image for crop mapping. They used a monocular video camera connected to a mobile inspection platform to get their image data. A series of experiments were carried out using feature descriptors such as SIFT [13] (Scale Invariant Feature Transform) and Harris methods [14], feature classifiers using SVM [15] and 3D reconstruction using structure from ellipsoid estimation techniques [17].

As shown in Figure 1.2 below, SIFT and SURF [18] methods were used individually to extract fruit feature keypoints and descriptors and their results were compared. SVM classifier was used to classify the image feature vectors and to detect the fruity part of the image. To find the SVM hyper-parameters, a grid search was performed and a cross validation method was used to choose the best set of hyper-parameters. Any point not on the fruit surface is classified as negative and is ignored while mathematical morphological operations are carried out on the remaining points which are on the fruit surface, which helped to remove small regions. In order to model the three dimensional structure of the fruit's surface, a three dimensional point cloud estimation method based on motion techniques was employed on the image data and finally, the point cloud data for each fruit region was modeled and the fruit's surface was estimated by fitting an ellipsoid to the data.

Although these methods were highly accurate (the authors reported 91% accuracy), the weather condition and illumination changes were not put into consideration, and these have the tendency to reduce the accuracy of fruit detection.

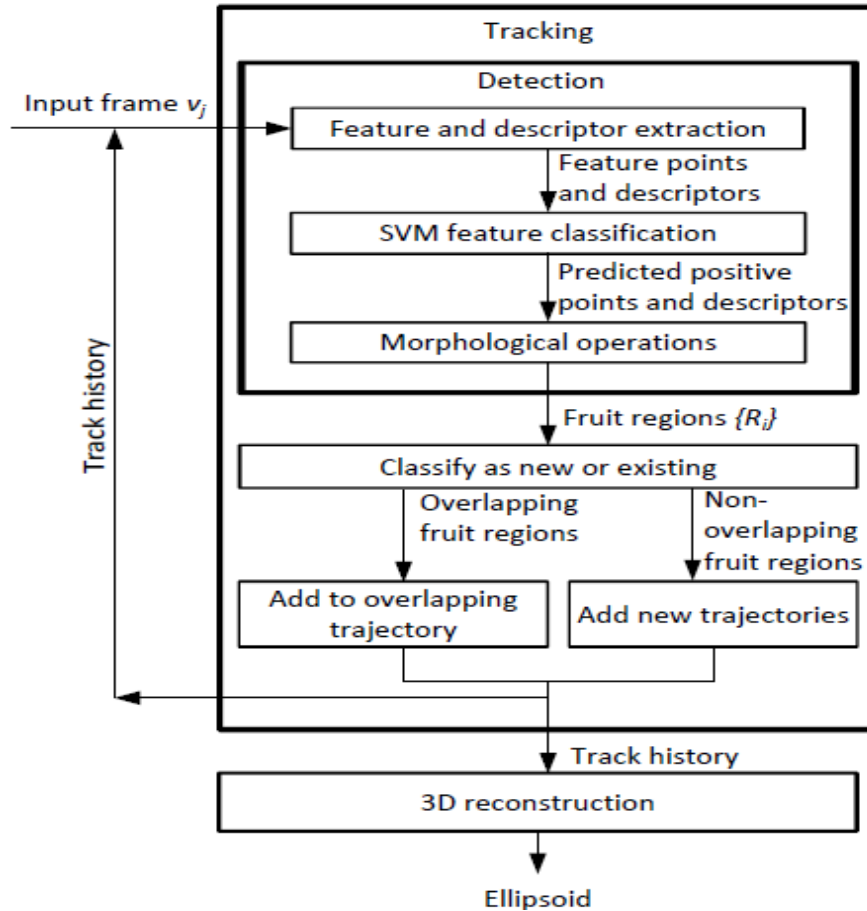


Figure 1.2: Fruit tracking system overview in [12]

A series of research [19]-[24] has been carried out recently on fruit recognition and localization that are all aimed at using a combination of features such as color and shape information to distinguish the fruit from the background. A common method used in all these experiments is to utilize information taken from image color spaces such as RGB or HSV to classify the objects in the image into different regions. For example, cucumber identification in green house was the purpose of one study in which the G component of the image's RGB color space was extracted and used as the threshold segmentation channel. The remaining pixel regions were removed by mathematical morphological operations. Another method applied on the vision system of an apple harvesting robot was machine-learning algorithm (SVM) to separate apple from the image background and other objects in the scene. It was found that using feature extraction method such as

SIFT as well as classification methods such as SVM could increase the apple recognition accuracy used in the apple harvesting robot.

There are three major parts that received attention in all the previous experiments and research in the design of a vision system for the automated harvesters. These steps can be categorized as follows [24];

- 1) Image acquisition methods
- 2) Pre-processing to enhance image quality for further actions and
- 3) Analyzing the image to detect and extract the object from the image background.

The sensors that have been used mostly in all the previous experiments were color camera, laser and stereo vision system. Some researchers used filter and other image enhancement tools to increase the contrast and resolution of the image. Some other studies employed laser range finders to measure the distance between the robot's end effector and the object on the scene. Stereo cameras have been used recently in some vision systems of robotic harvesters to take the depth map of the scene and compute the real world coordinate of the image pixels in order to locate the object in the image.

The types of the images that have been used in previous studies can be categorized into three main groups [24]:

- 1) Intensity: a high resolution color camera used to take this kind of image.
- 2) Spectral: this kind of image was obtained in cases where a specific wavelength should be used in order to get the reflected intensity. Hyperspectral technology was widely used in fruit disease detection and to ensure only high quality fruits reach the market.
- 3) Depth: Depth maps are mainly used to find the distance between the camera sensor and the object in the scene. There are many techniques to obtain the depth of the image. The main and major methods are stereo vision system and laser scanner which both are able to measure the depth and therefore the distance between the robot mechanical arm and the object on the scene. Intensity and spectral methods are only useful in cases where

brightness is high enough to recognize the object features, however shadow is a problem for color camera. The effect of shadow on the image can be diminished by using only the color information and ignoring the intensity channel of the image. The use of laser scanner and infrared camera can solve the problems caused by shadow and illumination changes. Moreover the data taken from laser does not suffer from change in illumination levels and it can be used to find the depth map of the image and therefore obtain the 3D position of the object in the scene which is sent to the robot's controller.

The analysis strategies that have been used to process the images can be explained in two main categories according to the features extracted from the image [24]:

1) Local features

2) Shape features

1) Local features involve image processing methods which utilize value of the each pixel in the image such as color or intensity to decide which part of the image belongs to fruit. In other words, pixel values and color components in the image taken by a spectral camera is enough to recognize fruits. The color information of the image can be particularly extracted by using other color spaces such as H^*S^*V or L^*a^*b . Fruit recognition and localization based on local features require less processing time and simple algorithms can be used. These algorithms have the advantage of being used in real time applications when processing time is an important factor. However, illumination changes can severely affect the performance of these methods as the threshold or other constant values need to be updated early in order to adapt with the new illumination condition.

2) Shape features are based on the methods that analyze region convexities or objects' contours to identify mathematical shapes such as ellipses and circles. Unlike local feature techniques, these methods are not affected by illumination changes, as they are independent of the image's color or intensity information so they are more general than local feature techniques. However, one of the disadvantages of using this technique is the long processing time caused by the mathematical computations, which makes it hard to

be employed in real time applications. Unless acceleration hardware is included, these methods require increased the processing time. Shape feature base methods have the advantage of recognizing fruit regardless of its color or intensity but increase false detections. The majority of recent experiments and research used both depth data and shape features to create a learning algorithm. The range information is independent of color or intensity data and its combination with shape feature based methods obtains good results.

In this section, a series of experiments on designing vision system for fruit automated harvesters were reviewed. Three main types of images [24] were used in these experiments: Intensity, spectral and depth image taken by color camera, spectral camera and stereo respectively. The analyzing and processing steps are highly dependent on the type of image that is taken from the crop; as such the type of the image is an influential factor on the final results. So the sensors that are used for taking the images have an important impact on the performance of the algorithm. Therefore, the appropriate sensor needs to be selected in order to take images containing useful and stable information that enhances the image analysis and true detection. One of the serious problems that affect the segmentation performance of all the previous research is occlusion and the fact that fruits or in our case, mushrooms, usually overlapped each other. This overlap can make the segmentation problem more challenging by hiding section of shapes.

1.3.2 Mushroom Harvesting Robots

A limited number of studies have been performed in the area of mushroom harvesting robots [25]-[29]. Researchers at the University of Warwick in collaboration with Silsoe Research Institute developed the first robot prototype for mushroom harvesting. This robot as shown in Figure 1.3 used a monochrome camera to spot and select mushrooms in the growing bed and this camera was fixed approximately 1 meter above the mushroom's growing bed. In order to provide a uniform light level over the growing bed, a 32 W fluorescent light was used.

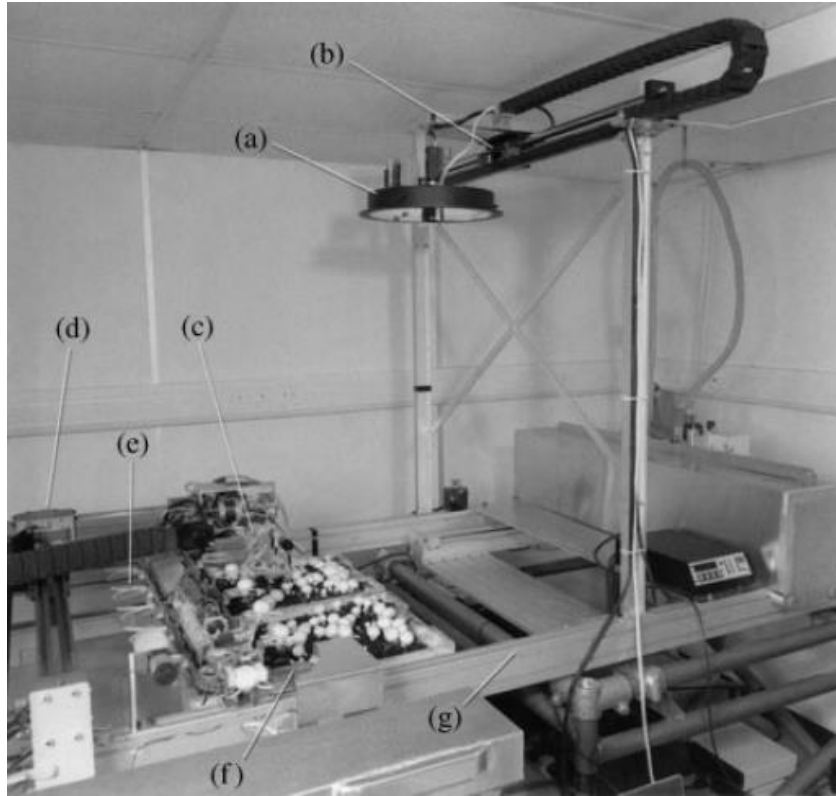


Figure 1.3: Mushroom harvesting vision system in [25]

The intensity information of the image was used to recognize the mushroom's cap and then locate the center coordinate of each individual mushroom. The mushroom's cap has a higher intensity than the mushroom edges and casing layer (background) because it serves as a reflector for the overhead lightning system; this helped in locating the X, Y-center coordinate of each individual mushroom. To find the mushroom's contour, the center position was used as the starting point and the algorithm then searched radially looking for pixels with lower intensities than the center point. The huge gap in pixel intensity was considered as a valley between two adjusting mushrooms. A suction cup at the end of the robot arm picked the identified mushrooms. The accuracy of this algorithm in detecting mushrooms was reported to be 84%.

A similar prototype was developed by the Centre for Concepts in Mechatronics (CCM) [30] in Netherlands with similar suction mechanism. They designed a new gripper that required both camera vision and laser beam for effective and accurate recognition of mushrooms. Researchers at Zhejiang University in China [31] also developed the

theoretical basis for image processing to locate mushrooms in the growing bed. They employed boundary information and Fourier transform methods to calculate mushroom centroid coordinates, however their algorithm performance was not reported.

Our research includes both white and brown mushrooms as compared to other researches whose emphasis is on white mushrooms. The low contrast between the brown mushrooms and the substrate makes our research more challenging.

1.3.3 Mushroom Microbial Spoilage Detection

To make sure that only high quality mushrooms reach the market, and to ensure that diseased sections of the farm are properly quarantined, an evaluation capability needs to be developed as a component of a mushroom harvesting robot vision system. Mushrooms cultivated in growing beds, are susceptible to different kinds of diseases. One such disease is *Pseudomonas tolaasii*, a common kind of disease in the mushroom industry and the most important bacterium of *Agaricus bisporus* [32]-[33] which causes some brown spots on the mushroom caps. Previous research in the field of microbial damage detection of mushroom include the work of Felfoldi and Vizhanyo [34] who tested the potential of a machine vision system to recognise and identify brown blotch, which cause discoloration in mushroom caps.

At Dublin Institute of Technology, a non-destructive technique for quality grading of mushrooms based on hyperspectral image analysis [35]-[36] was presented using a combination of algorithms based on chemometric techniques and image processing methods. Hyperspectral imaging (HSI) has recently been introduced as another technique that combines imaging and spectroscopy to take the spectral information of an object which could be particularly used for food analysis; however HSI has not been used to detect damage of microbiological origin in industry. By using Hyperspectral imaging, acquiring an image can take from seconds to minutes depending on both the image quality and sample size while processing and classification time are largely dependent on computer capabilities. In our project, the robot requires image capturing to be taken in a few seconds. Therefore Hyperspectral imaging cannot be used in our application. As

such, a faster and more efficient method based on machine learning algorithms to detect mushroom diseases and classify them was employed.

1.3.4 Cell Segmentation

Because of the similarity in the circular shapes of mushroom caps and cell bodies, much of the work done in the biomedical imaging community on cell segmentation is relevant here.

A recent study on cell segmentation and cell counting was done by Park and his team [37] in which they used a combination of ellipse fitting and watershed methods to detect and count each individual cells in a microscopic image. An adaptive nonlinear filter was first applied to eliminate the noise from the image. To extract the cell's boundaries, Canny edge detector was used. An ellipse fitting algorithm was applied to each contour segment obtained from the Canny edge detector. The segmented region categorized into three groups; not ellipse, single cell and clustered cell. In order to separate the clustered cell into individual cells, the Watershed algorithm was applied on the occluded cells. The reported accuracy of 97% showed the strength of this algorithm, however the number of input data was unknown.

Another similar study was carried out by Xiaomin Li and his team [38], which was aimed at automatic cell counting and cluster segmentation. In his approach, concavity detection and ellipse fitting techniques were tested and demonstrated. First, a pre-processing operation was applied on the image to filter and extract the objects contour.

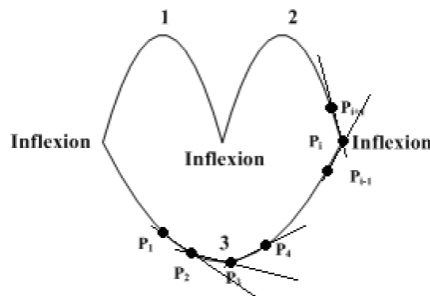


Figure 1.4: Curve searching method in [38]

Object edges were grouped into connected components and a curvature based curve searching algorithm was performed on each components. As shown in Figure 1.4, the main concept of curve searching algorithm was to find inflexions where the edge direction changes noticeably. After curve classification step, an ellipse fitting algorithm was applied on the outputted curves. Their experimental results show that this method is accurate but is limited by the quality of information gotten from the edges, which is directly proportional to the object's contour data.

In this Chapter, a complete literature review of experiments on designing vision system for fruit automated harvesters, automated mushroom harvesting robot and cell segmentation were reviewed. The information taken from previous research in the area of fruit segmentation and cell segmentation will be used and a new method to identify and locate mushroom as well as microbial damage detection will be proposed.

1.4 Thesis Contributions

The main contributions of this thesis are as follows:

- To present a new method using SIFT and BOW algorithms to detect and identify microbial spoilage of mushrooms in a mushroom harvesting robot vision system
- To develop an evaluation capability using Support Vector Machine (SVM) as a component for the mushroom harvesting robot vision system
- To propose a new method to recognize mushrooms and measure Z-depth coordinates of each mushroom in growing beds using a mushroom harvesting robot vision system
- To confirm the feasibility of the proposed methods by MATLAB Image Processing Toolbox
- To investigate the possibility of using infrared and stereo cameras as a piece of our vision system to identify grown mushrooms

1.5 Thesis Outline

The thesis is organized as follows:

- In Chapter Two, the vision system challenges are explained, the approach to extract features of images taken from mushroom growing beds is introduced and construction of visual words histogram and feature vector are presented.
- In Chapter Three, the feature vectors that are presented in Chapter two are classified using Support Vector Machine, the SVM's parameter selection approach is stated, the experimental work and classification results using Matlab software are presented
- In Chapter Four, the segmentation challenges are explained, the new method to recognize mushrooms and locate them in three dimensional coordinate is proposed, and the experimental work and segmentation results using Matlab software are presented.
- In Chapter Five, infrared and stereo vision systems are employed to investigate the possibility of using both depth information and temperature information of the image to enhance the identification strategy. Also, our experimental results using an infrared camera are shown.
- In Chapter Six, the contents of the thesis are summarized, the thesis contributions and conclusions are stated, and suggestions are made for future work.

Chapter 2

2 Image Feature Extraction Using SIFT and BOW Algorithms

2.1 Introduction

One of the important tasks a vision system of a mushroom harvesting robot carries out is the detection of mushroom damage either caused by microbial or mechanical origin where mushrooms are classified as either healthy or unhealthy. To solve the problem of identification, a fast and non-destructive method, SVM, is applied to improve the recognition accuracy and efficiency of the robot. As shown in Figure 2.1 below, a median filter is applied to remove the inherent noise in the colored image. After which, the SIFT features of the image are extracted and computed to form a vector, which is then quantized into visual words. In this Chapter, the proposed feature extraction approach is presented which involves using a combination of the SIFT algorithm, construction of visual words histogram and finally creating the image feature vector.

Unlike the other methods (hyperspectral imaging) previously discussed, this method does not suffer from the shortcomings like long processing time or high cost.

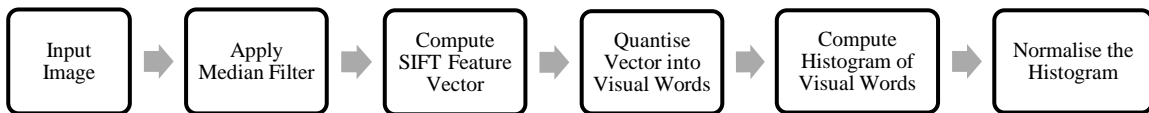


Figure 2.1: Overview of the classification approach

In this Chapter, challenges involved in the vision system will be covered, image pre-processing and image distinctive features extraction will be described, and feature vector construction will be covered in detail.

As pertains to mushroom classification on the other hand, the vision system aids in the real time identification and classification of mushrooms based on damage identified using the robot's camera. The non-uniformity of light reaching the growing bed must be taken into account, as mushrooms do not get the same amount of light, so the classification algorithm should not be affected by illumination changes when classifying mushrooms as either healthy or unhealthy. The primary objective of the healthy/unhealthy classification step is to identify diseased areas of the farm for quarantine in order to safeguard the remainder of the crop. As a result, performance must be fast enough to enable on-line use during mushroom harvesting but need not be fast enough to enable closed-loop visual servo style control.

In order to perform the classification algorithm, 240 sample images of mushrooms taken by digital camera is used. Our data set contains both healthy and unhealthy mushrooms images. In Chapter Three this data set will be used to perform classification algorithm.

2.2 Image Pre-Processing

To improve the quality of the acquired images of the proposed method and consequently the features extraction step, we used the median filter [39]. The median filter is applied to remove the noise from the original image, which included disturbances in the growing bed environment. The median filter allows removal of noise without significant blurring especially at edges, as the output of the filter is one of input values through the 5×5 window w size. In other words, there is no reduction in contrast through the window w since the output intensity values consists only of those present in neighborhood. The Figure 2.2 shows the operation of median filtering using a 5×5 sampling window with the border values unchanged.

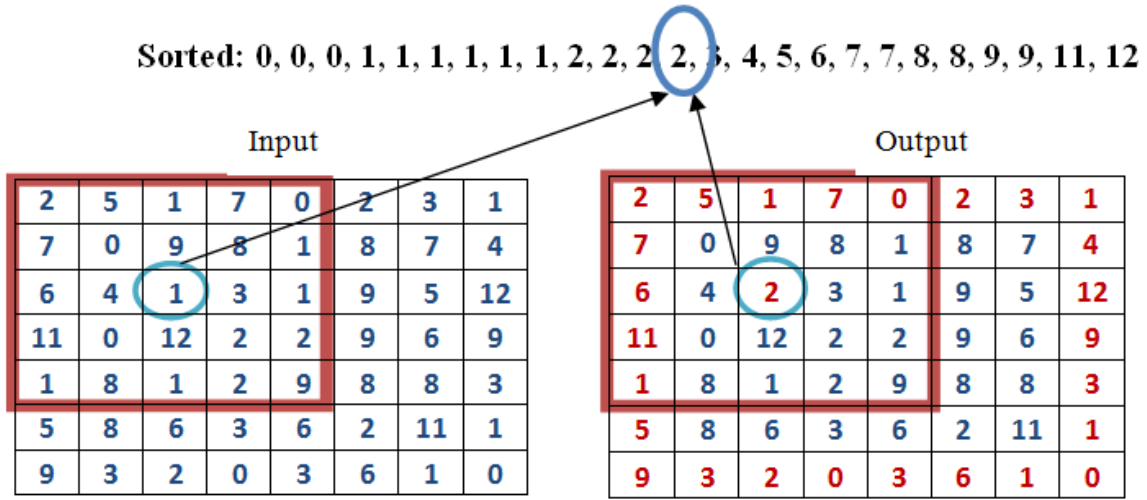


Figure 2.2: Median filter operation

In median filtering operation, the pixel values in the neighboring 5×5 window are ranked by intensity and the median value becomes the new value for the central pixel.

2.3 Image Feature Extraction

The SIFT algorithm [13] was used to extract the features from the mushroom images. SIFT is an appropriate method for our application because of some imaging challenges that are characteristic of the mushroom harvesting problem: the potential presence of different kinds of diseases, mushrooms in clusters partially occluding their neighbors, illumination change, different backgrounds, environmental noise and different image scale. The use of SIFT features can precisely increase the disease identification accuracy of the robot since extracted features are completely distinctive for each mushroom image.

The first step in the SIFT algorithm is to detect stable key points that are invariant with different scales of the image using scale space extremes in the difference of Gaussian (DoG) functions convolved with the image. A series of DoG images were constructed to extract the key points from the mushroom image. DoG image $D(x, y, \sigma)$ can be computed by defining the blur image $L(x, y, \sigma)$ which is obtained by the convolution of the image $I(x, y)$ with the Gaussian function $G(x, y, \sigma)$ as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) \times I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

The filtered original image of mushroom is convolved with Gaussian function and sigma σ is multiplied k times every process.

Adjacent images in the scale space are subtracted to create the DoG images. It should be noted that once the first pyramid is created the image is resized and a new pyramid for the second octave starts to be built.

For example if the mushroom image dimension is 512×512 then in the first octave all images are 512×512 . For the second octave the original image is down sampled and therefore all images in the second octave are 256×256 .

The key points which have the most information on series of DoG images are gotten by the Taylor expansion of the DoG pyramid by locating the extreme points and their scales. The extreme points are found by searching and comparing a pixel from the second image of the pyramid to its 26 neighbors in the 3×3 region in the upper and lower scale as shown in Figure 2.4. This procedure is repeated for second octave after finding the maxima in first octave is completed.

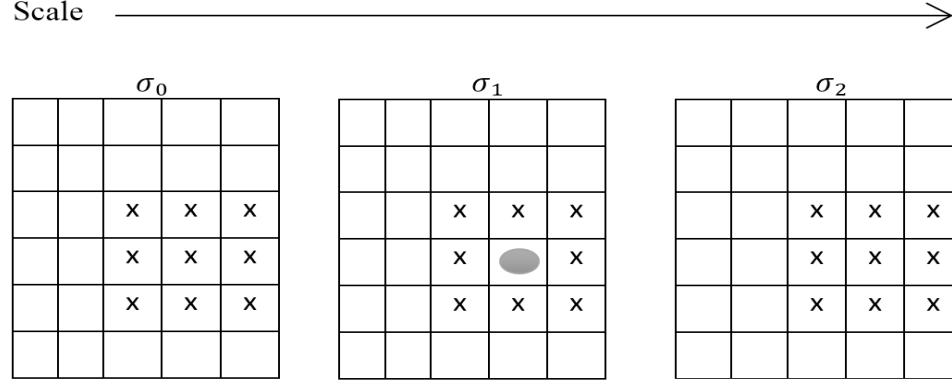


Figure 2.3: Keypoints detection on DoG Images

In order to keep the keypoint unaffected by the image rotation, an orientation is assigned to each keypoint. An orientation histogram with 36 bins covering 360 degrees is therefore created from the gradient orientation of samples extracted in the previous step within an area around the keypoint. The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are calculated as follows:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.1)$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (2.2)$$

Locating the highest peak in the orientation histogram and also other local peaks, which are 80% of the highest peak, multiple orientations can be assigned to keypoints. The aforementioned method computes the location, orientation and scale of SIFT features that have been found in the image which respond strongly to corners and intensity gradients. As shown in Figure 2.5, after creating the key points, a corresponding SIFT descriptor is computed around each key point's location. In this experiment, a 4x4 descriptor with 8-orientation histogram is calculated resulting in a vector dimension of 128.

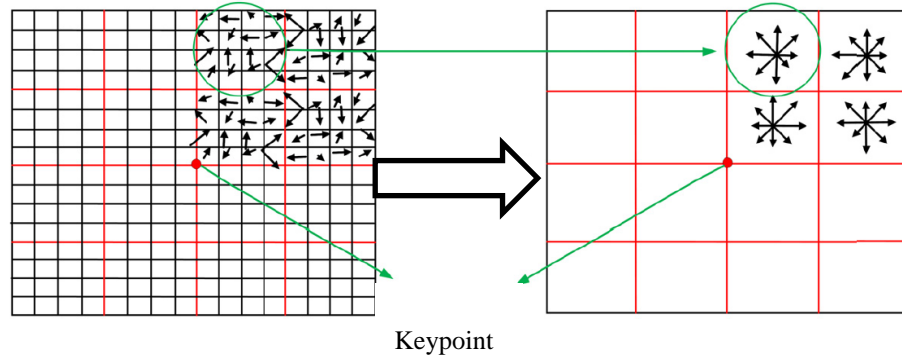


Figure 2.4: SIFT descriptor construction in [13]

2.3.1 SIFT Parameter Selection

Although the best set of SIFT parameters are given in the SIFT original paper [13], based on the image characteristics, some parameters are changed in order to improve the quality of the result in this thesis. In fact the SIFT algorithm has a set of parameters which can be varied based on the application and image characteristics that are being used.

Table 2.1: The parameters of the SIFT algorithm used in this experiment

| Parameter | Description | Value |
|--------------------------------|---|---------|
| Octaves | The number of octaves | Maximum |
| Intervals | The number of sampled intervals per octave | 3 |
| Sigma | The sigma value for initial Gaussian smoothing | 1.5 |
| Initial sigma | The assumed Gaussian blur for input image | 0.5 |
| Orientation histogram bins | The number of bins in histogram for orientation assignment | 36 |
| Orientation sigma factor | This determines the Gaussian sigma for orientation assignment | 1.5 |
| Descriptor histogram width | The height and width of the descriptor histogram array | 4 |
| Descriptor histogram bins | The number of orientation bins per histogram in descriptor array | 8 |
| Descriptor width | The height and width of the descriptor | 16 |
| Feature vector | The dimensions of the feature vector | 128 |
| Descriptor magnitude threshold | The threshold on the magnitude of the elements of the descriptor vector | 0.2 |

For example as shown in Table 2.1, the number of octaves being used in this experiment is the possible maximum number, in contrast with the original value, 3 which was given

by the Lowe's original algorithm, and the possible maximum number of octaves is directly related to the image dimension. Also the sigma value which has an important impact on the quality of the result was selected to be 1.5. In our implementation, a sigma of $S = 1.5$ is used because after systematic investigation, it was shown to be the best balance of runtime performance and classification accuracy

2.4 Feature Vector Quantization

The bag of words (BOW) algorithm [40] is a method that indexes local image features. In this application, it is used to count the frequency of each visual word and a set of high dimensional descriptor vectors can be coded to a number of visual words. Assigning high dimensional image features to a specific number of visual vocabularies has an important advantage in terms of computational cost, as such this method has many applications in object categorization and image classification. The low dimensional feature vector taken from bag of words method can be fed as an input vector to the machine-learning algorithm either for a supervised or unsupervised image classification.

In Figure 2.6, the procedure to construct the visual vocabulary and visual words histogram are demonstrated. Using this algorithm, the sampled features were clustered. In this way, the SIFT feature space is quantized into discrete number of visual words. In figure 2.6(a) the black circles illustrate the SIFT feature space and the ellipses on the images denote local feature regions. The cluster centers, called the visual word or visual vocabulary, was found using the K-means algorithm, where K value in K-means clustering algorithm is equivalent to the number of words shown in Figure 2.6(b). Once this quantized feature space is constructed, the features of the new given mushroom images shown in Figure 2.6(c) can be assigned to visual words by determining which cluster center in SIFT feature space the features are closest to. Finally the number of visual vocabularies for each image is counted and summarized in bag of word histogram demonstrated in Figure 2.6(d). The distance between the input mushroom image descriptors and cluster centers in SIFT feature space can be computed using Euclidean distance method.

The Euclidean distance (D) between two feature vectors P and F in feature space can be calculated as follows:

$$P = (p_1, p_2, \dots, p_n)$$

$$F = (f_1, f_2, \dots, f_n)$$

$$D(P, F) = \sqrt{(f_1 - p_1)^2 + (f_2 - p_2)^2 + \dots + (f_n - p_n)^2} = \sqrt{\sum_{i=1}^n (f_i - p_i)^2} \quad (2.3)$$

Finally a histogram is created that counts the frequency of each visual word in the mushroom image.

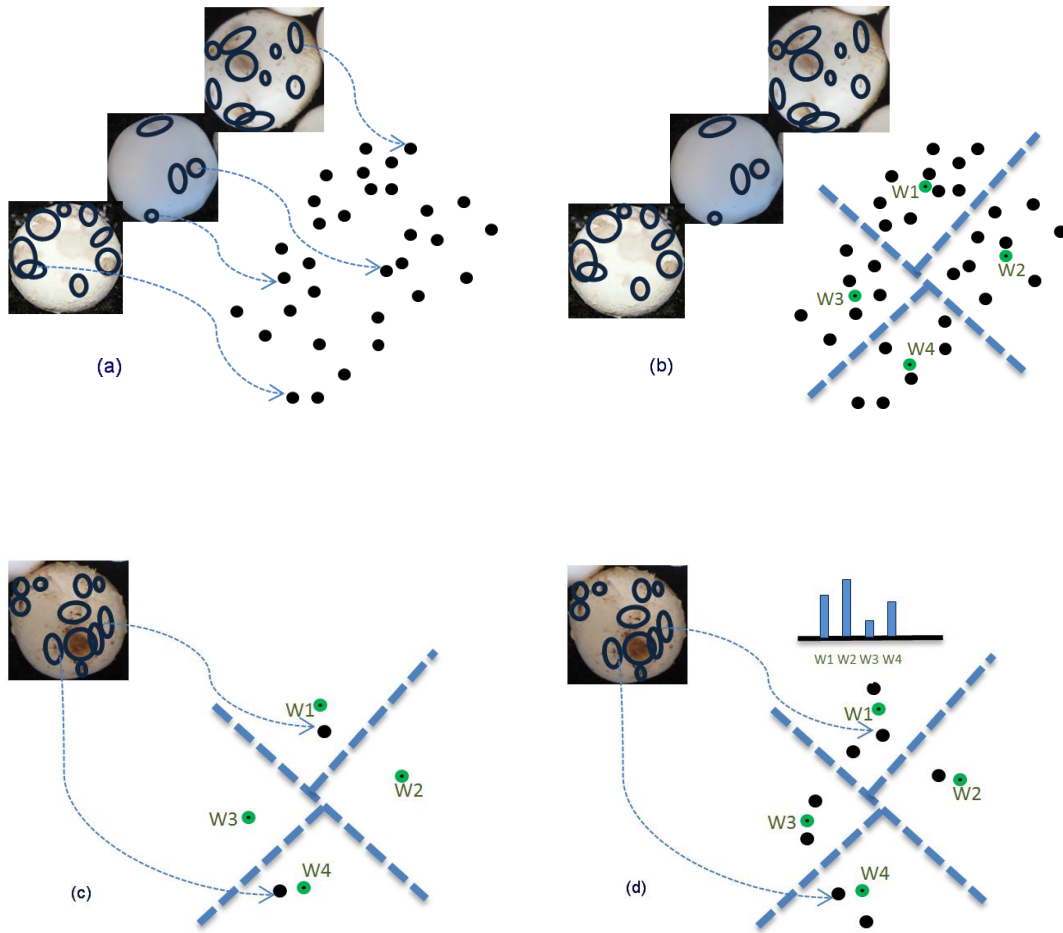


Figure 2.5: Histogram of visual words construction

In our implementation, a vocabulary of $V=50$ words is used because after systematic investigation, it was shown to be the best balance of runtime performance and classification accuracy. In K-means clustering algorithm, the number of cluster centers is a user-defined value determined by several trials and several experiments in order to get the best accuracy. Finding the precise number of clusters which gives us 100% accuracy when the number of clusters is high, however, is impossible. Chapter Three describes the reason for giving highest classification accuracy as 50, which is equal to the number of visual words.

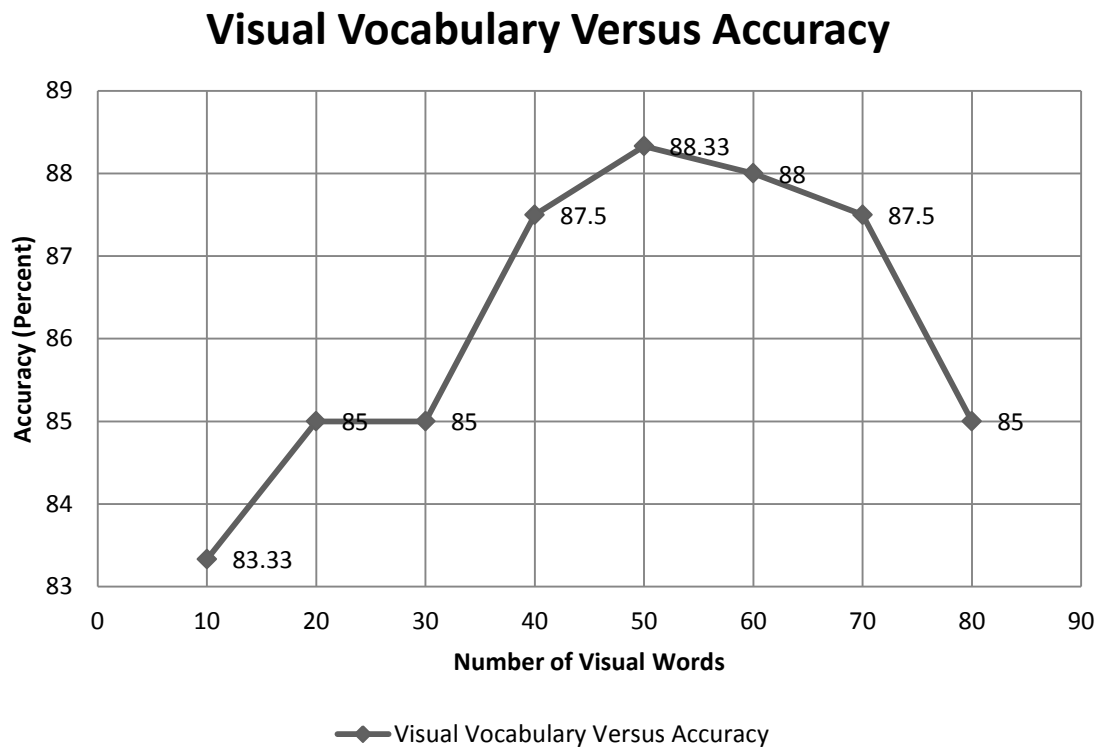


Figure 2.6: SVM accuracy versus BOW

As can be seen in the above Figure 2.7, in our application the SVM accuracy increases with the number of visual words. The highest SVM accuracy equal to 88.33% corresponds to visual words=50 and after this peak, a change in accuracy is observed. Above 50, there is an inverse relation between the number of visual vocabulary and the accuracy, that is, the accuracy drops with an increase in the number of visual vocabulary. This is understandable that after getting to a peak accuracy of 50 visual words, there is a

lack of variety of information in the image, which leads to a fall in the accuracy. In other words, the more the number of visual words, the closer the clusters are to each other, which imply that the number of wrongly detected mushrooms increase because of the similarity between visual words yielding lower accuracy.

The final feature vector for the mushroom image is a 50-dimension vector, which is as a result of counting the frequency of each visual word. If the number of visual words is 50 then, it means that the dimension of the final feature vector is 50 or a set of high dimensional image descriptors are converted to a list of visual numbers. The final feature vector or created bag of visual words histogram will be normalized within the range $[0, 1]$ and then fed into SVM classifier for further processing.

2.5 Conclusion

In this Chapter the image feature clustering challenges were discussed. The mushroom images were pre-processed, the high dimensional SIFT feature vectors extracted from the images and the advantage of using SIFT algorithm over other algorithms was discussed. Also the bag of words method was reviewed and the visual words or cluster centers for the image feature space were created. Finally the histogram of frequency of each visual word for the mushroom image was constructed and normalized. This final vector will be fed into classifier in Chapter three to perform image clustering.

Chapter 3

3 Application of Support Vector Machine to Detect Microbial Spoilage

3.1 Introduction

This Chapter explains the development of an SVM-based technique for classifying healthy and unhealthy mushrooms using feature vectors extracted from the previous Chapter. This approach performs classification using non-invasive data as only captured images were used for mushroom clustering. In this Chapter, the SVM-based approach for mushroom classification is described, SVM's parameter selection using 10-fold cross validation algorithm is stated and the classification results, experimental works and simulation in Matlab software are presented.

3.2 SVMs Based Classification Approach

Our proposed mushroom damage detection approach contains both training and detection procedure. As shown in Figure 3.1, the normalised histograms of visual words that were created in Chapter Two are used to train the classifier with SVMs. In the detection procedure, the image will be classified as either healthy (class one) or unhealthy (class two), and SVMs linear and nonlinear kernel functions will be employed to compare the results.



Figure 3.1: SVMs approach overview

Our dataset contains 240 images of both healthy and unhealthy mushrooms. Of the 240 images, 120 were used for training as a baseline for determining the status of the mushrooms while the remaining 120 were used for testing by using SVMs linear and nonlinear kernel functions.

Support vector machine as a supervised learning model is a pattern recognition method, which was first proposed, by Vapnik and Cortes in 1995 [15]. Each sample mushroom image in our training set contains a feature vector and a known label (either healthy or unhealthy).

The ultimate goal of SVM is to construct a model to predict the label of the mushroom testing data based on the information given by the training data. The main task of SVM is to detect and utilize complex patterns in data by ranking, classifying or clustering the data in high dimensional feature spaces. Furthermore, SVMs attempt to minimize an upper bound of the generalization error by maximizing the margin between the isolating hyperplane and the data instances.

The underlying theory behind SVM can be summarized as follows:

Having a training data set of label pairs $(x_i, y_i), i = 1, \dots, n$

If $x_i \in R^n$ and $y \in \{-1, 1\}^n$, where the notation x_i denotes the i^{th} vector in a data set $\{(x_i, y_i)\}_{i=1}^n$ and where y_i is the label associated with x_i . The object x_i is called examples.

The SVM can be computed by the following optimization theory:

$$\text{minimize}_{w,b} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (3.1)$$

$$\text{Subject to} \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0$$

Where vector W is known as the weight vector and b is called bias. The bias b translates the hyperplane away from the origin. The terms $\xi_i \geq 0$ are slack variables that allow an example to be in the margin ($0 \leq \xi_i \leq 1$, also called a margin error) or to be

misclassified ($\xi_i \geq 1$). The cost parameter C [16] allows some misclassification in order to create a more accurate model that separates the data into two categories. In this formula, the nonlinear function φ is employed to map the training feature vector x_i into high dimensional feature space in order to provide better classification accuracy. SVM includes two different classifiers; linear and nonlinear. For binary separable classification, the main concept of SVM is to construct a linear decision boundary or plane that divides the feature space into two groups. In nonlinear classification, the main goal of SVM is to map the data into a higher dimensional space by using kernel functions and then separate the data linearly. When the data are not linearly separable, the following kernel functions can be used:

- Polynomial Kernel Function: $K(X_i, X_j) = (\gamma X_i^T X_j + r)^d$, $\gamma > 0$
- RBF Kernel Function: $K(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|^2)$, $\gamma > 0$
- Sigmoid Kernel Function: $K(X_i, X_j) = \tanh(\gamma X_i^T X_j + r)$

The kernel function parameters are γ , r and d .

In this experiment, every mushroom image was pre-processed, feature vectors were extracted and then classified based on SVM. On this data set, different kernel functions were used to evaluate the detection performance of the SVMs. Kernel functions can improve detection performance at the cost of computational complexity. It is difficult to select the correct kernel for a given application domain. In order to get the best classification result, accurate SVM kernel hyper-parameters should first be selected.

3.2.1 SVMs Parameters Selection

In our experiment, the final feature vector of the mushroom image is extracted after pre-processing. The training and testing sets are recognized by support vector machine and SVMs linear and nonlinear kernel functions were employed and then compared. The 10-fold cross validation [41] method is employed on the training set to determine the kernel hyper-parameters with the best performance (highest classification accuracy). As shown in Figure 3.3, the data is divided into 10 parts in 10-fold cross validation.

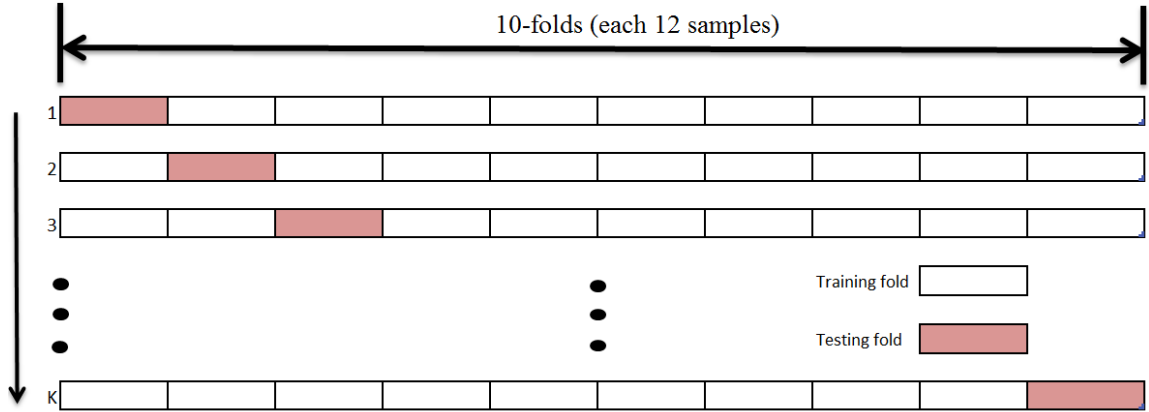


Figure 3.2: 10-Folds Cross Validation

The training data set of 120 images is divided into 10 equal subsets and the 10-fold cross validation algorithm is employed whereby 9 of the subsets are used for training the SVMs classifier while the remaining one set is used for validation, and cross validation accuracy is then obtained. The process is repeated 10 times so that every one subset is used as a test set for validation and the final cross validation accuracy is computed as the average of accuracies across all 10 iterations of the process. This average accuracy gives a general estimate of what the output error would be.

Simultaneously, a grid search parameter optimizer algorithm [16] was employed to find the best hyper-parameter C , γ and d in which SVM is trained with each points (C, γ) on the grid with both C and γ having a range of -10 and 10, which is the largest range of hyper-parameters being used. This large range is used iteratively to search the best hyper-parameters with the highest accuracies and the performance is estimated by using an internal cross validation. Changing the range of hyper-parameters first from large to medium and finally small gets the range of optimal SVM classifier performance, as illustrated in Figure 3.4 - 3.6 with the center point of the next smaller scale being the best point of the previous scale. The hyper-parameters that reached the maximum accuracy on the grid were chosen for training the SVM classifier. The complexity of the model is determined by a combination of both the cost parameter C and kernel parameter γ , and this explains why the optimal parameter is a range and not a single value.

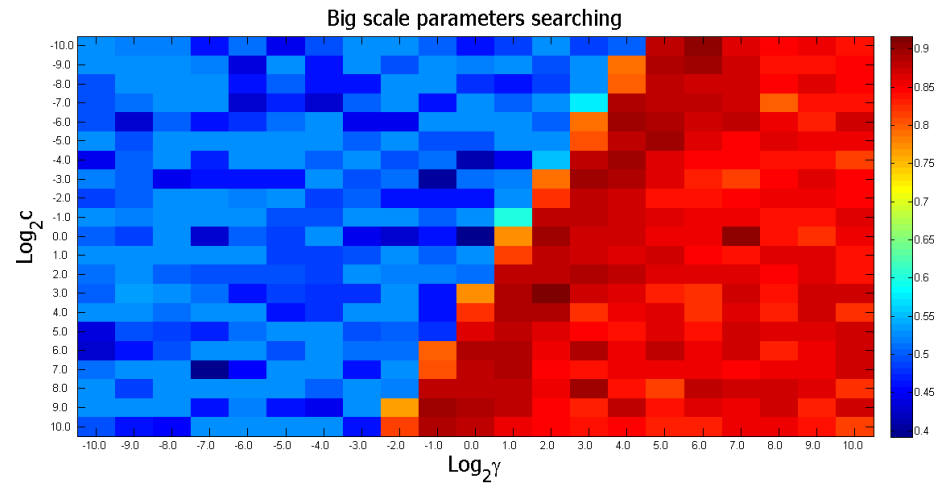


Figure 3.3: Parameters for polynomial Kernel (big scale)

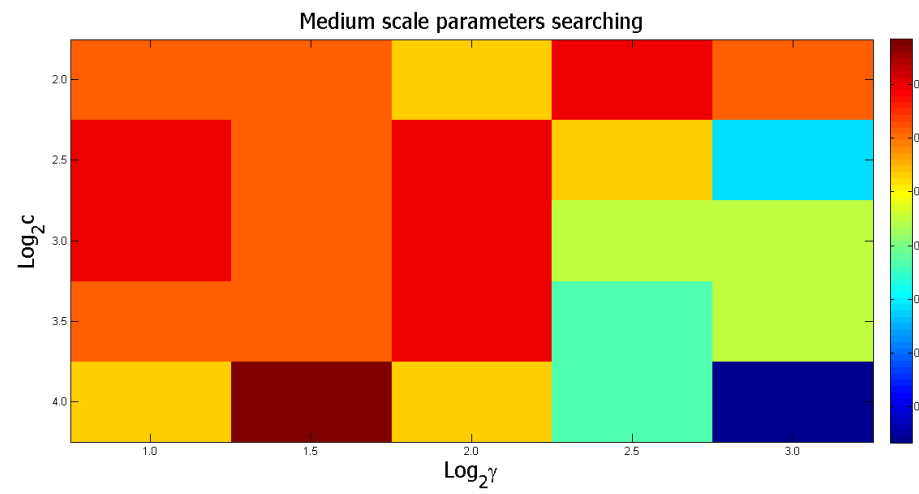


Figure 3.4: Parameters for polynomial Kernel (medium scale)

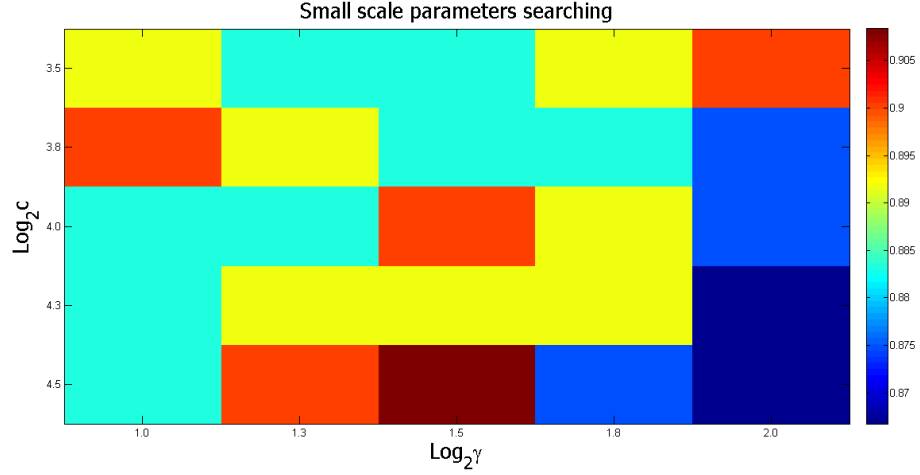


Figure 3.5: Parameters for polynomial Kernel (small scale)

For example as shown in Figure 3.4 - 3.6 the best hyper-Parameters for polynomial Kernel functions were $\log_2 C = 4.5$ and $\log_2 \gamma = 1.5$ which could provide highest cross validation accuracy on the training set. Table 3.1 below shows the grid range and the corresponding hyper-parameters for each kernel. Each hyper-parameters value is chosen from a discretization of an interval. For example -10, 1, 10 for $\text{Log}_2 \gamma$ means $\text{Log}_2 \gamma = -10, -9 \dots, 10$.

Table 3.1: Different kernels' hyperparameters

| Kernel | $\text{Log}_2 C$ | $\text{Log}_2 \gamma$ | <i>Best</i> $\text{Log}_2 C$ | <i>Best</i> $\text{Log}_2 \gamma$ | d |
|--------------------------|------------------|-----------------------|---------------------------------|--------------------------------------|-------|
| RBF kernel | -10,1,10 | -10,1,10 | 8 | -2 | --- |
| Polynomial kernel | -10,1,10 | -10,1,10 | 4.5 | 1.5 | 1,2,3 |
| Sigmoid kernel | -10,1,10 | -10,1,10 | 9.75 | -2.75 | --- |
| Linear | -10,1,10 | --- | 5 | --- | --- |

A set of hyper-parameters was first selected and validation was performed in order to find the kernel parameters with lowest output error, then the kernel functions were employed on the testing set to evaluate the final classifier.

As can be seen in Table 3.2, analysis of the results for the SVMs do not show any bias in the direction of either false positives (diseased mushrooms incorrectly identified as healthy) or false negatives (healthy mushrooms incorrectly identified as diseased).

Table 3.2: Confusion matrix

| Results Kernels | TP+TN | FN | FP |
|----------------------------|--------------|-----------|-----------|
| Polynomial | 112 | 3 | 5 |
| RBF | 108 | 4 | 8 |
| Sigmoid | 107 | 3 | 10 |
| Linear | 106 | 5 | 9 |

3.3 Experimental Results

The clustering results and running time of different kernel functions are shown in Figure 3.7 and 3.8 respectively where SVM was used. Although all four kinds have closely related running times, linear kernel is the fastest. The running time is a function of the number of visual words used: Increasing the number of visual words, leads to higher running time as shown in Figure 3.8, where W in visual words method is equivalent to the number of K in k-means clustering algorithm.

The accuracy of classified recognition should be high in all cases so that false identification is reduced. Since the objective is to identify diseased regions of the farm for quarantine purposes, rather than individual diseased mushrooms, classification accuracies for individual mushrooms of 90% are satisfactory. Unlike the linear kernel with the shortest running time and lowest recognition rate, the Polynomial SVM function

has the best recognition accuracy but the longest running time of all other kernel functions.

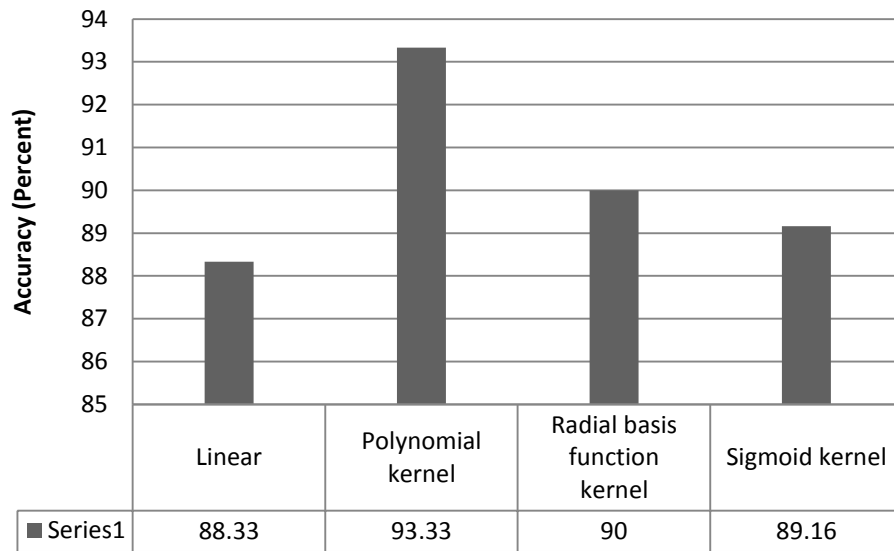


Figure 3.6: Classification Result

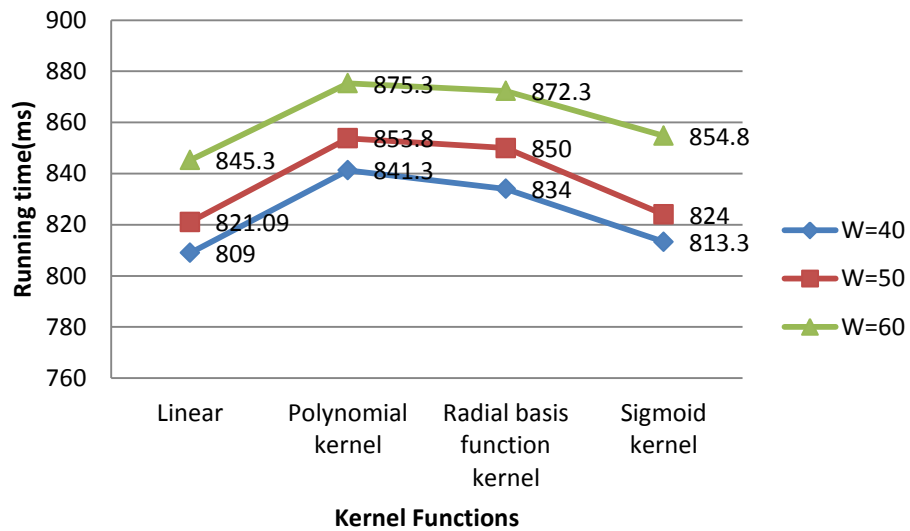


Figure 3.7: Classification running time

The classification results on the testing set for SVMs linear kernel and the three other kernel functions; Polynomial, RBF and Sigmoid are plotted in Figure 3.9 – 3.11, respectively, distinguished by different colors per class. In these plots, the vector dimension is reduced using Multi-Dimensional Scaling method (MDS) [42] to 2D space to enable visualization of the data set. It should be mentioned that MDS was employed to demonstrate the dissimilarity data between two classes by using Euclidean distance model. In fact the goal of MDS algorithm is to place each sample in 2 dimensional space in a way that the distance between samples are preserved as well as possible. We used MDS to project high dimensional feature space into the plot. As a result these plots are for illustration purposes.

The samples are classified as either belonging to class 1 of unhealthy mushrooms indicated by the orange symbols or class 2 of healthy mushrooms indicated by the red symbols. The unfilled circles are the training set and filled color indicates the class label for testing set assigned by SVMs. Misclassified samples (FP+FN) are represented by circles that have a border color. For example, for the linear kernel, there are 14 circles with border color, which is equivalent to the number of false detections according to our classifier results shown in Table 3.2.

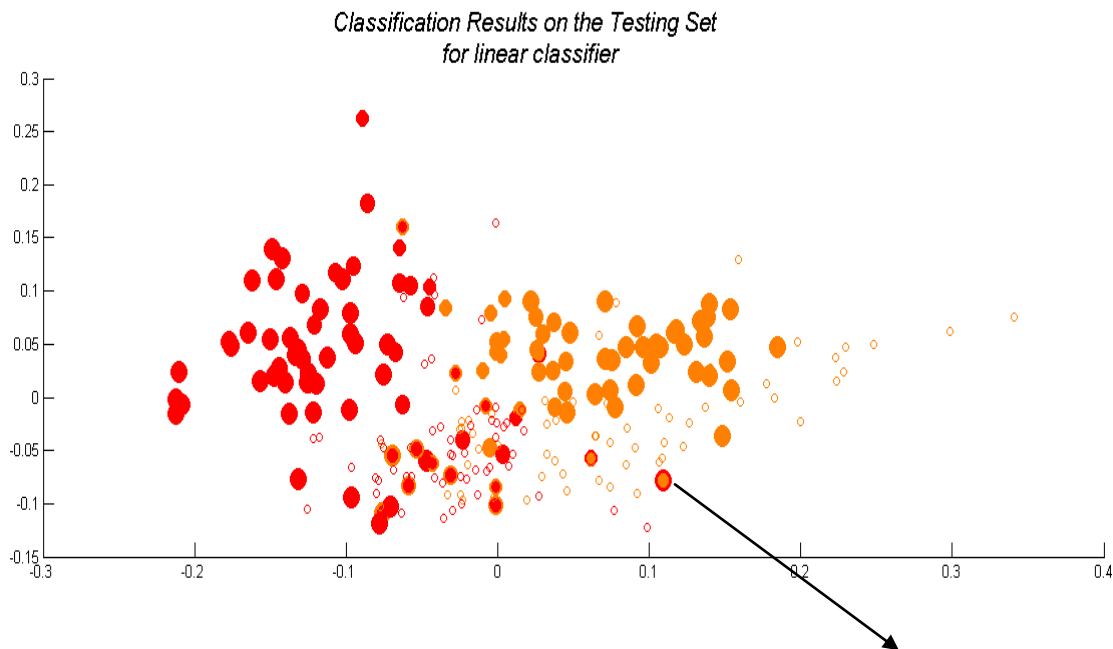


Figure 3.8: Linear classifier result

False detection (border color)

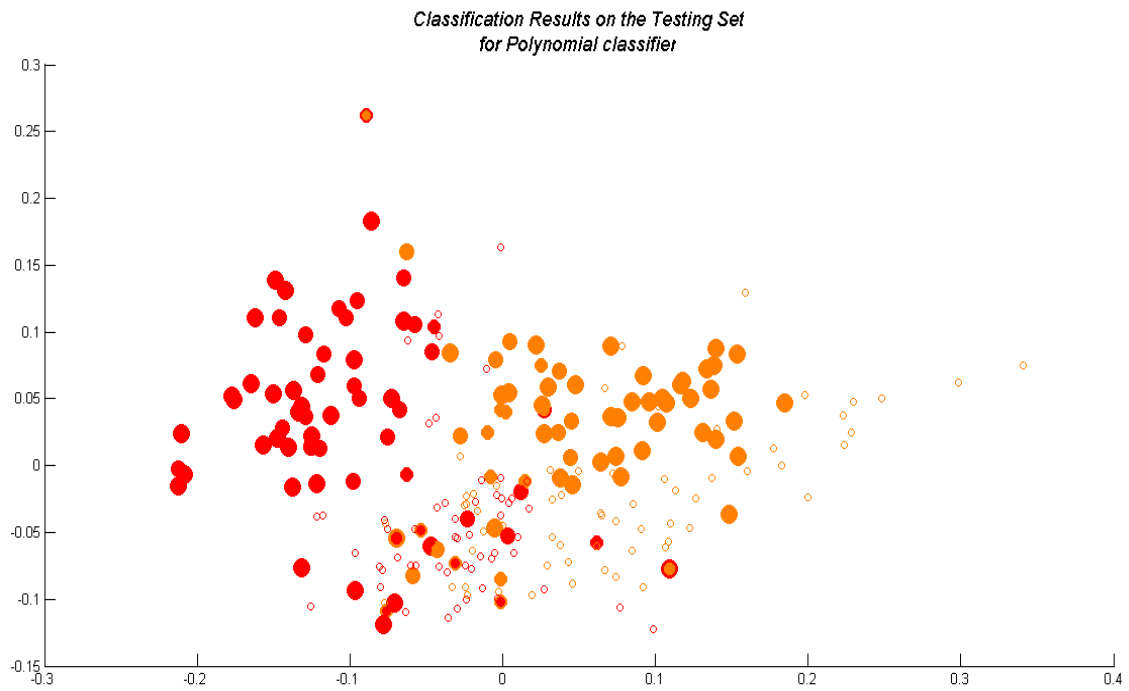


Figure 3.9: Polynomial classifier result

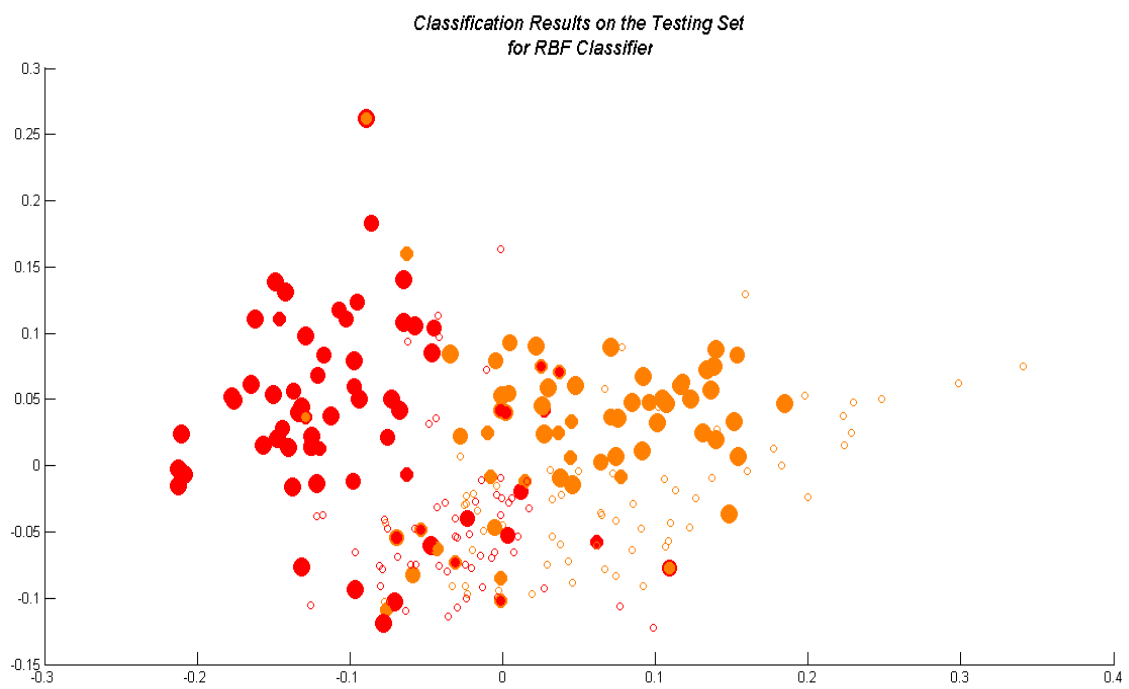


Figure 3.10: RBF classifier result

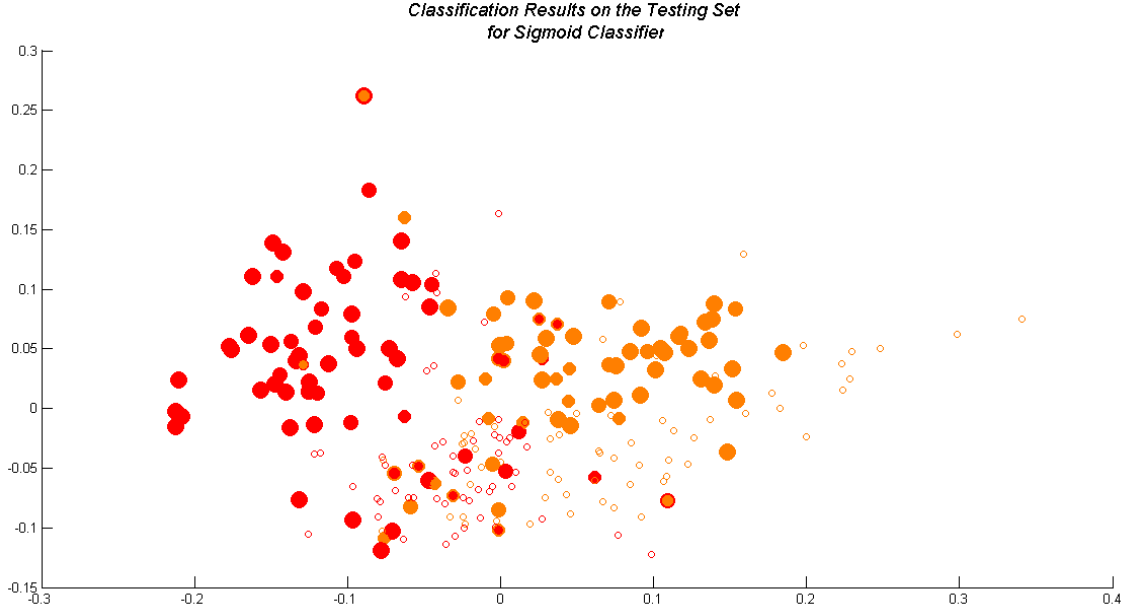


Figure 3.11: Sigmoid classifier result

Figure 3.13 – 3.15 show the receiver operating characteristics (ROC) curve of the three kernel functions for testing data set. In this paper, ROC curve is employed in such a way that horizontal axes belong to false positive rate (false alarm) and vertical axes show the true positive rate (recall). The true positive rate (TPR) and false negative rate (FPR) can be calculated as follows [43]:

$$TPR = \frac{TP}{P} = \frac{TP}{(TP+FN)} \quad (3.2)$$

$$FPR = \frac{FP}{N} = \frac{FP}{(FP+TN)} \quad (3.3)$$

The area under the curve (AUC) indicates the performance of our classifier which is high for all the kernel functions and it is also higher than the random classifier with a performance of 0.5, indicated by the green diagonal line. The ideal classifier is a one which generates an ROC curve with AUC=1. Another criterion that can be used to

evaluate the overall quality of ROC is EER (equal error rate) [43]. Equal error rate can be described as the value which satisfies the function $FPR=FNR$ so lower EER scores are better and the minimum is zero.

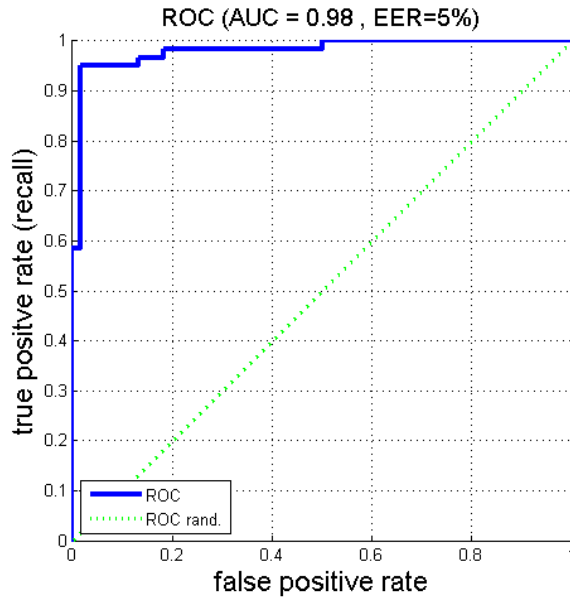


Figure 3.12: ROC curve for polynomial kernel

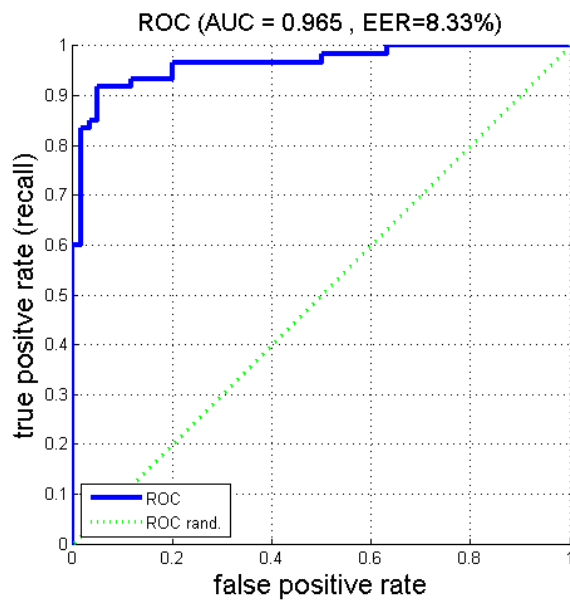


Figure 3.13: ROC curve for RBF kernel

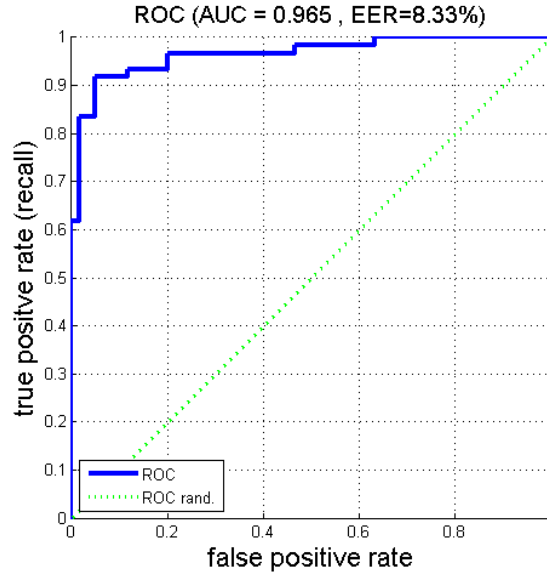


Figure 3.14: ROC curve for sigmoid kernel

3.4 Conclusion

In this Chapter, a new approach to the mushroom damage detection problem for a harvesting robot vision system using SVMs algorithm was proposed. The extracted feature vectors from Chapter two were fed into a classifier and different kernel functions were employed to compare the classification results.

The mushrooms were classified into two categories: class one and class two, and the clustering was useful for finding the contaminated area of the farm. To validate the classifier results, ROC curve was employed as well as MDS technique to visualize both the training and testing data set in 2D space. Our experimental results on both the training and testing data set show that our proposed approach performs well using the combination of SVMs kernel functions with SIFT and BOW methods. In our application, the robot is supposed to find the contaminated area in the mushroom growing bed as such an accuracy of more than 90% is desirable. The VLFeat open source library in Matlab [44] was used to compute the SIFT features and visual vocabularies.

Chapter 4

4 Segmentation Algorithm for Mushroom Identification

4.1 Introduction

In this Chapter, a new method to identify and locate mushrooms in the growing beds is proposed. In order to design a vision system capable of identifying and tracking mushrooms in the growing beds, the following challenges need to be considered:

- The overlapping of mushrooms caused by the irregular growth pattern, which makes the contour detection more difficult.
- Mushrooms do not get the same amount of light so the segmentation algorithm should be insensitive to illumination changes.
- The image background contains a lot of irregularities and makes it difficult to separate the mushrooms from the background.
- Mushrooms are located at different angles in the growing bed which makes the segmentation task challenging and as such, the robot is supposed to detect the inclination and direction of each individual mushroom for pickup.

In order to separate mushroom caps from the background, color information of the image as well as the K-means clustering algorithm were used. Hough transform method was used to locate mushrooms in the growing beds such that the gradient information of the image was used to locate the center position of the mushroom in the XY plan. To get the depth (Z-coordinate) of each pixel in the image, Microsoft Kinect was employed. The performance of this algorithm was confirmed by using several images taken from mushroom growing beds. The identification algorithm was presented in this Chapter and

to ascertain its accuracy, several mushroom images were tested by MATLAB image processing toolbox and the segmentation results are presented and discussed.

4.2 Segmentation Based on HSV Color Space and K-Means

For this experiment, images were taken under different lighting conditions and locations in the mushroom growing bed using an 8-mega pixel digital flash camera in 24-bit RGB format. The camera was placed in different orientations; horizontally and vertically and the camera's flash was used for illumination, which made it easier to capture the brown color of the mushroom. In Figure 4.1 and 4.2 the RGB image channels are demonstrated as an example.

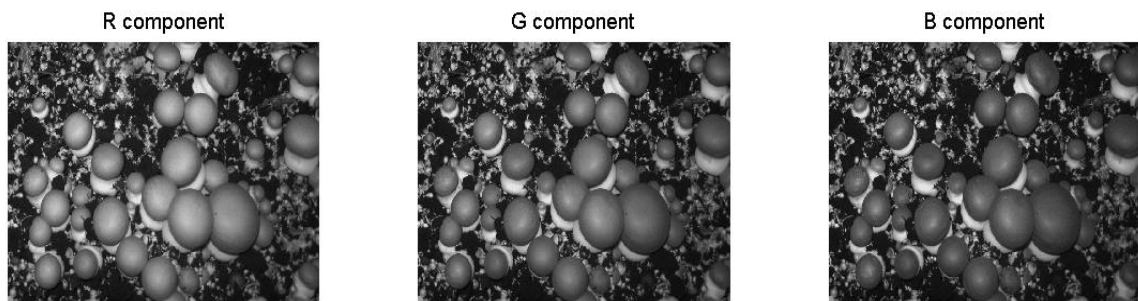


Figure 4.1: RGB channels of sample one

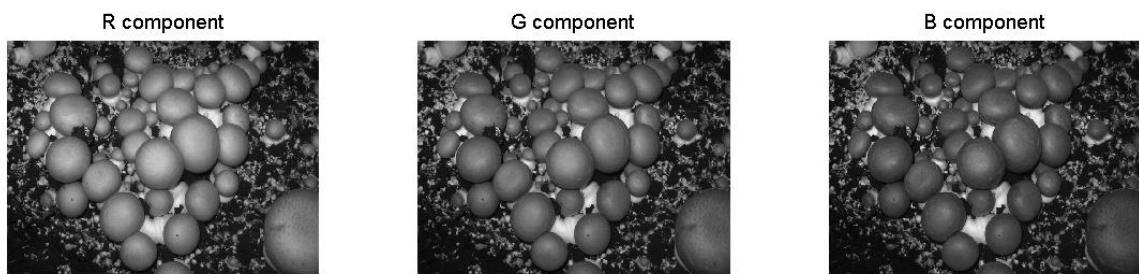


Figure 4.2: RGB channels of sample two

The image is converted from the RGB color space into the HSV [45] color space because the RGB color space does not contain enough information to represent the image features. The HSV color space makes decomposition of the image into meaningful parts for further analysis easier.

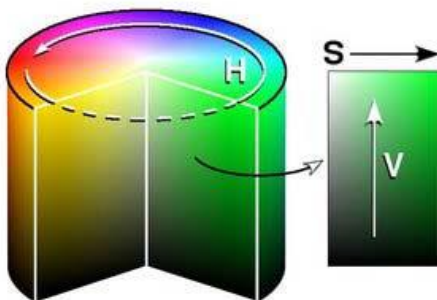


Figure 4.3: HSV color space in [45]

As shown in Figure 4.3, Hue (H) contains 360 degree of rotation and is normalized within the range of [0 255] where red, which is the beginning is at an angle of 0. The depth of the color can be measured with Saturation (S) whose value ranges from 0 to 255 while the brightness of the color denoted by Value (V) ranges between 0 to 255 where 0 corresponds to darkness and 255 represents full brightness.

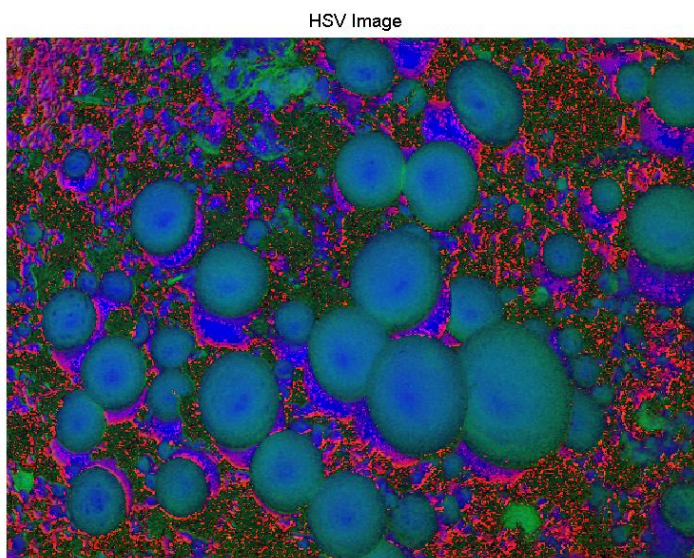


Figure 4.4: HSV sample image of mushroom

As shown in Figure 4.4 - 4.6, HSV color space explains colors in a more meaningful way in terms of Hue, Saturation and Value than RGB color space.

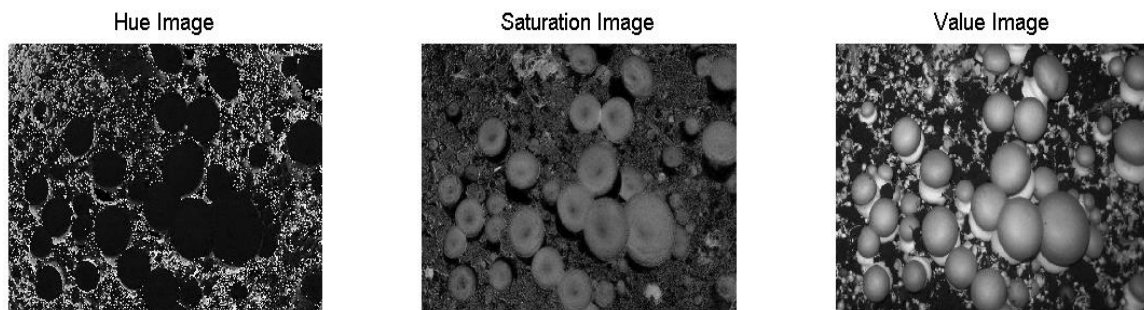


Figure 4.5: HSV channels of sample one

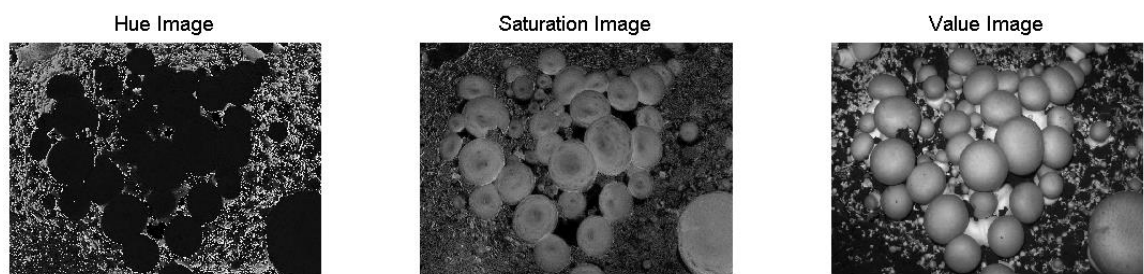


Figure 4.6: HSV channels of sample two

Since the color information of the mushroom's caps plays a vital role in this project, HSV color space is readily used because it is easier to extract and separate the appropriate features from the background of the image, thus providing the necessary information that are needed to measure the mushroom's dimension and locations in XY coordinate space of the growing bed. Ignoring the variation in brightness, the number of colors in the image can be visually estimated.

HSV color space was employed for segmentation because it gives the most useful color information for mushroom cap separation, in terms of separating the brown colored mushroom cap from the background, which is basically composed of black soil, white spots and mushroom stipe. By using the HSV color space, the illumination variation problem can also be solved such that non-uniform illumination does not affect the segmentation process.

The H and S values of each pixel are taken and classified based on K-means [46] clustering algorithm because color information of the image exists in the H*S color space and these pixel features are clustered and categorized by similarity in colors.

K-Means clustering algorithm is one of the learning algorithms that can be used to categorize a given data set to a specific number of clusters. The main concept of K-Means is to identify the center point of each cluster of data set that is the point nearest to the centroid of that cluster rather than the next cluster.

Given a d dimensional vector: $D = (X_1, X_2, \dots, X_n)$ $X_i \in \mathcal{R}^d$

Assume K clusters with unknown centers: $\mu = (\mu_1, \mu_2, \dots, \mu_K)$ $k \leq n$

The task is to identify K groups of data points and assign each point to one of those groups so as to minimize the sum of squares distance to the centers in equation 4.1.

$$\sum_{i=1}^k \sum_{x_j \in \mu_i} \|x_j - \mu_i\|^2 \quad (4.1)$$

The algorithm can be summarized in the following steps:

- Initializing group centroid $\mu_1, \mu_2, \dots, \mu_k$ arbitrarily
- Assigning each sample of d dimensional vector D to the nearest centroid μ
- Recalculating the center points μ again once all the samples are assigned to one group
- Iteratively repeat these steps until convergence (center points do not move)

Although in K-means clustering algorithm, the number of clusters is an input parameter and is user defined, the number of clusters is obvious because from the analysis in the previous section if we ignore the variation in brightness, the H and S pixel values can separate the image into 3 groups. K-means clustering algorithm is started with $K=2$ and

increased till the best clustering result is reached. In our experiment $k=3$ was chosen since the image consists of 3 major colors; white as mushroom stipe, black as nutritional substrate soil and brown as mushroom cap. As shown in Figure 4.7 and 4.8 by using K-means clustering method, each object in the image is assigned a cluster index.



Figure 4.7: Cluster index for image one



Figure 4.8: Cluster index for image two

We take the cluster index assigned to the mushroom cap automatically and separate the light brown from dark brown by using a threshold as shown in Figure 4.9. The threshold value is a user-defined value which was selected automatically using Otsu's method [47].

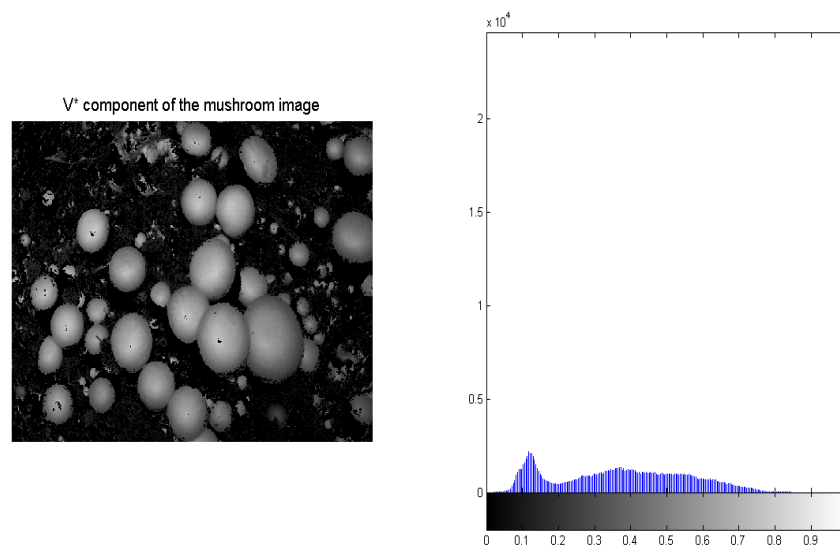


Figure 4.9: Intensity histogram of the image

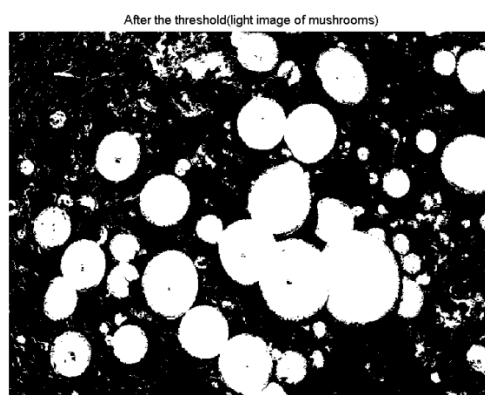


Figure 4.10: Image thresholding

As it can be seen in Figure 4.10, the result is an image mostly containing the area of interest (light brown regions).

Mathematical morphology was used to reconstruct the image obtained from thresholding process because the image was distorted as a result of the process leading to a noisy image and a change in the shape of the mushroom cap.

A disk shaped structuring element of size 10 was used to carry out the opening process, which involved the application of erosion followed by a dilation method to remove the small objects of the image, and subsequently a disk shaped structuring element of size 2

was employed to carry out the closing process involving the application of dilation, followed by erosion to fill small holes.

As it can be seen in Figure 4.11, the small regions are eliminated through the opening process. The user based on the required size of mushrooms needed to be picked determines the size of the structuring elements, and regions with values beneath the specific value were eliminated.



Figure 4.11: Mathematical morphology

This user- defined value helps to identify only mushrooms that are in an optimal range size. The resulting color image after morphological operations is shown in Figure 4.12.

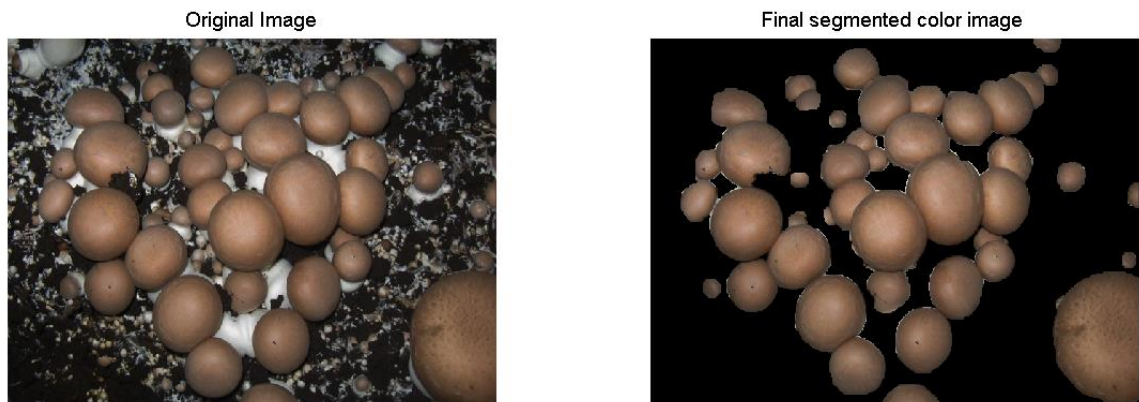


Figure 4.12: The resulted image (sample one) after morphological operations

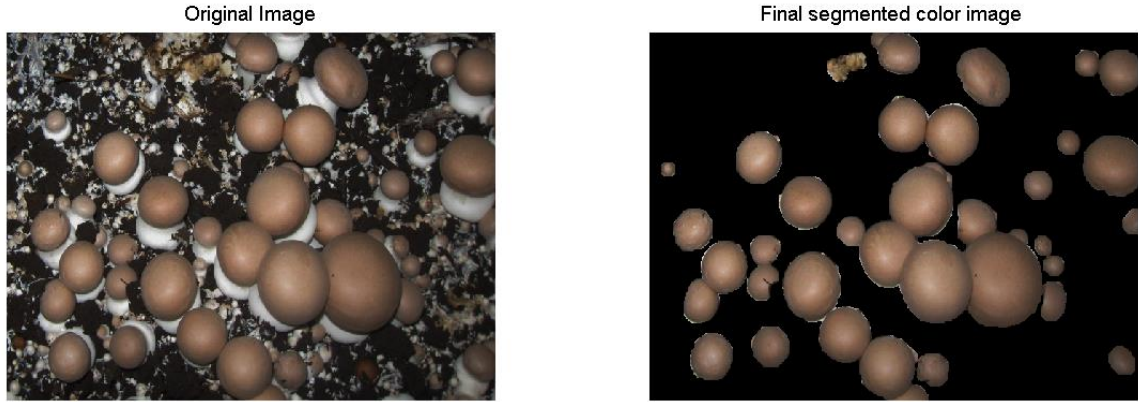


Figure 4.13: The resulted image (sample two) after morphological operations

The image boundary (contour) was extracted in order to detect the size and position of each individual mushroom in the image. The image contour's information could be used to estimate the number of pixels that exist in each mushroom cap and consequently the size of the mushrooms caps. In order to extract image contours, many edge detection methods were employed and the results compared. The magnitude of the gradient of the image was used as one of the first edge detection methods.

Gradient [1] of an image points in the direction of the most rapid change in intensity. The gradient vector can be created by combining the partial derivative of the image in the directions x and y . We can write this as follows:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \quad (4.2)$$

Also the image gradient direction and gradient magnitude can be computed as follows:

$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right) \quad (4.3)$$

$$|\nabla| = \sqrt{\nabla f_x^2 + \nabla f_y^2} \quad (4.4)$$

The result of extracting the gradient of an image for two samples is demonstrated in Figure 4.14 and 4.15 where the mushroom edge contour lines have been extracted.

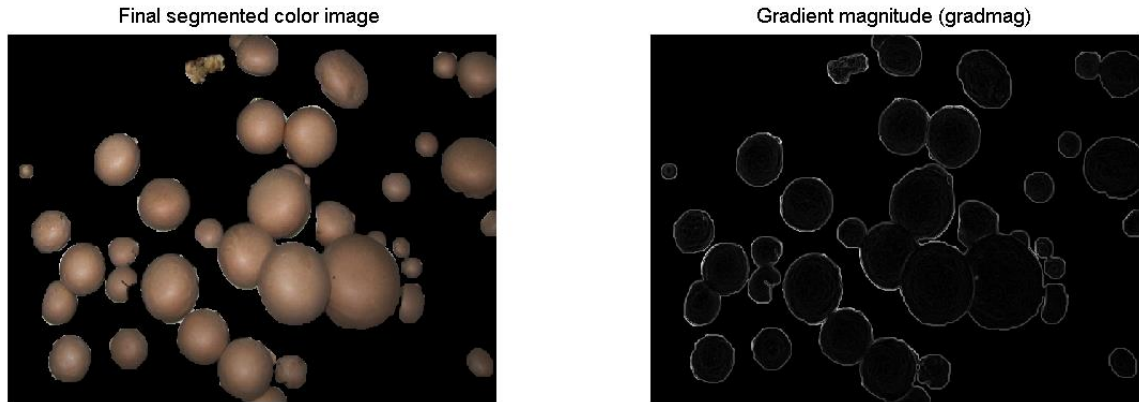


Figure 4.14: Contour extraction for sample one

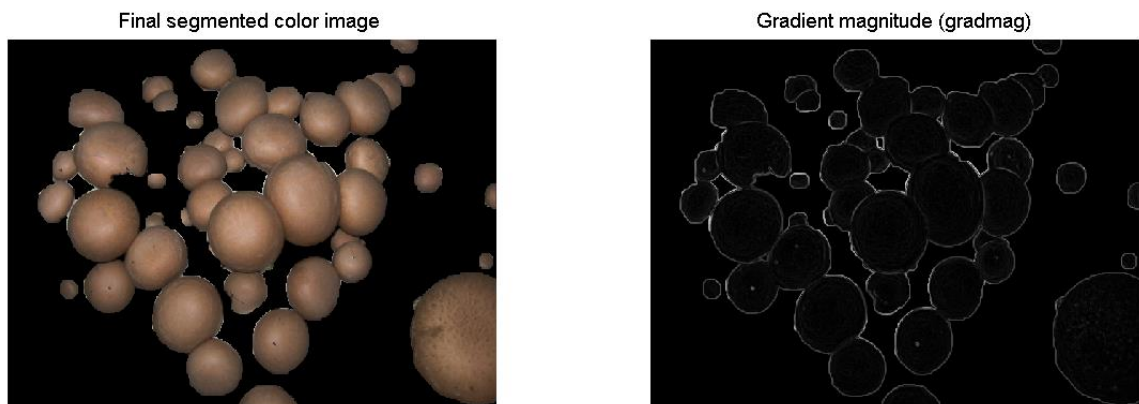


Figure 4.15: Contour extraction for sample two

The major problem with the edges of the images above is that the edges and borders of the mushrooms cannot easily be distinguished when they overlap, which made it hard for the edge detection algorithm to detect the mushroom's border.

Other edge detection algorithms [1] such as Sobel, Canny and Log were also applied on the mushroom images, however, none of those methods could fully extract the contour points for mushrooms that are partially visible or the ones that are overlapped.

Thresholding was also one of the methods that used to separate the connected mushrooms thereby extracting their edges. In order to do so, the final segmented color image of mushroom was converted to grayscale image and a threshold value was chosen based on

the histogram of the grayscale image. The grayscale image was then converted to a binary image as shown in Figure 4.16.

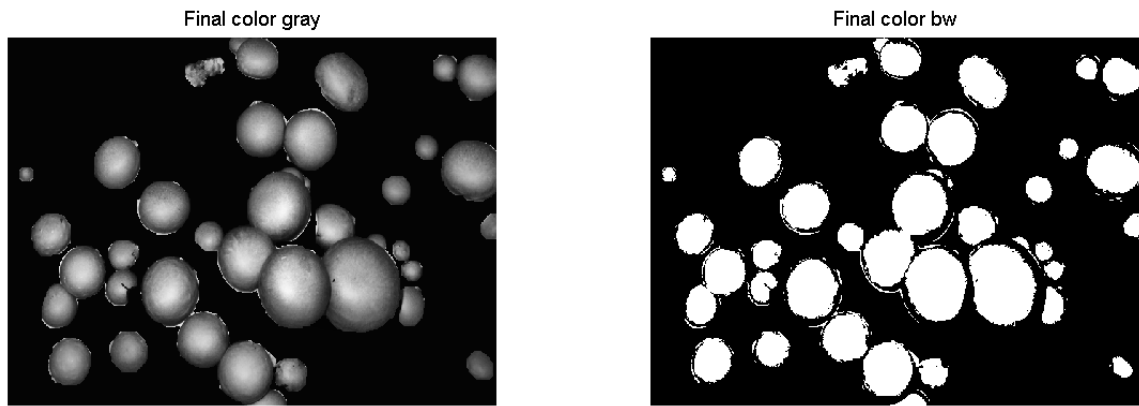


Figure 4.16: Binary image converted from grayscale image

As can be seen in the image above, thresholding cannot be used as a reliable method to extract the mushroom edges because of illumination changes and shadow. Mushrooms in the binary image format are smaller than the original because of the selected inaccurate threshold value. To overcome these difficulties in extracting mushrooms edges, a new method is proposed to accurately identify mushrooms and measure their dimensions even in case when they are clustered together, and this method will be further explained in the next section.

4.3 Mushroom Size Detection

As it was previously explained, two of the major challenges in vision system of the mushroom harvester are separation of overlapping mushrooms and identification of different-sized mushrooms. Different methods like Watershed algorithm [1], threshold method, contour base method and feature based segmentation methods have been proposed for the harvesting of mushrooms but these methods have failed at detecting individual mushrooms clustered together.

In order to find the XY position of a mushroom in image, the pixels that belong to that mushroom have to be identified and the center position (centroid) of those pixels

calculated. In cases of clustered mushrooms however, there is no clear way of separating pixels that belong to one mushroom from pixels that belong to another.

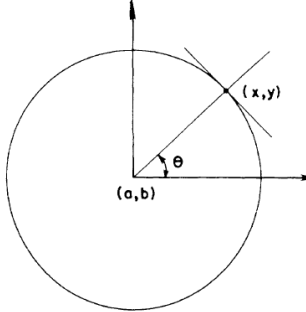
An algorithm based on circular Hough transform was used to locate the XY position of each individual mushroom in the 2D image because of the circular shape of the mushroom cap, and the circular Hough transform [48]-[50] was used because it is the most common algorithm to detect circular shapes. In the original implementation of circular Hough transform method, features in the image space are mapped to a three dimensional parameter space.

A circle is defined by the following equation:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (4.5)$$

Where r is the radius and (a, b) are the center of the circle.

$$\tan \theta = \frac{y - b}{x - a}$$

$$b = \tan \theta a + (y - x \tan \theta)$$


The main concept of HT is to transfer each (x, y) in this formula into parameter space. All the resulting points in parameter space lie on the surface of an inverted right- angled cone and the intersection points of these cones are on the circle's parameters. The three dimensional Hough transform method requires a large accumulator array which makes implementing this method inefficient in terms of both computer storage and computation time. For a more efficient HT implementation, the gradient field information of the image should be used [48].

As shown in the Figure 4.17 this method can be explained in the following steps:

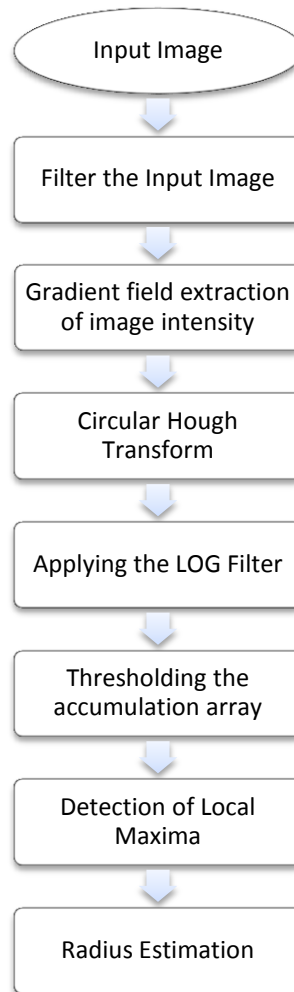


Figure 4.17: Mushroom size detection overview

The first step is to apply a filter on the image to smooth out the small scale irregularities along the edges because jagged edges result in false detection as this algorithm is based on the gradient field of the input image. It is therefore necessary to make the boundaries of the mushrooms smooth so that caps with less rounding can be easily detected. A 5-by-5 Gaussian filter is used to remove the high frequency components from the input image.

Then the gradient field of the mushroom image was calculated using the following formula:

$$\nabla I(i, j) = (Vx, Vy) |_{(i, j)} = (I(i, j) - I(i, j - 1), I(i, j) - I(i - 1, j)) \quad (4.6)$$

Where $\nabla I(i, j)$ is the gradient vector which includes x and y components and $I(i, j)$ is intensity at (i, j) .

The vectors that are perpendicular to the mushroom's edge points must all intersect at the mushroom's center coordinate (centroid). In other words, due to the circular shape of the mushroom, all the nonzero gradient vectors in the gradient field cross the centroid of each individual mushroom, which can be used to create an accumulation array by mapping the gradient field of the mushroom image. This mapping leads to the creation of spots with higher intensities in locations the centroid of mushrooms exist.

An accumulation array with the same dimension as the gradient field was constructed using the following steps [49]:

- For each bin in the accumulation array which lies on the line segment described by $\nabla I(i, j)$, a strength value was added to that bin.
- The strength value was defined by the gradient magnitude.
- The length of the line segment was set to be the maximum diameter of the mushroom. This value was determined by the user.

Figure 4.18 shows an example of the construction of the accumulation array. The maximum intensity in the accumulation array shows the center coordinate of the circle and the next step is to locate the XY positions of the peaks in the accumulation array.

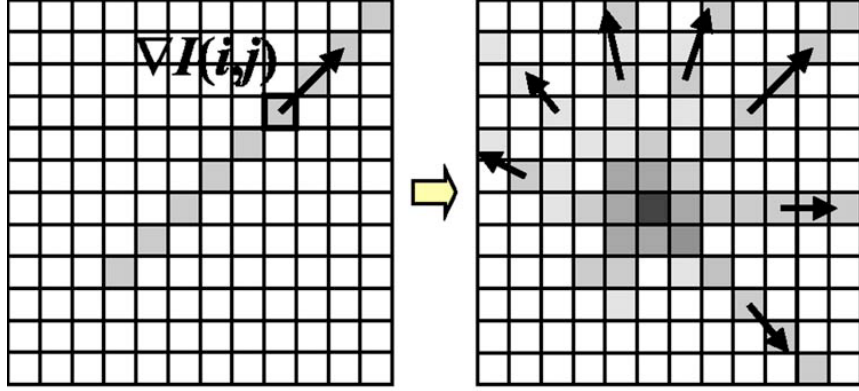


Figure 4.18: Construction of the accumulation array in [49]

The detection procedure of the local peaks or in general center position of mushrooms was implemented using the following steps [49]:

- LoG filter implementation

One of the problems in locating peaks in the accumulation array is that there are some peaks that are very close to each other making separation a difficult task. The area between these local peaks needs to be filled with zero in order to separate and detect these local peaks individually. The most common filter used to perform this task is the Laplacian of Gaussian filter [1]. LoG filter's response in the vicinity of a change in intensity was used to separate the local peaks in an accumulation array as LoG computes the second spatial derivative of an image. The Laplacian of an image can be calculated as follow:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (4.7)$$

Where $I(x, y)$ is the pixel value and $L(x, y)$ is the Laplacian of the image. The LoG filter was used to fill the transition area between two peaks in accumulation array.

- Accumulation array thresholding

To remove the uniform intensity such as image background, the accumulation array was thresholded. 25% percent of the maximum intensity in the accumulation array was used as the threshold value. To separate peaks in the accumulation array, the local peak's bins were filled with ones and the other areas in the accumulation array were marked with zeros.

- Detection of isolated regions

A region growing algorithm [1] was used to detect and isolate different regions in the accumulation array. Region growing is a common method in image processing which uses the similarities in image pixels to isolate regions in an image. The isolated regions in the mushroom's image were detected and the position of the local pick is set to be the intensity weighted center of a region.

The position of the detected local peaks in the accumulation array was converted into the actual positions in the image. The mushroom cap's area was calculated in pixel with the aid of the center position and radius of each individual mushroom.

Several images were taken from the mushroom growing beds and the maximum radius of a mushroom was measured manually. This measured value was used in the Hough transform algorithm as the possible maximum radius of the circles. The result for the three sample images are demonstrated in Figure 4.19, 4.22 and 4.25. As it can be seen in those figures the algorithm works well even for the mushrooms that are overlapped or partially occluded. There are only two false detections in figure 4.19 and they are caused by the color similarity between an object in the background and the mushroom's cap.

The other issue that severely affects the performance of this algorithm was caused by camera flash as shown especially in Figure 4.25 in which some mushrooms could not be detected or mistakenly detected.

In Figure 4.19 the image contains 31 mushrooms being segmented correctly. The processing time was 15.31 seconds for the whole mushrooms being identified so the approximate recognition speed is 0.49s for each mushroom with an Intel Core-3 2.4-GHz CPU. The accumulation array for circular Hough transform is also demonstrated in Figure 4.20 in which local points with more intensity are the center position of each individual mushroom.

Moreover in Figure 4.22 the image contains 31 mushrooms being segmented correctly. The processing time was 14.68 seconds for the whole mushrooms being identified so the approximate recognition speed is 0.47s for each mushroom with an Intel Core-3 2.4-GHz CPU. Also the 3D view of the accumulation array for this image is shown in Figure 4.24. As can be seen in this figure even in cases when mushroom shape is less perfect the detection algorithm still gives a useful result.

It should be mentioned that false positives include the mushrooms that were wrongly detected by the identification algorithm due to similarity in color between the substrate and the mushroom's caps. Based on the fact that these false detections will not like mushrooms in IR images, they can be prevented using a sensor fusion algorithm.

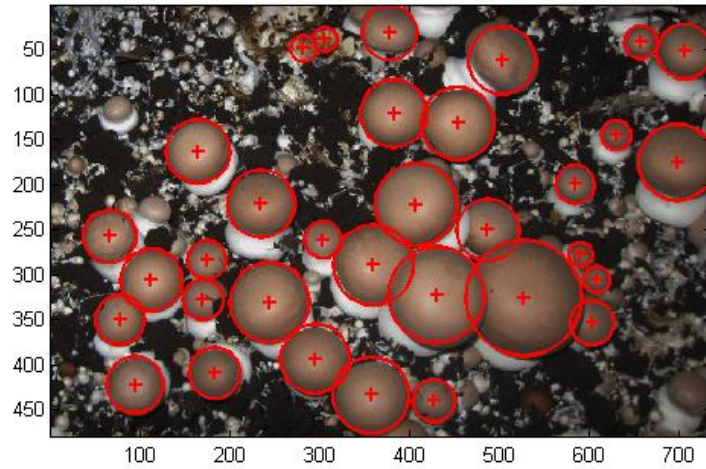


Figure 4.19: Segmentation result for first image

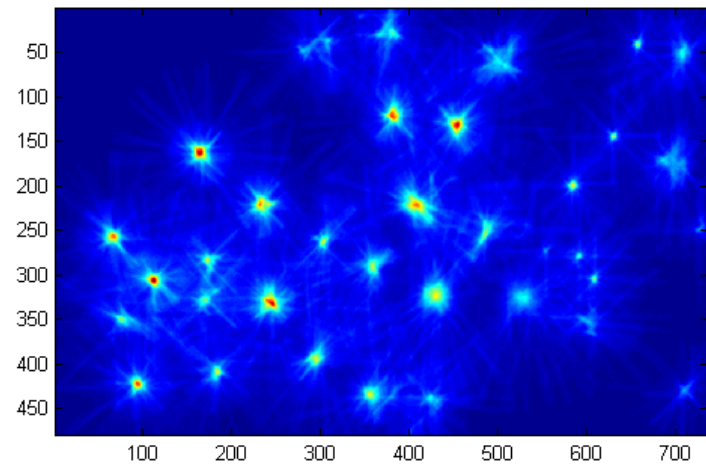


Figure 4.20: Accumulation array for first image

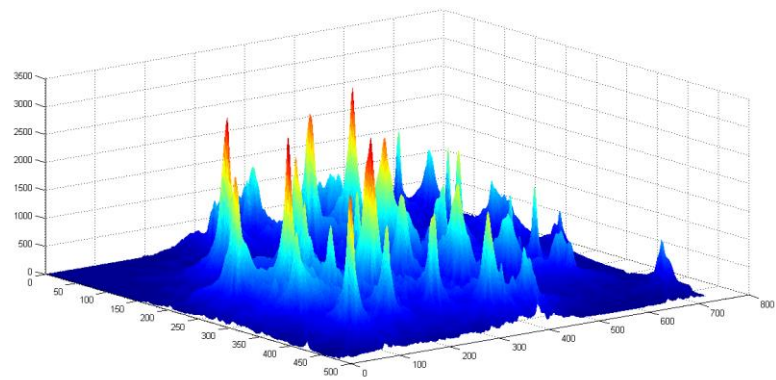


Figure 4.21: 3D view of the accumulation array for first image

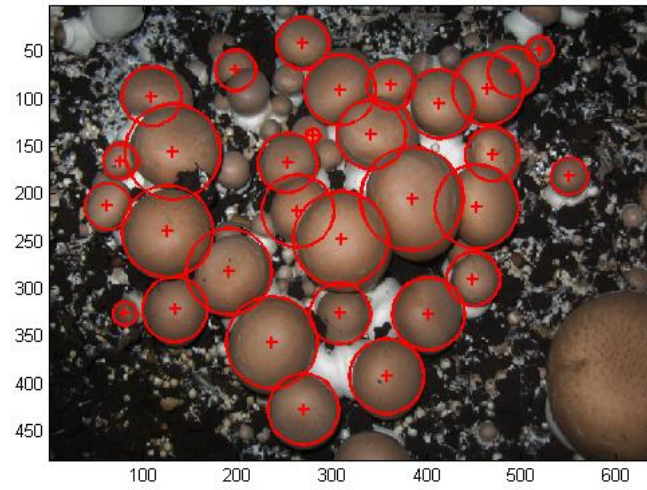


Figure 4.22: Segmentation result for second image

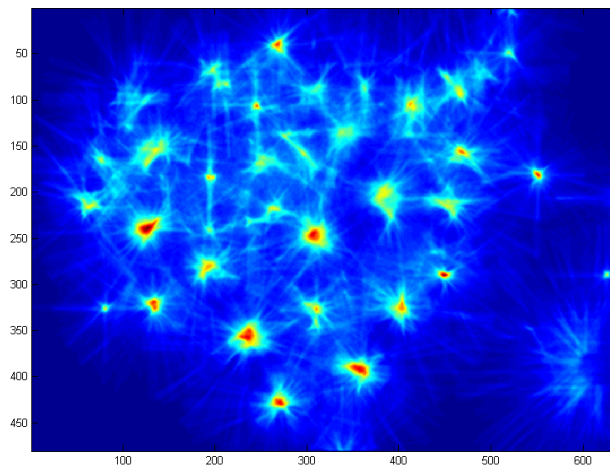


Figure 4.23: Accumulation array for second image

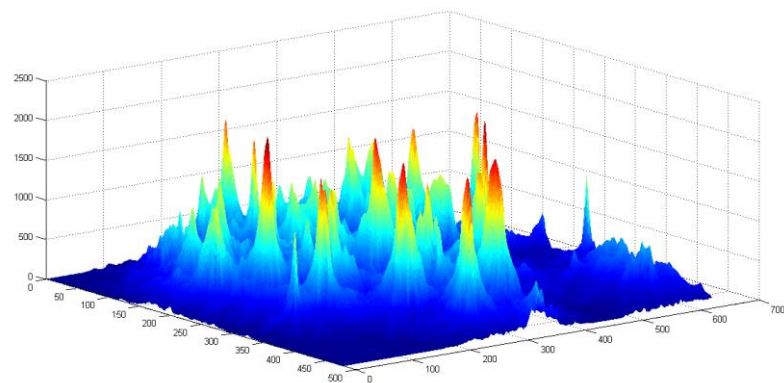


Figure 4.24: 3D view of the accumulation array for second image

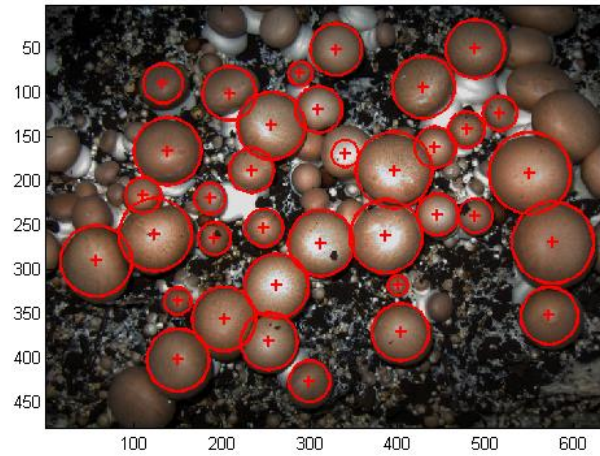


Figure 4.25: Segmentation result for third image

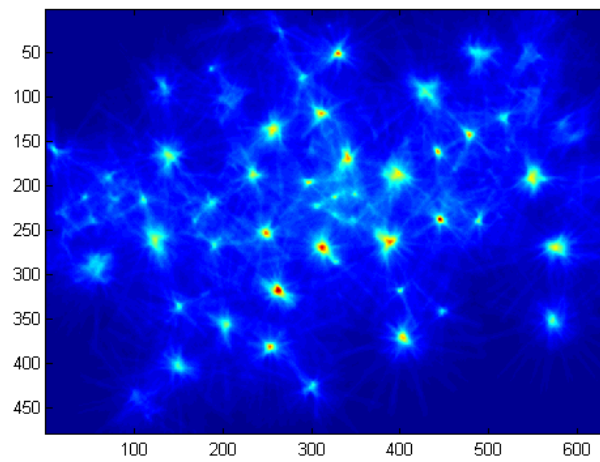


Figure 4.26: Accumulation array for third image

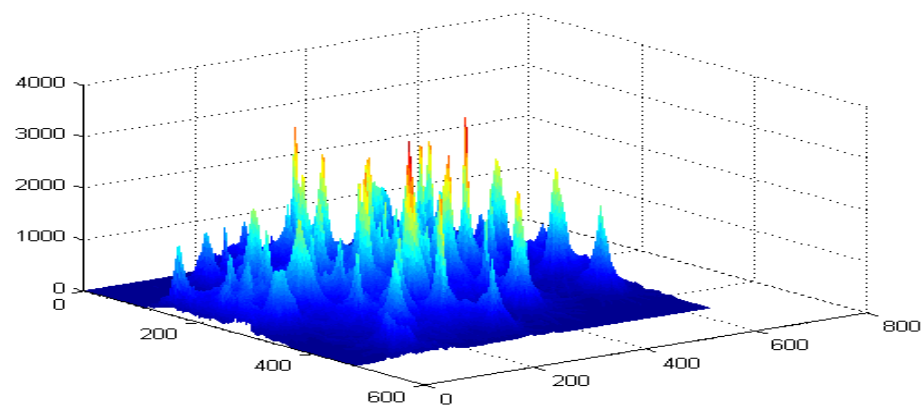


Figure 4.27: 3D view of the accumulation array for third image

4.4 Conclusion

In this chapter, a new approach for identifying brown mushrooms and measuring their dimensions was proposed. A set of experiments was carried out to find the best method of recognizing the right mushroom to pick up. A new algorithm was developed and the segmentation results, experimental works and simulation in Matlab software were demonstrated.

Chapter 5

5 Application of Infrared and Stereo Cameras in Mushroom Harvesting

5.1 Introduction

In this Chapter, infrared and stereo vision systems were employed to study the feasibility of using both depth and temperature information of the mushroom images to enhance the identification algorithm. Several images are taken using both stereo and infrared cameras and the results are demonstrated.

5.2 Image Depth Extraction

In the previous section, the centroid and the mushroom cap's area for each mushroom were calculated in pixels. A stereo camera was used to investigate the possibility of using stereo vision system to calculate the XYZ position of the mushroom's centroid and mushroom's cap area in millimeter. The Microsoft Kinect was used to extract the depth map of the mushroom image.

The Microsoft Kinect [51] module provides an interface to the Microsoft Kinect XBOX sensor. The sensor is being used mostly for robotics. It can be used to provide a 640×480 depth map in real time which demonstrates the distance between the objects and the sensor. This information is useful for navigation and target distance estimation to the sensor.

Figure 5.1 shows the depth map which was extracted for the image shown in Figure 5.2. The depth map information of the image or the distance of the object in the image to the robot's camera can be used as a guide for the robot arm in order to find the position of

each mushroom in the growing bed and also distinguish the mushrooms from their neighbors. This depth data can be used in order to find the orientation of the mushrooms to the substrate and therefore define a 3D model for the mushroom's cap.



Figure 5.1: Original Image for depth extraction

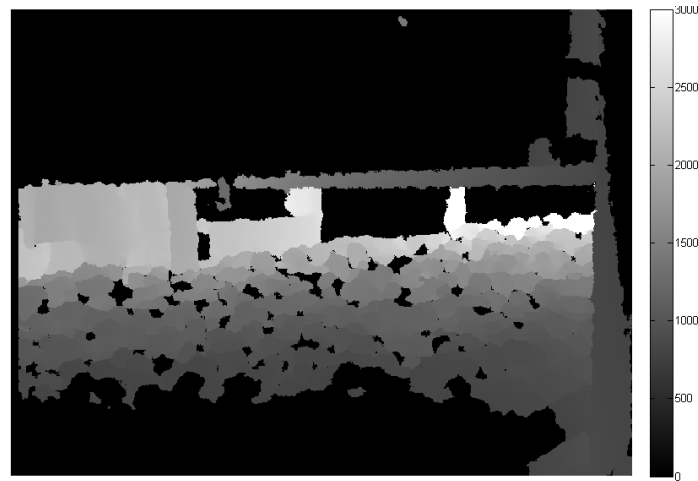


Figure 5.2: Depth of the mushroom image

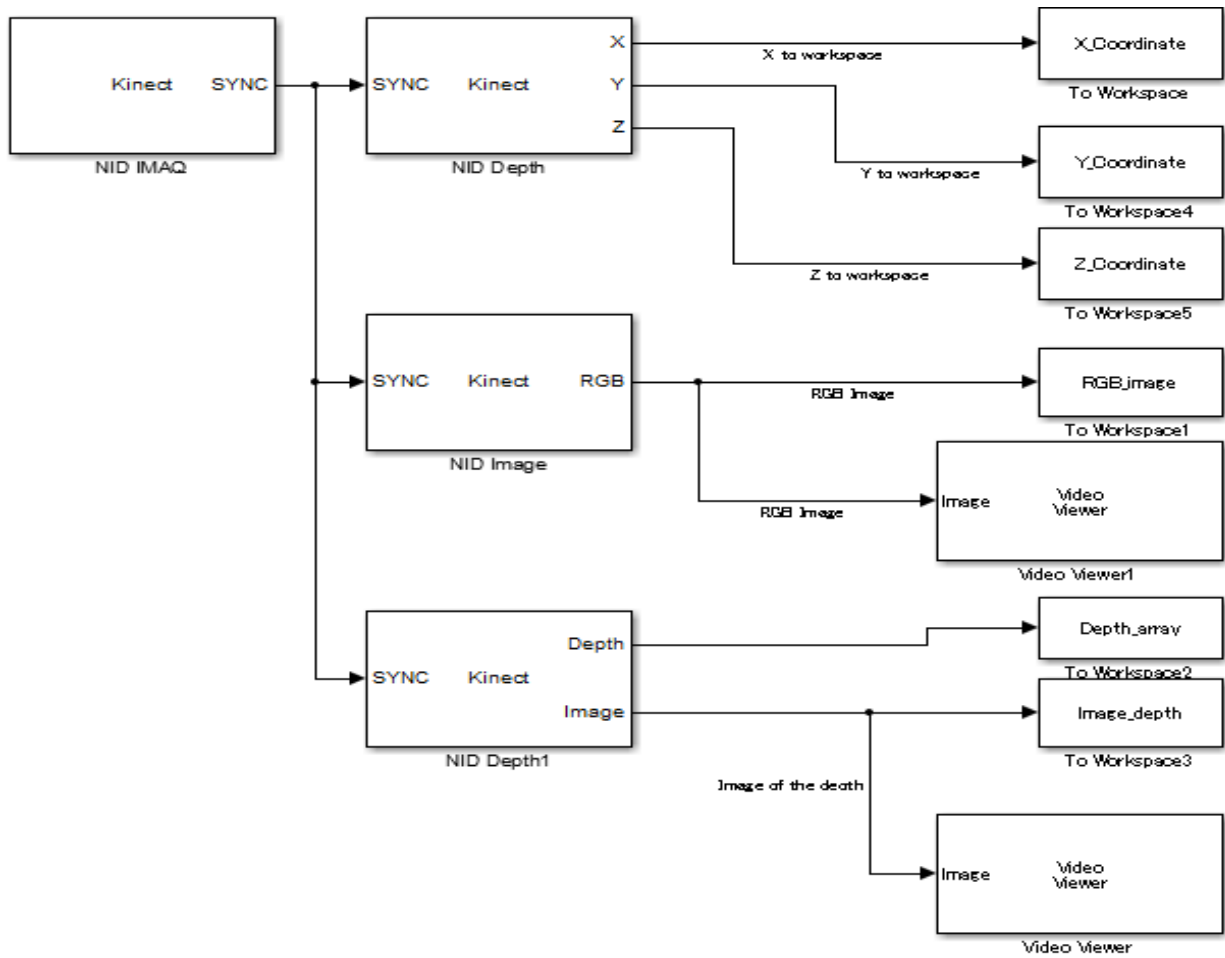


Figure 5.3: Simulink function to extract the XYZ position

As it can be seen in Figure 5.3, Simulink can be used to extract in real time, the X, Y and Z coordinate of each pixel in millimeter, and this information can be used in computing the mushroom cap's size in millimeter. The Kinect Simulink function was modified in such a way that for each axis X, Y and Z of image, a matrix was saved in Matlab workspace, that is to say, the X, Y, Z position of each pixel in the image was saved in millimeter for further processing.

5.3 Temperature Information

An important tip to distinguish the mushrooms that are still growing from the ones that have stopped growing is the temperature information of the mushroom. As such, an infrared camera was employed to investigate the possibility of using thermal images features to inform the robot. Several images were taken from the mushroom growing beds and the results are shown in Figure 5.4 - 5.8. It is proven that growing mushrooms are cooled by an evaporation process because of respiration. When growing stops this effect is reduced and temperature rises. Also in order to compare temperature between mushrooms, an expert mushroom grower selected ripe and unripe mushrooms from samples.

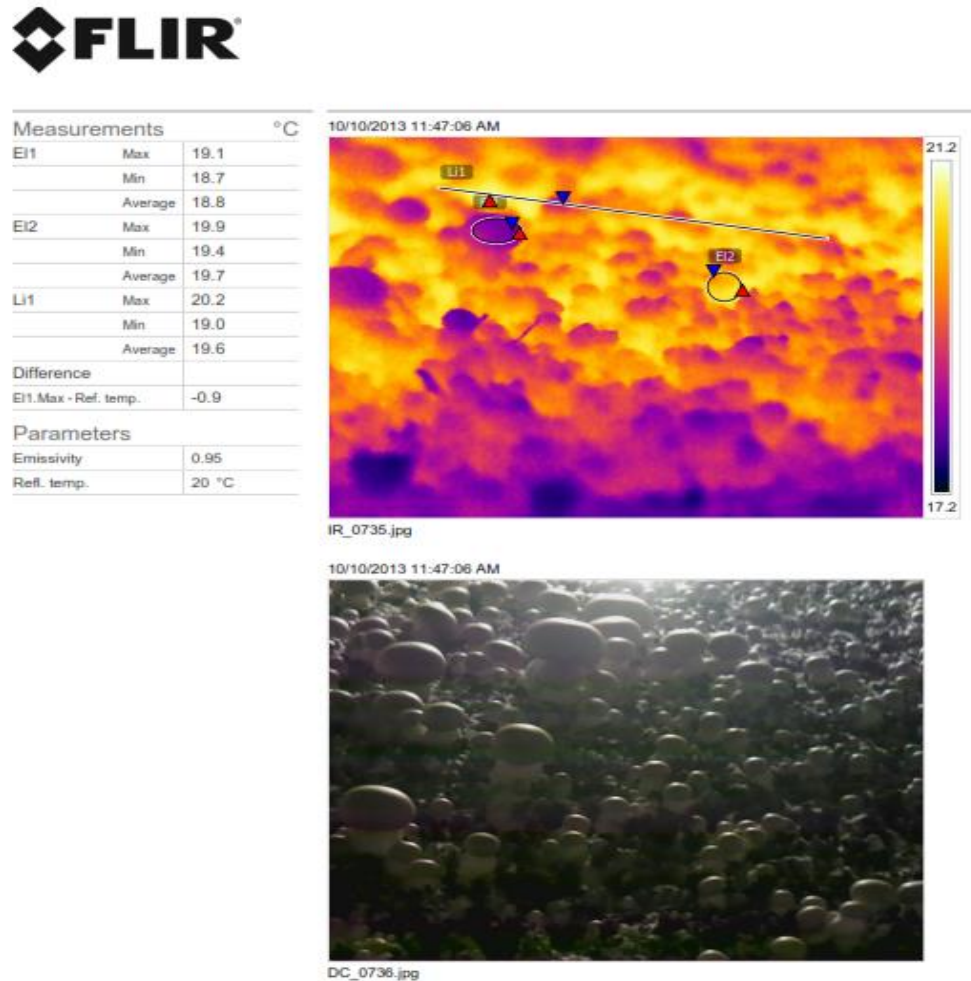


Figure 5.4: Thermal image taken by infrared camera (1)

As an example, the temperature difference between the two mushroom caps in Figure 5.5 is measured and demonstrated. The average temperature in the first area is 18.8 degrees Celsius while the second area has an average temperature of 18.6 degree Celsius. This noticeable temperature difference between these two mushrooms can be used to inform the robot in terms of finding the optimal size mushroom to pick.

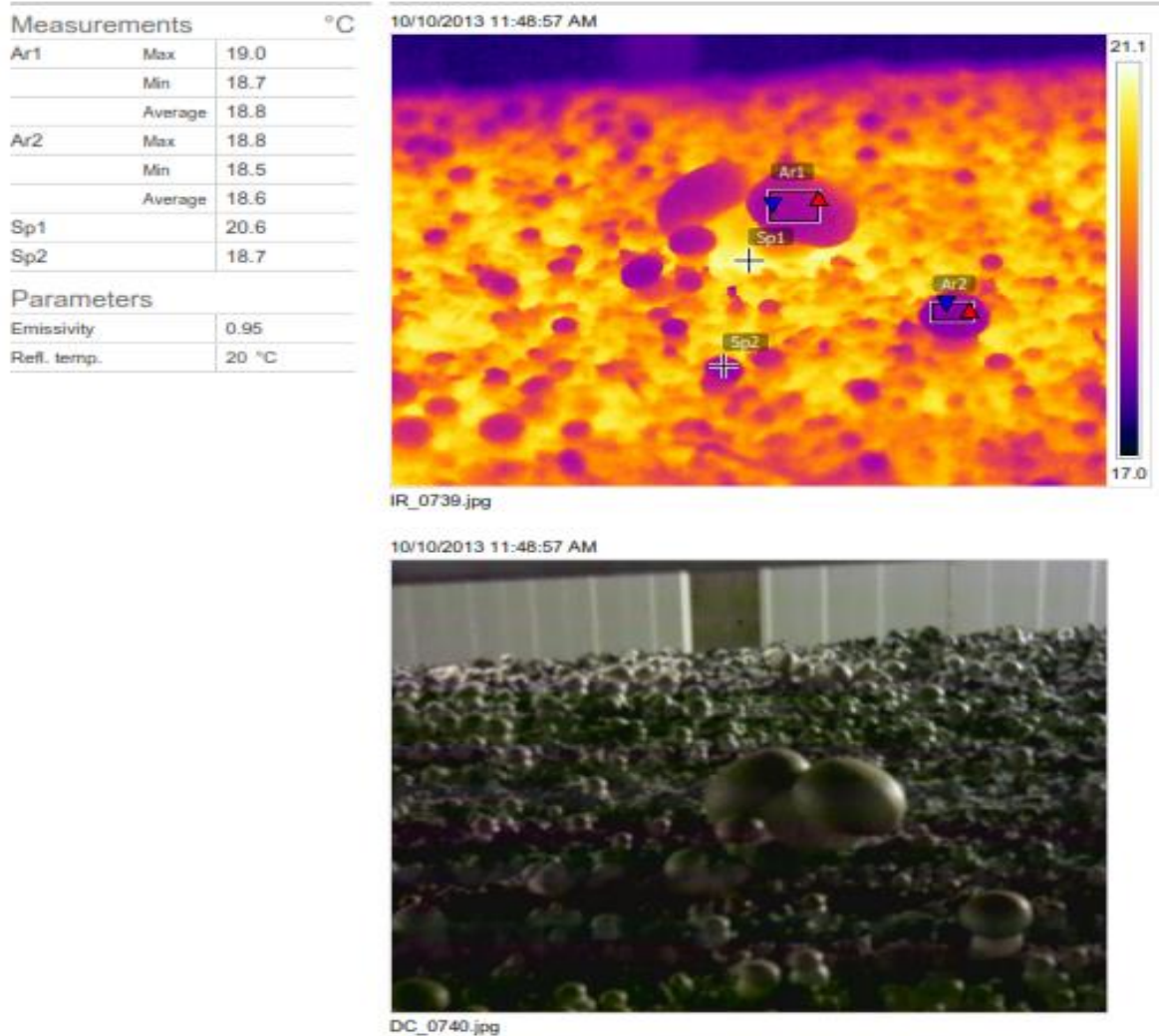
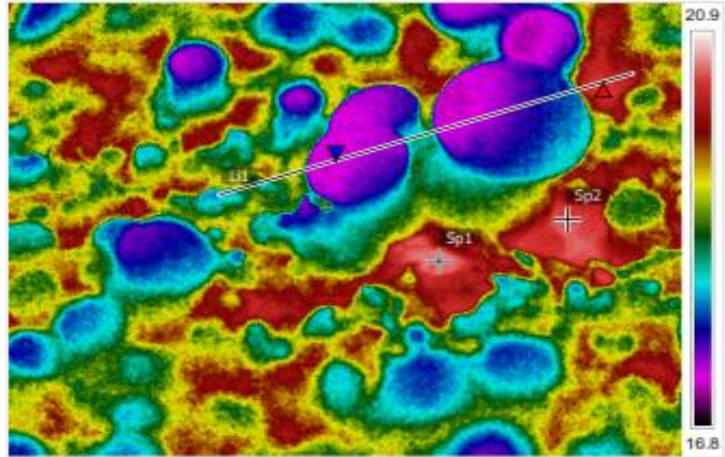


Figure 5.5: Thermal image taken by infrared camera (2)



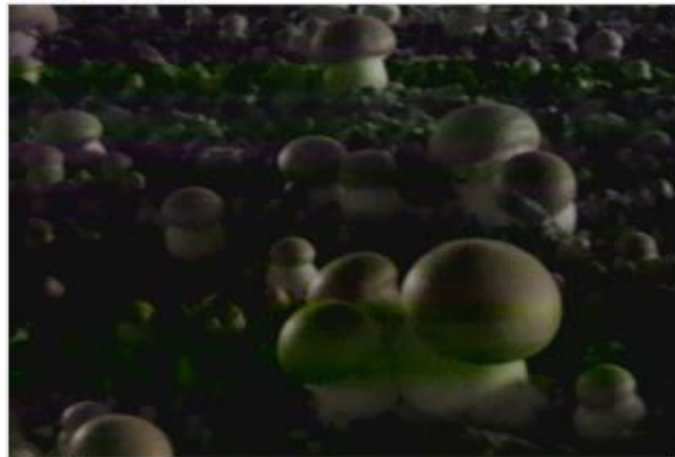
| Measurements | | °C |
|--------------|---------|-------|
| Sp1 | | 19.7 |
| Sp2 | | 19.6 |
| Li1 | Max | 19.6 |
| | Min | 18.1 |
| | Average | 18.6 |
| Parameters | | |
| Emissivity | | 0.95 |
| Refl. temp. | | 20 °C |

10/10/2013 11:49:15 AM



IR_0741.jpg

10/10/2013 11:49:15 AM



DC_0742.jpg

Figure 5.6: Thermal image taken by infrared camera (3)

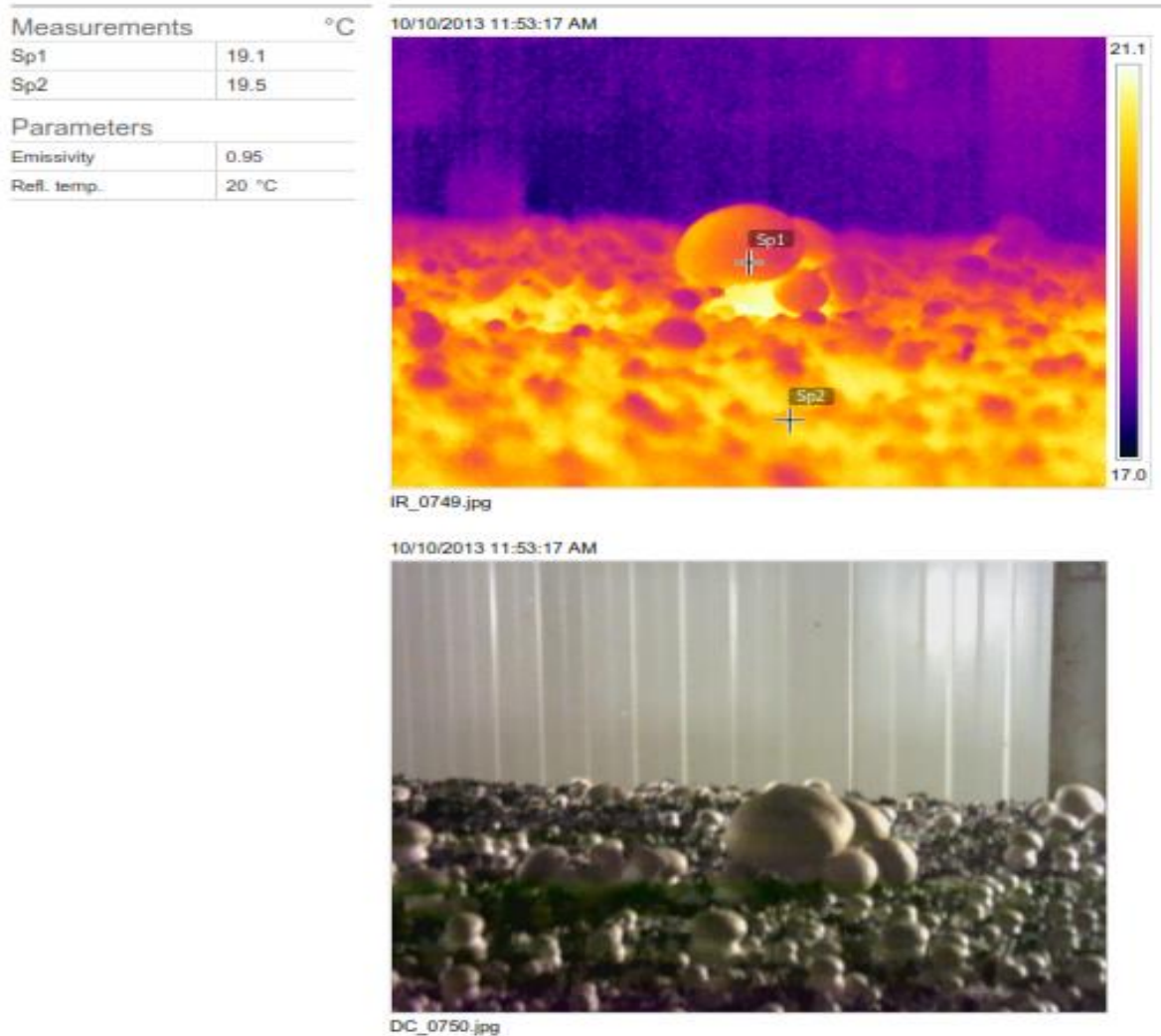


Figure 5.7: Thermal image taken by infrared camera (4)

In the image above, the temperature difference between two spots are demonstrated in the table shown. The temperature for the bigger mushroom is approximately 19.1 degree Celsius while the smaller mushroom has the temperature of 19.5 degree Celsius and this proves that as a mushroom gets bigger, the temperature reduces. As such, the combination of the thermal images of the mushroom and images taken by digital camera can be used to develop an algorithm capable of identifying and appropriately picking the grown mushrooms.

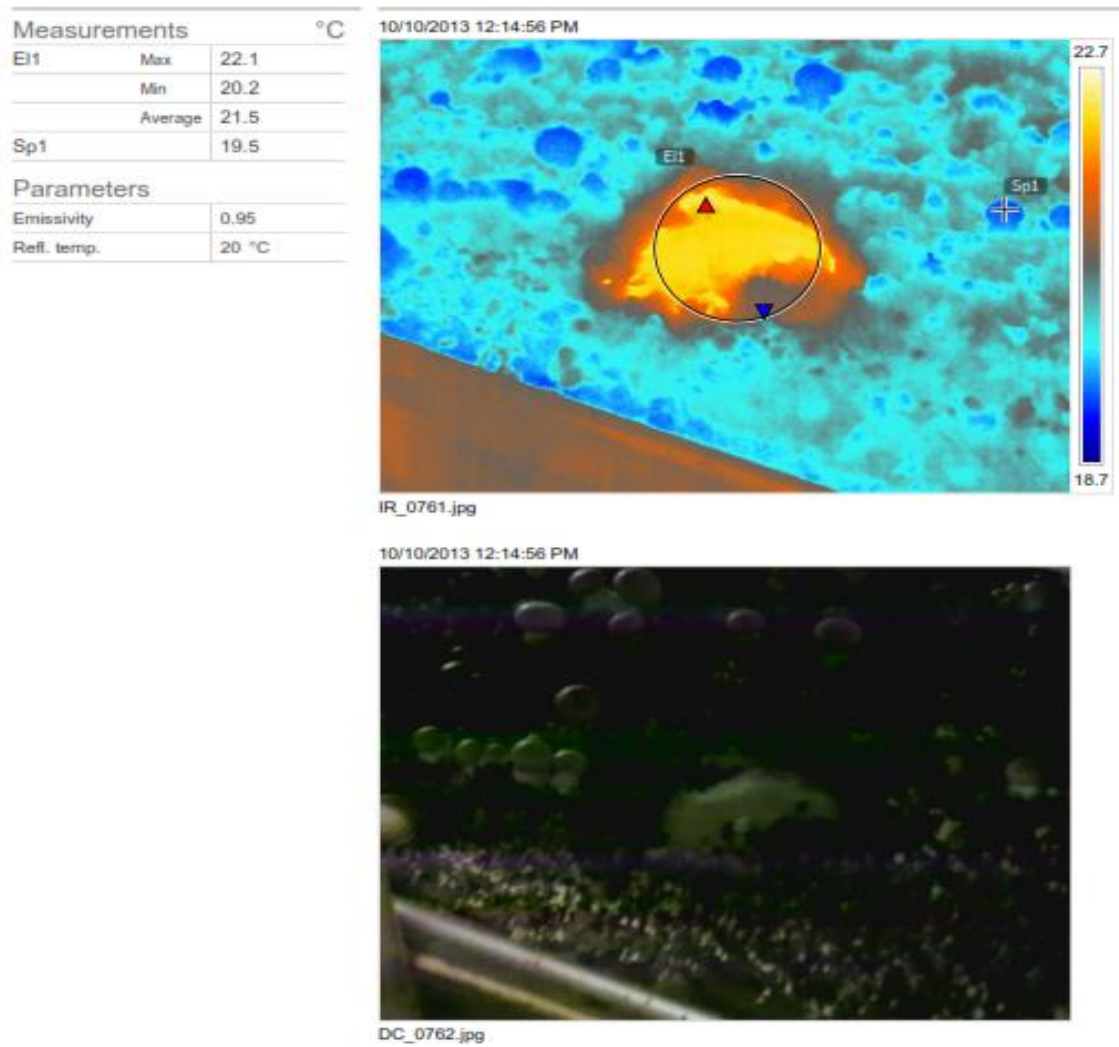


Figure 5.8: Thermal image taken by infrared camera (5)

5.4 Conclusion

In this chapter a sensor fusion based approach based on a number of different non-contact sensing modalities such as stereo and infrared systems was investigated. The mushroom image depth map was extracted and our preliminary results showed that Z-depth information can be extracted from stereo imagery for use in robot end-effector positioning. Finally thermal images indicate that temperature changes throughout the growing cycle of a mushroom can be detected by non-contact means and used to identify ripe mushrooms.

Chapter 6

6 Conclusion

6.1 Introduction

In this Chapter, the contents of the thesis are summarized, the conclusions that have been reached in this thesis are presented and the main contributions of the thesis are stated. The Chapter concludes by suggesting potential future research studies that can be done based on the thesis work.

6.2 Summary

The main objective of this thesis was to investigate a sensor-fusion based approach, based on a number of different non-contacts sensing modalities such as 2-D and 3-D cameras and infrared camera to determine which combination of these inputs can best be used to select ripe, full-grown mushrooms and distinguish diseased mushrooms.

The second objective of this project was to develop algorithms and strategies for identifying and classifying mushrooms by a mushroom harvesting robot. The contents of this thesis are summarized as follows:

In Chapter 1, basic fundamental concepts related to computer vision, machine learning, stereo vision system and hyperspectral imaging were introduced. Previously proposed methods for identifying white mushrooms, microbial spoilage detection, designing vision system of fruit harvesting robot were reviewed and the thesis objectives were stated.

In Chapter 2, a new approach to detect microbial spoilage of mushrooms using a combination of SIFT, BoW and SVM was proposed. The mushroom image feature vector was extracted and quantized using visual vocabulary algorithm and was then used in chapter 3 to feed the SVM classifier. The advantages of using this method over the

hypersepctral imaging as an alternative method were explained. The advantages of this mushroom diseased area detection method were also stated and they include lower processing time and cost.

In Chapter 3, the final mushroom image feature vector that was extracted in chapter 2 was used to classify the mushroom into two main categories; healthy or unhealthy, and the SVM classifier was used for this. A new Matlab based program was developed to detect the contaminated area of the farm and it is employed in on the vision system of mushroom harvesting robot.

In Chapter 4, a new method to identify mushrooms and compute their size was proposed. HSV color space and Hough transform methods were mostly used to detect the size of mushrooms caps. A stereo camera was employed to measure the Z coordinate of each individual mushroom, which was then used to measure the distance between the robot's mechanical arm, and each mushroom. To confirm the feasibility of the proposed algorithm, several images taken at the mushroom farm were evaluated using a Matlab developed program.

In Chapter 5, infrared and stereo cameras were employed to investigate the possibility of using depth and thermal information of the mushroom images to enhance the performance of the identification strategy.

6.3 Conclusions

The following conclusions can be made based on the results of this thesis:

- i. In order to add the capability of detecting contaminated area of the farm to the vision system of the mushroom harvesting robot, SIFT image feature extraction method worked efficiently due to its insensitivity to illumination changes. The combination of SIFT, BOW and SVM algorithm could provide fast and accurate detection of diseased areas on the farm. In our application, the robot was

supposed to find contaminated areas in the mushroom growing bed so an accuracy of more than 90% was required.

- ii. A sensor fusion based approach based on a number of different sensing modalities including digital, stereo and infrared cameras, provided the most useful information to accurately perform recognition and classification of mushrooms.
- iii. Mushroom image segmentation was carried out based on the color information of the mushroom images taken by a high-resolution camera. The color information of the mushroom image is critical to the operation of recognition algorithm. This segmentation method provided sufficient computational speed and accuracy.
- iv. The temperature difference between a growing mushroom and one ready for harvesting is approximately 0.5 degree Celsius. This information coupled with proposed segmentation algorithm could provide a high rate of identification accuracy.

6.4 Contributions

The principal contributions of this thesis are as follows:

- i. A new method to detect the contaminated area of the farm for the vision system of the mushroom harvesting robot was proposed and its operation was explained.
- ii. A new method to accurately identify and locate mushrooms in the mushroom growing beds was proposed and its operation was explained.
- iii. A new Matlab based program to perform identification and classification of mushroom was developed and its feasibility was confirmed by implementing this program for several sample images taken from mushroom farm.

- iv. The possibility of using infrared camera to find the temperature difference between growing mushrooms and the ones that have stopped growing was investigated and the results were demonstrated.

6.5 Proposal for Future Works

The following suggestions are made for future work:

- i. The segmentation method only works for mushrooms that grow vertically to the substrate. For each individual mushroom, an orientation to the substrate needs to be computed and sent to the robot controller, so that the robot arm will pick the mushroom in the right orientation. In this case, accurate depth information can be used to calculate the orientation of mushroom to the mushroom growing bed.
- ii. The thermal images taken by infrared camera need to be registered with high-resolution images in order to assign a temperature value to each individual mushroom and therefore pick the grown mushrooms.

References

- [1] Gonzales, R.C. Woods, R.E, “Digital Image Processing”, *Addison-Wesley*, 2nd edition, Sept1993.
- [2] Z.Q. Liu and N. Barnes, “Knowledge-Based Vision-Guided Robots”, *Springer-Physica Verlag*, Vol. 103, 2002.
- [3] R.B. Smith, “Introduction to Hyperspectral maging with TMIPS”, *Available online at: Micro image tutorial website*, July 2006.
- [4] Tom Mitchell , “Machine Learning”, *McGraw Hill*, 1997.
- [5] Parrish E. and A.K. Goksel, “Pictorial pattern recognition applied to fruit harvesting”, *Transactions of the ASAE*, 1977, pp.822-827.
- [6] DEsnon A.G., G. Rabatel, and R. Pellenc, “A self-propelled robot to pick apples”, *Transactions of the ASAE*, 1987, pp. 87-1037.
- [7] Rabatel G., “A vision system for the fruit picking robot”, *Int. Conf. in Agricultural Engineering*, 1988, Paris.
- [8] Whittaker, Miles, Mitchell, and Gaultney, “Fruit location in a partially occluded image”, *Transactions of the ASAE*, 1987, pp. 591-597.
- [9] Slaughter D. and R.C. Harrel, “Discriminating fruit for robotic harvest using color in natural outdoor scenes”, *Transactions of the ASAE*, 1989, pp. 757-763.
- [10] Grasso G. and M. Recce, “Scene analysis for an orange picking robot”, *Int. Congress for Computer Technology in Agriculture (ICCTA'96)*, 1996.
- [11] Jimenez A.R., R. Ceres, and J.L. Pons, “Shape-based methods for fruit recognition and localization using a laser range finder”, *Int. Workshop on robotics and automated machinery for bio-productions*, 1997, pp. 155-160.

- [12] Moonrinta J., Chaivivatrakul S., Dailey M.N., “Fruit detection, tracking, and 3D reconstruction for crop mapping and yield estimation,” *11th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1181-1186, Dec. 2010.
- [13] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, *Int. Conf. of Computer Vision*, pp. 91–110, Nov. 2004.
- [14] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.
- [15] C. Cortes and V. Vapnik, “Support vector networks”, In *Proceedings of Machine Learning*, vol. 20, pp. 273-297, 1995.
- [16] H.C. Wei, C.C. Chung, and L.C. Jen, “A Practical Guide to Support Vector Classification (Technical report)”, Department of Computer Science and Information Engineering, National Taiwan University.
- [17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.
- [19] Hongpeng Y., Yi C., Yang Simon X., Mittal G.S., “Ripe Tomato Recognition and Localization for a Tomato Harvesting Robotic System,” *Int. Conf. on Soft Computing and Pattern Recognition*, pp. 557-562, Dec. 2009.
- [20] Liyong Q., Qinghua Y., Guanjuan B., Yi X. and Libin Z., “A Dynamic Threshold Segmentation Algorithm for Cucumber Identification in Greenhouse,” *Int. Conf. on Image and Signal Processing*, pp. 17-19, Oct. 2009.
- [21] Jin W., De Z., Wei J., Jun-Jun T. and Ying Z., “Application of support vector machine to apple recognition using in apple harvesting robot,” *Int. Conf. on Information and Automation*, pp. 1110-1115, June 2009.

- [22] Hongshe D., Jinguo S. and Qin G, "A Fruit Size Detecting and Grading System Based on Image Processing," *Int. Conf. on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp.83,86, Aug. 2010
- [23] Sural S., Gang Q. and Pramanik, S., "Segmentation and histogram generation using the HSV color space for image retrieval," *Int. Conf. on Image Processing*. Pp. 589-592, vol.2, 2002
- [24] A.R. Jimenez, R. Ceres and J. L. Pons, "A Survey of Computer Vision Methods for Locating Fruit on Trees", *Transaction of the ASAE*, pp. 1911-1920, 2000.
- [25] R.D. Tillett, B.J. Batchelor, "An algorithm for locating mushrooms in a growing bed," *Journal of Computers and Electronics in Agriculture*, vol. 6, issue 3, pp. 191-200, Dec. 1991.
- [26] J.N. Reed, R.D. Tillett, "Initial experiments in robotic mushroom harvesting," *Journal of Mechatronics*, vol. 4, issue 3, pp. 265-279, April. 1994.
- [27] R. Noble, J.N. Reed, S. Miles, A.F. Jackson, J. Butler, "Influence of mushroom strains and population density on the performance of a robotic harvester," *Journal of Agricultural Engineering Research*, vol. 68, issue 3, pp. 215-222, Nov. 1997.
- [28] J.N. Reed, S.J. Miles, J. Butlery, M. Baldwin, R. Noble, "Automatic Mushroom Harvester Development," *Journal of Agricultural Engineering Research*, vol. 78, issue 1, pp.15-23, Jan. 2001.
- [29] C. Connolly, "Gripping developments at Silsoe," *Journal of Industrial Robot*, vol. 30, issue 4, pp. 322-325, 2003.
- [30] Center for Concept in Mechatronics, "Mushroom Picking Robot," Available online at <http://www.ccm.nl/en/projects/project-archive/28-mushroom-picking-robot>.
- [31] G. Yu, Y. Zhao, G. Li, H. Shi, "Algorithm for locating individual mushroom and description of its contour using machine vision," *Trans. Chinese Society of Agricultural Engineering*, vol. 21, issue 6, pp. 101-104, June. 2005.

- [32] S.G. Paine, "Studies in bacteriologist, a brown blotch disease of cultivated mushrooms". *Annals of Applied Biology*, 1919, pp. 206-219.
- [33] J.M. Olivier, J. Guillaumes and D. Martin, "Study of a bacterial disease of mushroom caps", *4th Int. Conf. on Plant Pathogenic Bacteria*, 1978. Proceedings p.903.
- [34] T. Vizhanyo and J. Felfoldi, "Enhancing colour differences in images of diseased mushrooms". *Computers and Electronics in Agriculture*, vol. 26, pp. 187-198, April 2000.
- [35] A.A. Gowen, C.P. Donnell, P.J. Cullen, G. Downey, J.M. Frias, "Hyperspectral imaging, an emerging process analytical tool for food quality and safety control", *Trends in food science and technology*, vol. 18, pp. 590-598, Dec. 2007.
- [36] E. Gaston, "Visible and Hyperspectral Imaging Systems for the Detection and Discrimination of Mechanical and Microbiological Damage of Mushrooms". *PhD thesis, Dublin Institute of Technology*, 2010.
- [37] Park, M., Jin J.S., Peng Y., Summons P., Yu D., Cui, "Automatic cell segmentation in microscopic color images using ellipse fitting and watershed," *Int. Conf. on Complex Medical Engineering (CME)*, pp. 69-74, July 2010
- [38] Xiaomin L., Yuanyuan W., Yinhui D. and Jinhua Y, "Cell Segmentation Using Ellipse Curve Segmentation and Classification," *Int. Conf. on Information Science and Engineering (ICISE)*, pp. 1187-1190, Dec. 2009
- [39] J.S. Lim, "Two-Dimensional Signal and Image Processing", *Englewood Cliffs, Prentice Hall*, pp. 469-476, 1990.
- [40] G. Csurka , C.R. Dance , L. Fan , J. Willamowski and C. Bray, "Visual categorization with bags of key points", *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004

- [41] G. McLachlan, K.A. Do, C. Ambroise, “Analyzing microarray gene expression data”, Wiley, 2004.
- [42] T.F Cox, M.A.A Cox, “Multidimensional scaling”, *Chapman & Hall*, London, 2001.
- [43] Bradley, A. P., “The use of the area under the ROC curve in the evaluation of machine learning algorithms”, *Pattern Recognition*, pp. 1145-1159, 1997.
- [44] The VLFeat open source library, [Online], *Available at* <http://www.vlfeat.org>
- [45] Sharma, Gaurav, and H. J. Trussell, “Digital color imaging”, *Image Processing, IEEE Transactions*, pp. 901-932.
- [46] John A. H., Manchek A. W., “A k-means clustering algorithm”, *Journal of the Royal Statistical Society*. pp. 100-108, 1979.
- [47] Otsu, N., “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 62-66, 1979.
- [48] J. Illingworth , J. Kittler, “The adaptive Hough transform”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 690-698, Sept. 1987
- [49] Peng T, Balijepalli A, Gupta SK, LeBrun T, “Algorithms for On-Line Monitoring of Micro Spheres in an Optical Tweezers-Based Assembly Cell”, *J. Comput. Inf. Sci. Eng.* 2007, pp. 330-338
- [50] Ito. Y., Ohyama. W., Wakabayashi. T., Kimura. F., “Detection of eyes by circular Hough transform and histogram of gradient,” *Int. Conf. on Pattern Recognition (ICPR)*, pp.1795-1798, Nov. 2012
- [51] Robo Realm-vision for machines, [Online], *Available at* <http://www.roborealm.com/>

Curriculum Vitae

Name: Alireza Masoudian

Post-secondary Education and Degrees: 2005 - 2009 B.Sc.
Electrical Engineering
Azad University, Central Tehran Branch
Tehran, Iran

Jan 2012 – Sept 2012 MEng
Electrical and Computer Engineering
The University of Western Ontario
London, ON

Sept 2012 – Dec 2013 MEng
Electrical and Computer Engineering
The University of Western Ontario
London, ON

Publications:

- [1] Masoudian, A., McIsaac, K.A., “Application of Support Vector Machine to Detect Microbial Spoilage of Mushrooms,’ *Int. Conf. on Computer and Robot Vision (CRV)*, pp. 281-287, May 2013, Canada
- [2] Another paper related to segmentation part of this project is ready to publish

Work Experiences:

Teaching Assistant

The University of Western Ontario, London, ON Sept 2012- Dec 2013

- Assisted lecturer in “Digital Logic Systems” and “Introduction to Electrical Engineering” courses

Research Assistant

The University of Western Ontario, London, ON Sept 2012- Dec 2013

- Researched and conducted feasibility study to determine appropriate computer vision algorithms for an automated harvester

- Developed MATLAB-based program for a vision system of an automated harvester

Electrical and Control Engineer

Automotive Industries Development Co., Tehran, Iran

Sept 2009-Sept 2011

- Designed and analyzed control and automation systems for Schuler press robotic automation lines
- Reduced the down time of a Schuler press robotic automation line and increased production efficiency
- Reduced press Automatic Die Change (ADC) time and increased production efficiency