PINEAPPLE SLICES CLASSIFICATION USING A HYBRID FEATURE EXTRACTION TECHNIQUE AND MULTICLASS SVM

JOHN NYUTU KAMAU

MASTER OF SCIENCE

(Electrical Engineering)

JOMO KENYATTA UNIVERSITY

OF

AGRICULTURE AND TECHNOLOGY

2022

Pineapple Slices Classification Using a Hybrid Feature Extraction Technique and Multiclass SVM

John Nyutu Kamau

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering of the Jomo Kenyatta University of Agriculture and Technology

2022

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signature…………………………….. Date…………………….

John Nyutu Kamau

This thesis has been submitted for examination with our approval as university supervisors:

Signature……………………………… Date……………………..

Prof. P. K. Hinga, PhD

JKUAT, Kenya

Signature……………………………. Date……………………..

Prof. S.I. Kamau, PhD

JKUAT, Kenya

DEDICATION

This thesis work is dedicated to my children; Edward, Margaret and Nancy. You have made me better, stronger and more fulfilled than I could have ever imagined.

ACKNOWLEDGMENT

First and foremost, I want to express my gratitude to God Almighty for providing me with the strength, knowledge, skill, and chance to conduct this research study, endure, and successfully complete it. This accomplishment would not have been achieved without his blessings.

Secondly, I'm very grateful to my supervisors; Prof. Hinga and Prof. Kamau for their overwhelming support and supervision throughout my study, and for their motivation, patience, and immense knowledge.

I would also like to thank my correction supervisors; Dr.Martin, Dr. Irungu, and Dr. Kihato, for their insightful comments and encouragement, but also for the hard questions that incented me to widen my research from various perspectives.

My sincere thanks also go to Del Monte Kenya who provided me an opportunity to do research in their factory, and who gave access to the cannery production facilities. Without their precious support, it would not be possible to conduct this research.

Finally, I would like to thank my family for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

2-D	Two Dimensions
3-D	Three Dimensions
A/D	Analog to Digital
CV	Computer Vision
DAG-SVM	Directed Acyclic Graph Support Vector Machine
DDC	Data Driven Control
GLCM	Gray Level Co-occurrence Matrix
GRB	Gaussian Radial Basis
GUI	Graphical User Interface
HPOL	Homogeneous Polynomial
HSV	Hue, Saturation, Value
kSVM	kernel Support Vector Machine
L*a*b	Lab colour space (L for lightness, a for green–red and b for blue yellow)
LIN	Linear
MBC	Model Based Control

MWV-SVM	Max-Wins-Voting Support Vector Machine
PCA	Principal Component Analysis
RGB	Red, Green, Blue
SVM	Support Vector Machine
SVM ML	Support Vector Machine Machine-Learning
WTA-SVM	Winner-Takes-All Support Vector Machine

ABSTRACT

Automatic fruit quality inspection is important to improve quality and reduce production costs. Fruit grading and sorting is primarily visual. Real-time fruit inspection and grading requires high-quality photos with easily recognized features and fast software and hardware to process the images. Since forever, pineapples have been graded using a labour intensive and expensive method. Due to high labour costs and inconsistent in manual methods, industries have turned to automation. Visual sorting of pineapple slices is challenging to automate in food processing. The research intends to build a computer vision based algorithm for sorting and automating pineapple slices based on texture, colour and shape. This research designs a unique classification approach based on Max-Wins-Voting SVM employing Gaussian Radial Basis kernel for real-time automatic implementation. First, digital photos of pineapple slices are obtained at factory floor and their backgrounds are removed using Otsu segmentation. Then, features (colour, colour moment, texture, and shape) of each image are extracted using a hybrid method to create a feature space. Third, Principal Component Analysis reduces feature space dimensions. Finally, multi-class SVMs (Max-Wins-Voting SVM, Directed Acyclic Graph SVM and Winner-Takes-All SVM) are built. The accuracy and calculation time of SVMs with different kernels (Linear, Gaussian Radial Basis, and Homogeneous Polynomial) are compared. SVMs are trained with the reduced feature space vector using 5-fold stratified cross validation. Max-Wins-Voting SVM utilizing Gaussian Radial Basis kernel was found reliable and robust, as shown in result. The approach provides fast, accurate real-time automation of pineapple slice sorting achieving over 90% accuracy. The method classifies as good as human operator with added advantage of high quality standards and reduced production costs.

CHAPTER ONE

INTRODUCTION

1.1 Pineapple cannery processing

Worldwide, the second most traded fruit across the globe is pineapple, after bananas. Kenya's largest single manufactured export is canned pineapple, and the country ranks among the top five pineapple exporters in the world (Wangwe, 1995). Every year, tons of pineapples are cultivated, harvested, processed, and consumed all over the world. Pineapple planting and harvesting are done manually by humans using basic optical sensory mechanism. The harvested fruit are loaded to a lorry and ferried to the factory for processing. The fruits are first washed in flume, graded into various sizes and then processed through a Ginaca machine, where they are skinned and cored in a single operation. Ginaca is a machine that can peel and core pineapples automatically. It was invented by an American engineer, Henry Gabriel Ginaca at the direction of Hawaii pineapple magnate James Dole. The output of Ginaca operation are three products.

First is pineapple cylinder which is then processed into pineapple slices through trimming, slicing, grading and sorting processes. Each pineapple cylinder produces 8-10 ring slices of about 2cm thick. These slices are then sorted and graded by a human operators using manual method based on conventional visual quality inspection performed. The method is time-consuming, slow, tedious and non-consistent. The research seek to automate this process and save the production cost and improve the quality standards.

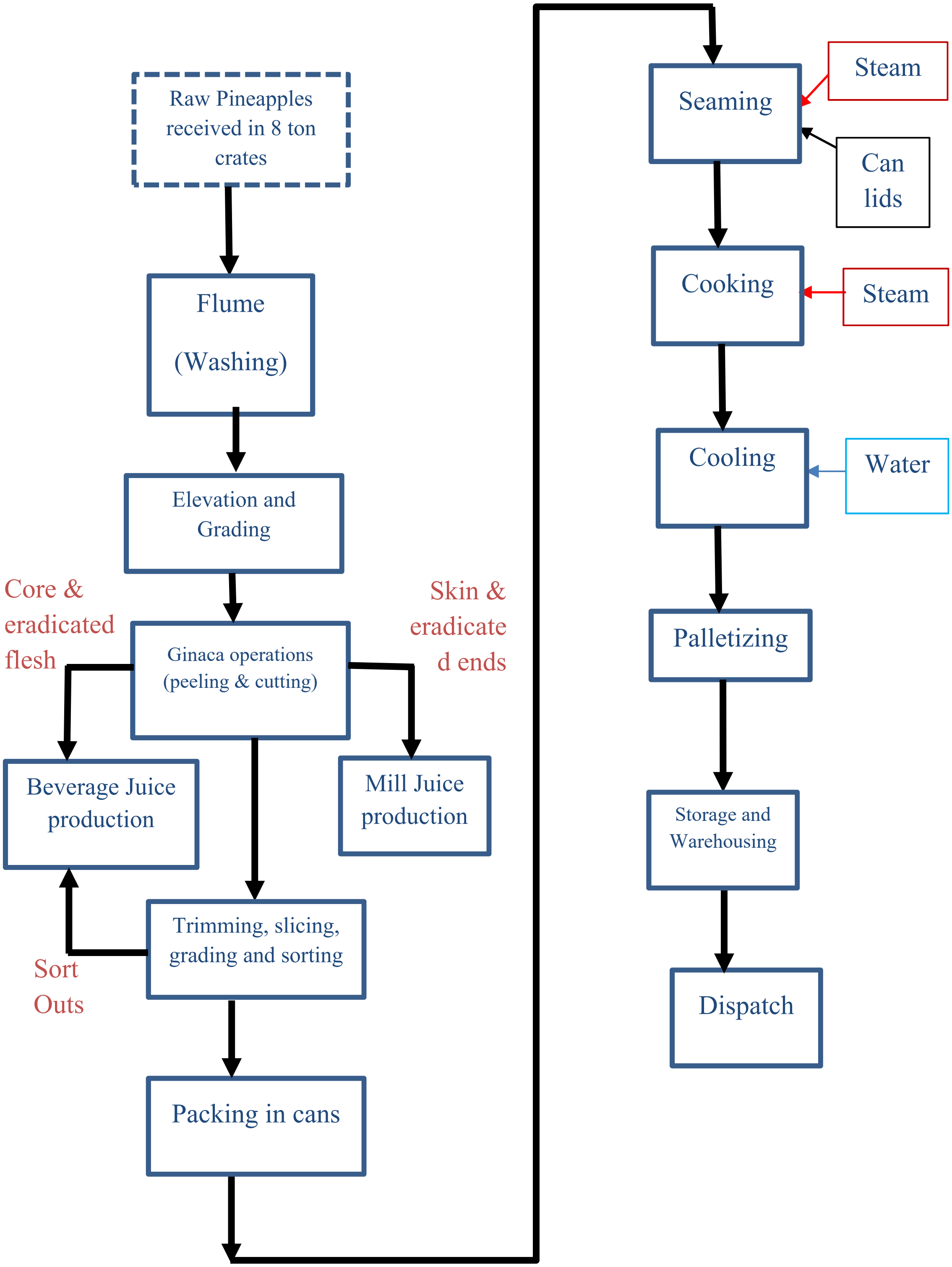


Figure 1.1: Pineapple processing

The slices are then packed into cans, seamed, heat treated for preservation, cooled, palletized and then incubated in warehouses. Then the product is labelled according to customer specification and dispatched to its destination. Figure 1.1 shows canned pineapple process flow diagram.

The second Ginaca product is core and eradicated fresh which are used for beverage juice production. The third Ginaca product skin and eradicated ends which are processed for mill juice production.

From the demerit of manual method, it is important to look for the possibilities of adopting quicker systems, which will be more accurate and save time in grading and sorting of pineapple slices. Automated computer vision system for grading and sorting is one of such reliable method. Using computer vision technique, a consistent, superior speed, cost effective and accurate grading method can be achieved.

To minimize the losses incurred during harvesting, production and marketing, manufacturing and processing sectors requires automated inspection and grading systems. Assisted grading and sorting of pineapple slices can be done based on colour, shape, appearance texture, and sizes. Recently machine vision systems have been broadly used to evaluate external quality of other fruits (Heinemann, Varghese, Morrow, Sommer III, & Crasswelle, 1995), (Zou, Zhao, & Li, 2010). However, the conventional systems can't be applied directly to pineapple problem, because of the high uniqueness and complexity of the problem. The pineapple slices have a great similarity in term of colour, texture and shape. Also they have noticeable difference from one slices of one type of pineapple to another. This make the pineapple slices classification very unique.

1.2 Background of food processing technology

Computer Vision (CV) was first applied to food processing fields in 1989 for grain quality inspection (Zayas, Pomeranz, & Lai, 1989). One of the most important human senses is vision and hence the increase in dominance of computer vision. In recent years, the CV technique has gained popularity in fruit grading. For more than four decades, computer vision method has been a subject of research and application. This method is applied in many engineering fields such as industrial image processing, food processing, robotics and other fields. Some of benefits in favour of application of computer vision to engineering problems include non-destructive evaluation possibilities, quickness, easy procedures for application and quantity of output per unit time.

Large post-harvest losses in processing and handling, as well as rising demand for safe, high-quality food items, necessitate the expansion of objective, quick, and precise quality control of food and agricultural goods (Narendra & Hareesh, 2010). Main areas of application of computer vision technology in food industry include quality evaluation of fruits, food grains, vegetables and processed foods. The method had also been used for determination of blemishes in vegetables and fruits and insect infestation in grains.

The fresh agricultural products-appearance is a main criterion in influencing purchasing decisions. Image processing techniques are regularly combined with instrumental and mechanical devices in order to make an automated system for food quality evaluation and to replace human manipulative effort in the carrying out a given process. In order to provide high-quality products to the customers, quality inspection is very significant.

Currently in many industries, grading is performed mainly by visual inspection for a specific quality attribute. The application of image processing for sorting is being used to many products, such as carrots, green peppers, tomatoes oranges, potatoes, peaches, and apples. Modern guidance research includes harvesting tomatoes, mushrooms, oranges, apples, melons and cucumbers.

1.3 Problem Statement

Pineapple slices sorting and grading is done manually. This task is tiresome, labour intensive monotonous and repetitive task. This manual method can be automated using machine vision technology and save pineapple industries huge production cost. This research develop an algorithm to automated pineapple slices classification by solving the main challenges for real-time inspection and grading of fruit.

1.4 Justification

Pineapple slices classification has entirely employed costly traditional labour intensive manual method but due to its high cost, tiresome monotonous activity, low quality products demanding close supervision and lack of overall consistency, industries have resulted in a search for automated solutions and this research seek to address the problem using computer vision based system addressing the main challenges for real-time inspection and grading of fruit which are to produce quality images that deliver clearly identifiable features and to have both efficient software and hardware to process the images quick enough for real-time implementation. The MWV SVM method chosen offer the following benefits:

i). The CV method is reliable, quick, and objective inspection technology that has been used in a wide range of sectors. Its accuracy and speed satisfy ever-increasing quality and production requirements, hence helping in the growth of fully automated processes.

ii). The traditional labour intensive method of grading the fruit has proved to be very expensive due to high labour cost and a lack of overall consistency. With automation the labour cost will be greatly reduced.

iii). The automatic sorting and packing machine will increase throughput and productivity, improve quality and predictability of quality, improve consistency of the pineapple sorting processes and reduced direct human labour costs and expenses. This will lead to an exponential growth to commercial pineapple production and contribute greatly to the economy.

iv). Support Vector Machines (SVMs) are a machine learning (ML)-based advanced classification technology (Patil, Shelokar, Jayaraman, & Kulkarni, Regression models using pattern search assisted least square support vector machines, 2005). SVMs have considerable advantages over other methods such as artificial neural networks, decision trees and Bayesian networks due of their exquisite mathematical tractability, direct geometric interpretation and high accuracy.

v). Likewise, a hybrid feature extraction technique using colour moment, texture, colour histogram and shape features will produce better and identifiable features more effective than using any single feature in classification of pineapple slices.

1.5 Objectives

1.5.1 Main Objectives

The main objective is to develop an algorithm to automate the manual sorting and grading of pineapple slices using multiclass SVM and a hybrid feature extraction technique.

1.5.2 Specific Objectives

To achieve the main objective the following specific objective are done:

1. To develop an image preprocessing Otsu's algorithm and obtain a segmented output image.
2. To extract 90 features from each output image and create a feature space, then reduce its dimensions using PCA.
3. To build three multiclass-SVM classifiers and three kernels and tune parameters to achieve target accuracy of above 85%.
4. To compare the three m-SVMs performance on pineapple slice classification and select the best method.
5. Compare selected method for pineapple slice classification with J. I. Asnor, 2013, pineapple classification using RGB colour that achieved 75% accuracy.

1.6 Scope of the Research

This study will cover a pineapple slices classification in a canning industry. The study will include visiting the canning process at the floor of factory, observe and study the process of slices sorting and grading. 250 photos of actual sorted slices into five categories will be taken, each category containing 50 slices. Then develop an algorithm which will classifier these sample photos of slices taken in the actual cannery setup. The algorithm will be simulated, tested in MATLAB platform and results generated and documented.

The study will not include building the real model or prototype and will not deal with legal issues raised on job losses as result of automation. Since the study was done using MATLAB algorithm simulation, the actual model implementation has not be covered as it’s beyond the scope of this study. The study assumes the following effect of application of AI in pineapple will be pre addressed:

i). Expected a cultural and skills shift

In three to five years, organizational culture and employee skills will alter dramatically. Due to human nature's resistance to change, AI adoption must start with the CEO and permeate throughout the workforce. Management's leadership will promote cultural shift acceptability.

ii). AI lacks an important human trait: Empathy

AI replaces mundane, repetitive, and error-prone tasks so humans may focus on creative, flexible processes. Empathy and judgment are traits robots lack.

1.7 Organization of thesis

The thesis is organized into five chapters, which include Chapter 1, introduction, which introduces the thesis research area in Computer Vision. It then describe pineapple-processing stages, identified problem, justify the method chosen to solve it and highlight objectives and scope

Chapter 2 deals with Literature Review. The chapter covers an overview of related work, reviews the advancement of computer vision in the food processing and agricultural, an overview of the numerous characteristics and types of CV technology applications in the fruit sector. The motivation for employing computer vision algorithms, data driven control is covered in detail and its great relevance to the study as well as the advantages they provide, are examined, and a research gap is identified.

Chapter 3 discusses methodology in details. It start with colleting of images of pineapple slices at Del Monte Kenya, preprocessing them to remove the background and only focus on the slices. Then extract features of the images and make a feature space which is used to train the SVM classifiers and automate the slices sorting and grading.

Chapter 4 presents the results obtained from the algorithm simulation. The performance of 3 types of multiclass SVMs: MWV-SVM, WTA-SVM and DAG-SVM is compared in terms of accuracy and computation time. Also, three kinds of kernels is used: linear kernel, Homogeneous Polynomial kernel and Gaussian Radial Basis kernel and proposed method justified. The result are discussed in details.

Chapter 5 the study concludes that automation of the manual sorting of pineapple slices is possible using computer vision techniques. A hybrid feature extraction techniques is found reliable in producing the required features to sort the pineapple slices into five categories required. The study end by identifying some limitation that could not be solved and were beyond the scope of the study and they form the basis of future researches. This section is followed by reference materials used in the development of the research thesis.

1.8 Contribution of the thesis

The key contribution made through this research is to automate the manual pineapple slices sorting and grading. The research develop an innovative classification method based on the Max-Wins-Voting SVM using Gaussian Radial Basis kernel with the desirable objective of fast and accurate classification for real-time automatic implementation. The automatic sorting and packing machine will increase throughput and productivity, improve quality, predictability of quality and consistency of the pineapple sorting processes. This is expected to directly reduced human labour costs and expenses. This will lead to an exponential growth to commercial pineapple production and contribute greatly to the economy.

CHAPTER TWO

LITERATURE REVIEW

This chapter reviews and work on existing computer vision technique employed in fruit sorting and grading, highlight their merit and shortcoming and bring out the existing research gap which this work intends to fill.

2.1 Overview

Manual grading and sorting techniques are based on conventional visual quality inspection performed by human operators, which is time-consuming, slow, non-consistent and tedious. Colour, appearance, texture and odor are the quality attributes frequently used for deciding on the harvest maturity (Sun & Brosnan, 2003). It is important to search the possibilities of adopting quicker methods that are more accurate and save time in grading and sorting of agricultural and food products, especially in large scale production. Automated computer vision system is one of such reliable method for grading and sorting. This research reviews the advancement of computer vision in the food processing and agricultural industry. A consistent, superior speed, cost effective and accurate grading and sorting can be done using computer vision technique. To minimize the losses incurred during harvesting, production and marketing, manufacturing and processing sectors requires automated inspection and grading systems. Grading and sorting of pineapple slices can be done based on colour, shape, appearance texture, and sizes.

Computer Vision (CV) is a technique that uses cameras and image analysis tools to automate visual assessment and measurement operations. This section gives an overview of the different elements and types of CV technology applications in the fruit business. The motivation for employing computer vision algorithms, as well as the advantages they provide, are examined (Lind & Murhed, 2012) .

2.1.1 Fruit Grading Systems

Table 2.1 Summary of various fruits and their parameters (Rashmi, Sapan, & Roma M., 2013)

Fruit type	Various Parameters	Percentage Accuracy	Reference
Apples	Colour	95 %	Kazuhiro N,1997
	Calyx, Colour, stem and defect	78 %	Leemans V, 2002
	Colour, shape, texture	72 %	Leemans V, 2004
	Bruises , Stem end and calyx	89%	Qiabao Xu, 2009
		94%	Zhang Dong, 2013
Orange	Colour	70%	Levi P,1988
	Shape	75%	Slaughter D,1987
	Colour and Intensity	80%	Juste F,1991
	Colour	87%	Dah-Jye L,2011
Tomatoes	Shape	68%	Whitakers,1987
		84%	Rokunuzzaman
	Colour	88%	M,2013
		90%	Buemi F,1995
		95%	Dah-Jye L,2011
	FD (Fractal dimension) and Colour	89% 85%	Zheng Hong,2012

Image processing has been extensively used for grading of fruits into uniform categories (shape, colour, size and texture, bruises, calyx and stem). Grading has been applied to many vegetables and fruits including potatoes, carrots, green paper, peaches, apples, oranges, tomatoes. In this subsection, review is made on how various parameters can be used for automatic fruit sorting and grading system.

To sort and grade apples, neural network plays important role to classify apples into five quality classes (Nakano, 1997). Pixels are acquired from image based on colour and are fed to the neural network as an input. Mean colour of fruit is acquired from fruit image and sorting is done based on fruit mean colour, its availability and variability of diseased pixels and ratio of red colour of fruit image.

Precision was attained up to 95%. Table 2.1 gives summary of various fruits with various parameters and their accuracy.

2.1.2 Colour features extraction methods and algorithms

Table 2.2 Colour feature extraction techniques and their percentage accuracy (Rashmi, Sapan, & Roma M., 2013)

Fruit types	Various Method	Advantages	Disadvantages	% Accuracy	Reference
Pineapple	Colour space RGB	Better for colour display	i). Require better preprocessing algorithm to improve accuracy of grading pineapple ii). Not best for colour image, due to the high correlation between R, G, and B component.	75%	J. I. Asnor,2013

Apple	RGB colour space to Nine colour characteristic data	Classification judgment of 95%.	Poor Ability to recognize fruit in A-grade.	70.76%	Kazuhiro Nakano,1997
	HSI colour space to 17 colour features data	i). In the classification of two of the four apple colour grades, namely ‘Extra' and ‘Reject,' high grade judgment ratios were achieved. ii).. OFPs method had higher accuracy than BP-ANN	i). Slow process to grade apple in class 1 and class 2 ii). OFPs identification method was poor than SVM.	-	Zou Xiaobo,2007
	HSV colour space	Successful approach to categorize apples in defected and non-defected class.	Very low contrast between healthy skin and defected skin.	-	Shivleela R Arlimatti,2012

Tomato	Method of Dominant colour histogram matching	Appropriate for real-time application and has high accuracy	Method is less sensitive to lighting variations	97%	Li Changyong,2009
	Method of Direct colour mapping	i). Effective, simple and high accuracy ii). The overall sorting and grading results based on both consistency averages and colour index are consistent with human grading.	i). Method is not generalized for different type of fruits.	95 %	Lee Dah-Jye,2010

Most of the sorting and grading system depends on the colour extracted from the image. Colour is the most visually prominent feature of any image. Therefore, colour features mining plays an important role in developing sorting and grading system and also to detect defective fruits from normal fruits. In this subsection colour feature extraction techniques of fruit grading system are explained in table 2.2.

2.2 Computer Vision

Computer vision is a technology that uses cameras and image analysis tools to automate visual evaluation and measuring tasks. Image data is used by the computer to conduct predefined measuring tasks and to derive inferences based on the results. In most cases, a vision system works in four-step cycles as shown in Figure. 2.1:

i). Take an image

ii). Analyze the image

iii). Report result

iv). Take action

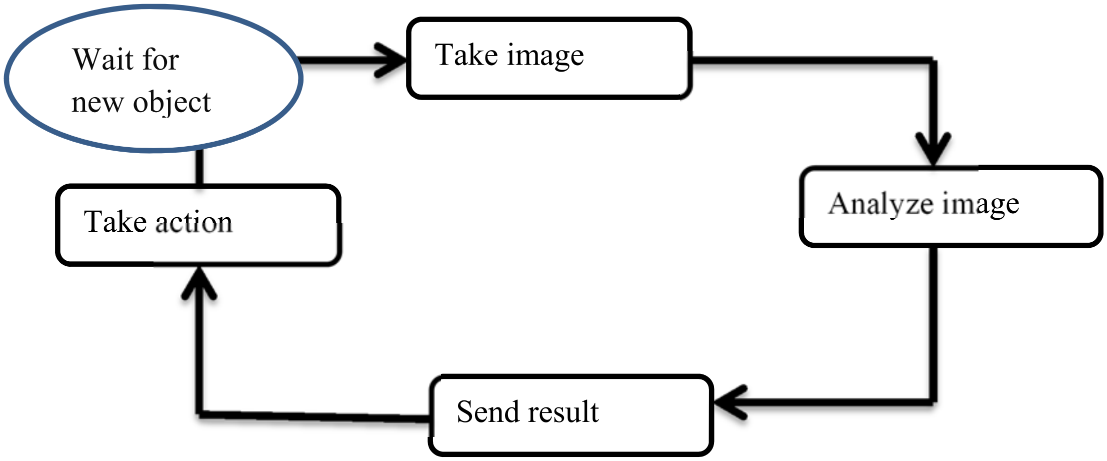


Figure 2.1: A typical vision system operating in four-step cycles

2.2.1 Reasons of using Computer Vision (CV)

In situations where simpler technologies are inadequate, computer vision is employed to solve inspection and measuring jobs. The following are key reasons for investing CV technology in food business:

i). *Reduce labour:* Monotonous manual chores can be avoided, and operating costs can be cut. A vision system, for example, can perform high-speed, uncomplicated jobs so that the operator may focus on activities that need complex judgements (Lind & Murhed, 2012).

ii). *Reduce give-away:* A CV system can help prevent the squandering of costly chemicals. The sprinkling process, for example, can be adjusted in a control feedback loop by counting sesame seeds on each hamburger bun and hence reducing the wastages.

iii). *Increase throughput:* Throughput can be boosted by automating sorting, portioning, and packaging. A vision guided water jet, for example, can portion chicken nuggets at maximum production speed (Sun & Brosnan, 2003).

iv). *Reduce waste.* By automatically detecting faulty objects early in the process, loss can be avoided by recycling or rejecting them as garbage before any value is added. For example, the waste created by pouring chocolate over a damaged wafer can be eliminated, which would otherwise be discarded later in the process at a higher cost.

v). *Improve quality:* Food safety liability claims can be avoided and consequently brand reputation can be enhanced, and consumer complaints can be reduced if quality is improved. For example, computer vision technology may be used to ensure that the cap on juice bottles is properly tightened, improving hygiene standards.

2.2.2 Successful investment in computer vision

Because the vision system is more complex than basic sensors, the following financial considerations are critical when deciding whether or not to invest:

- How long does it take to get your money back?
- In the end, what will the entire return on investment be?
- How long will the product last and how much will it cost to maintain?

When each factor is quantified, it is typically possible to determine whether or not the expenditure is justified. Acceptable pay-back times in the food industry typically range from a few weeks to a year (Lind & Murhed, 2012).

Computer vision systems are often sold with the notion that they can accomplish anything and that, once installed, they will work smoothly and require no maintenance. Practical experience has shown that these expectations are often incorrect. As a result, recognizing the technology's limitations is equally as vital as understanding its possibilities. Although judging each specific scenario requires experience, the following elements should be addressed in general when purchasing a vision system from an outside source:

- *Requirements and scope:* Is it possible to document and agree on the application's requirements? Is it required to incorporate all of the scope's criteria, or are they included merely because they are possible? Cost and complexity can be decreased by selecting where to draw the line when requirements are ordered in order of importance.

- *Testability and acceptance:* It's frequently a lot easier to state a criteria than it is to verify that it's been accomplished. Are the specifications written in such a way that they can be tested? Unless the consultation is paid for by the hour, it is in the best interests of both the vendor and the buyer to include an acceptance test specification in the business model. This will need a little more effort from both parties in the beginning, but it will pay off in the long run.
- *Degree of innovation and risk:* How new are the inspection concepts that must be established, especially when viewed through the lens of the integration resource's competence? Risk management should be considered, for example, by implementing a modest proof-of-concept prototype and using a project model with milestones.
- *Operating expenses:* Investing in computer vision is frequently more than a one-time cost. To run and maintain the vision system, what skills and resources are required? These considerations should be factored into the overall financial planning and structure (Lind & Murhed, 2012).

2.3 Basic techniques and their application

The fundamental concepts and approaches are defined first, followed by a discussion of what may be accomplished and how CV (Computer Vision) technology can be implemented.

2.3.1 Texture analysis

The surface of an object is considered to have texture when it is rough, braided, or made up of many microscopic particles or strands. In such conditions, they form various intensity patterns with varying degrees of regularity and randomness. For more regular patterns, periodicity will be detectable, but it will vary significantly throughout the surface; they will also exhibit high directionality, which will manifest in multiple directions – similar to the weave of a fabric. A mound of sand or seeds will have a far more random texture than a fabric, and it will not be directed (Davies, Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th edition,, 2012).

Because most textures have a certain amount of randomness, quantifying texture patterns poses statistical challenges, as a result, it's critical to average over a broad enough area to provide a good assessment of the surface texture. This is required for object recognition and delineation of object boundaries, as well as the detection of any flaws, defects, or foreign bodies (Sun & Brosnan, 2003). Because each pixel must be visited multiple times in order to test textural coherence over diverse distances and orientations, achieving a unique texture characterization could take a long time and effort.

2.3.2 Thresholding and feature detection

A simple technique to perform image processing is to take the pixel value at each pixel position and place a modified value at the corresponding point in an output picture space. In the thresholding approach, for example, each pixel whose intensity is darker than a specific threshold receives a binary value of 1, while other pixels receive a value of 0. Such a process is known as a pixel–pixel operation. The more sophisticated 'window–pixel' processes look at the pixel values in a window enclosing each pixel position and compute a value to be placed at that point in the output image space.

Convolutions are one of the most common and commonly utilized window–pixel processes, and for $(2k + 1) \times (2k + 1)$ windows, these are defined by the formula:

$$P'[i,j] = \sum_{m=-k}^{k} \sum_{n=-k}^{k} P[i-m, j-n]M[m,n] \tag{2.1}$$

where P is the original image, M is the applied convolution mask and P' is the output image. Local averaging to assist reduce noise, enhancing small holes, enhancing vertical and horizontal edges, and enhancing corners are all examples of this type of convolution. The following convolution masks can be used to perform these actions within 3x3 windows:

Application: locating insects in cereals

In the agri-food business, thresholding and feature detection have a variety of uses; for example, they can be used to detect the presence of insects in cereals.

2.3.3 Thresholding

Thresholding is one of the most prevalent (and fundamental) segmentation techniques in computer vision, allowing us to separate the foreground (i.e., the objects of interest) from the image's background. In instances where there are no transparent images and therefore no alpha channel mask, the image is thresholded to obtain the desired mask. Here's how thresholding works: a threshold value is applied to each pixel. If the pixel value is less than the threshold value T, it is set to zero; otherwise, it is set to the maximum value. We have three types of thresholding:

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix},\frac{1}{8}\begin{bmatrix}-1 & -1 & -1\\-1 & 8 & -1\\-1 & -1 & -1\end{bmatrix},\frac{1}{3}\begin{bmatrix}-1 & 0 & 1\\-1 & 0 & 1\\-1 & 0 & 1\end{bmatrix},\frac{1}{3}\begin{bmatrix}1 & 1 & 1\\0 & 0 & 0\\-1 & -1 & -1\end{bmatrix}$$

$$,\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix},\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix},\frac{1}{20}\begin{bmatrix}-5 & 4 & 4\\-5 & -5 & 4\\-5 & 4 & 4\end{bmatrix} \tag{2.2}$$

- Simple thresholding: manually provide image segmentation parameters. This method works exceptionally well in controlled lighting conditions. Ensure a high contrast between the image's foreground and background.
- Otsu's thresholding is more dynamic and computes automatically the optimal threshold value based on the input image.
- Adaptive Thresholding: divides the image into smaller pieces and thresholds each of these pieces independently.

2.3.3.1 Simple Thresholding

Thresholding is the process of binarizing an image. In general, we aim to convert a grayscale image into a binary image in which pixels are either 0 or 255. A simple thresholding example would involve selecting a threshold value T and setting all pixel intensities below T to zero and intensities above T to 255. We are able to create a binary representation of the image in this manner.

This procedure requires human participation. The threshold value T must be specified. All intensities of pixels below T are set to 255. And all pixel intensities greater than T are set to 0 — this is known as thresholding in reverse. The majority of the time, you want your segmented objects to appear white on a black background, so we employ inverse thresholding. We could also reverse this binarization by setting all pixels with intensities greater than T to 255 and those with intensities less than T to 0.

Simple Thresholding is effective for simple images under controlled lighting conditions, in which it may be possible to hardcode the Threshold value T. Since we must manually provide a thresholding value, there is no assurance that this threshold value T will work from one image to the next when lighting conditions change. The solution is to utilize techniques such as Otsu's method and adaptive thresholding to achieve better results.

2.3.3.2 Otsu's method

In actual conditions where we have no prior knowledge of the lighting conditions, Otsu's method (Bangare, Dubal, Bangare, & Patil, 2015) is used to automatically determine the optimal value of T. Assumption – The method of Otsu assumes that our image contains two classes of pixels: background and foreground. In addition, the Otsu method assumes that the grayscale histogram of our image's pixel intensities is bimodal, which simply means that the histogram has two peaks.

Otsu's method computes an optimal threshold value T based on the grayscale histogram such that the variance between the background and foreground peaks is minimal. Importantly, Otsu's method is an example of global thresholding, meaning that a single T value is calculated for the entire image. In some instances, having a single T value for an entire image is acceptable; however, in others, this can result in subpar results.

We can save a great deal of time using Otsu's method to estimate and verify the optimal value of T (Sindhuri & Nallapareddy, 2022). The issue with Otsu's method is that it assumes our input image's grayscale pixel intensities have a bimodal distribution. If the grayscale image does not have a bimodal distribution, Otsu's method will still function, but it may not produce the desired outcomes. When the lighting conditions are not uniform for example, when different parts of the image are illuminated more than others we may encounter significant difficulties. And in such a case, we will have to employ adaptive thresholding.

2.3.3.3 Adaptive thresholding

In situations where the image's illumination is not uniform, having only a single value of T can severely impair our thresholding performance. Having only one value of T may not be sufficient. To solve this issue, we can employ adaptive thresholding, which considers pixels' small neighbors and then determines the optimal threshold value T for each neighbor. This method allows us to handle instances in which pixel intensities may vary dramatically and the optimal value of T may vary across the image. The objective of adaptive thresholding, also known as local thresholding, is to statistically examine the pixel intensity values surrounding a given pixel p. All adaptive and local thresholding methods are based on the general premise that smaller regions of an image are more likely to have approximately uniform illumination. This implies that the lighting conditions in adjacent regions of an image will be comparable.

Choosing the neighborhood size for local thresholding is absolutely essential. The surrounding area must be large enough to encompass sufficient background and foreground pixels; otherwise, the value of T will be largely irrelevant. Nevertheless, if the neighborhood value is set too high, the assumption that local regions of an image will have approximately uniform illumination is completely violated. Adaptive thresholding's objective is to statistically examine local regions of our image and determine an optimal value of T for each region, which begs the question: which statistic do we use to compute the threshold value T for each region? The arithmetic or Gaussian mean. Adaptive thresholding typically yields good results but is more computationally intensive than Otsu's method or simple thresholding; however, in instances where illumination conditions are not uniform, adaptive thresholding is a very useful tool.

2.3.4 Shape analysis

Shape analysis is one of the most significant methods for people to distinguish and recognize objects, and it is just as vital for computers. Any input photos are believed to have been thresholded to produce binary images with the objects appearing as 1s against a background of 0s. At this point, the computer must figure out how many objects are in the image and which regions they cover. A human seeing the image would see all of this as "obvious," but the computer must figure it out using "connected components analysis." This is not an easy technique for the following reasons. When using a forward raster scan on an analogue TV screen, one object appears first, then another, and then the first one again, so labeling the pixels in order will not offer separate labels for the objects. While it's true that combining scans (for example, forward raster scans followed by reverse raster scans) can aid in the creation of unique labeling, it's more reliable and efficient to review the initial set of labels and create a "clash table" that shows which labels co-exist. After that, a thorough iterative study of the collision table will reveal how to accomplish optimum labeling. After that, the objects must be enumerated, and their areas, circumferences, and linear measures can be easily tabulated. Clearly, connected components analysis allows for numerous more size and form measurements to be taken and recorded.

2.3.5 Object location based on point features

When the objects being sought are characterized by sets of point features such as corners or holes, a separate approach known as graph matching is typically used (Davies, Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th edition,, 2012). In such circumstances, subsets of the observed point features must be searched for that exactly match subsets in the object template. The solution with the largest subset of precise matches is rated the most plausible. It can be seen that this is a voting scheme in which the best option is the one that receives the most votes. Unfortunately, rigidly following this concept results in a computing load that grows exponentially with the number of features to match. Because the concept of a subset is so broad, all subsets of all sizes and compositions must be tested. Fortunately, using a specific variant of the Hough transform, this problem may be easily solved for 2-D interpretation (Davies, Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th edition,, 2012).

2.3.6 Robust object location

A radical new strategy is required to achieve a high level of robustness: it becomes necessary to infer the presence of objects from partial information. The Hough transform method does this using a voting process that counts only positive evidence for objects. The recognition of circular objects is the most basic example of this. All edge points in the picture are made to vote for a circle center by traversing a distance equal to the presumed radius along the edge normal and casting a vote in a separate image space termed a parameter space. If only p pixels are recognized on the circle's edge, p votes will be cast at the circle's center, resulting in a peak in the parameter space. As a result, even if most of the circle's boundary is misplaced, the circle may be reliably located (Davies, Design of cost-effective systems for the inspection of certain food products during manufacture, 1984). Furthermore, the problem of an unknown circle radius can be solved in a variety of ways, for as by trying a series of radius values. This approach may be used to identify straight lines and ellipses, as well as other shapes made from of these shapes. It can also be used to find arbitrary shapes if they are described appropriately. As previously stated, the effectiveness of the technique is contingent on the accumulation of good evidence; false material must be removed because it will simply skew the results.

2.3.7 Morphological analysis

While morphology and shape analysis are fundamentally the same, they evolved from a distinct approach: the use of simple window operations to modify shapes so that they might be filtered to locate certain structures. Eroding objects to remove the outermost layer of pixels or dilation of objects to add an additional boundary layer were the beginning points (Sun & Brosnan, 2003): both methods may be employed isotropically or directionally – for example, parallel to the image's x-axis. The aim of these activities becomes clear when a succession of vertical striations on a smooth surface must be located. The striations would be erased by horizontally eroding the image, and then dilation would return the image to its same condition, but without the striations. By subtracting the final image from the original image, the striations that had been erased would be visible. For example, a technique like this would reveal any scratches on a computer disc or a polished metal surface.

2.4 Colour Moments

Colour moments are measures that can be used to distinguish images based on their colour characteristics (Noah, 2005). Once calculated, these moments provide a colour similarity measurement between images. These similarity values can then be compared to the values of images indexed in a database for image retrieval tasks. The assumption underlying colour moments is that the colour distribution in an image can be interpreted as a probability distribution. Probability distributions are distinguished by a number of distinctive moments (e.g. Normal distributions are differentiated by their mean and variance). If the colour in an image follows a particular probability distribution, then the moments of that distribution can be used as features to identify the image based on its colour.

Researchers (Noah, 2005) have utilized three central moments of the colour distribution of an image. Mean, standard deviation, and skewness are the three. A colour is defined by at least three values. Here, we will limit ourselves to the hue, saturation, and brightness (HSV) colour scheme (Diwash, Ankit, P., & S., 2022). Moments are computed for each of these image channels. Consequently, nine moments define an image. Three moments for each of the three colour channels. The ith colour channel at the jth image pixel will be denoted as pij. The three colour moments can be described as follows:

MOMENT 1 – Mean: Mean can be understood as the image's average colour value.

$$E_i = \sum_{N}^{j=1} \frac{1}{N} p_{ij} \tag{2.3}$$

MOMENT 2 Standard Deviation: The square root of the distribution's variance.

$$\sigma_i = \sqrt{\left(\frac{1}{N}\sum_{N}^{j=1}\left(p_{ij} - E_i\right)^2\right)} \tag{2.4}$$

MOMENT 3 – Skewness: The skewness of a distribution is a measurement of its degree of asymmetry.

$$S_i = \sqrt[3]{\left(\frac{1}{N}\sum_{N}^{j=1}\left(p_{ij} - E_i\right)^3\right)} \tag{2.5}$$

A function of the similarity between two image distributions is defined as the sum of the moments' weighted differences. Specifically, it is:

$$d_{mom}(H,I) = \sum_{i=1}^{r} w_{i1}\left|E_i^1 - E_i^2\right| + w_{i2}\left|\sigma_i^1 - \sigma_i^2\right| + w_{i3}\left|S_i^1 - S_i^2\right| \tag{2.6}$$

where:

(H, I) compares two image distributions, i represents the current channel index (1 = H, 2 = S, 3 = V, etc.), r is the quantity of channels (e.g. 3), E_i^1, E_i^2 represents the initial moments (mean) of the two image distributions, σ_i^1, σ_i^2 are the standard deviations (std) of the images' second moments, s_i^1, s_i^2 are the third moments (skewness) of the distributions of the two images, W_i is the weights for every second.

2.5 Data-Driven Control (DDC)

2.5.1 The growth of control theory

Throughout the history of system control, control theory has evolved from model-free tuning methods, such as PID control, to Model Based Control (MBC) theories, such as transfer function model-based classical control and state space model-based modern control, and finally to knowledge or rule-based intelligent control. This pattern of development can be pictured as a helix that flows from model-free to model-based to deviation from. The Data Driven Control (DDC) is logically the next thing we can imagine.

Based on the validity of control theory, the existing control methods can be categorized into three groups:

a). Control methods designed based on the model of the system, including aerospace control, optimal control, linear and nonlinear control, large-scale system decomposition and coordination control, and pole placement.

b) Control methods, such as robust control, sliding-mode variable-structure control, adaptive control, fuzzy control, expert systems, neural network control, and intelligent control that are designed partially based on the system model.

(c) Control methods based on system I/O data, such as PID and MFAC control.

DDC enhances control theory's validity. In MBC theory, the problems of unmodeled dynamics and robustness are inevitable from the perspective of control theory research. This can result in unsafe controllers and enormous gaps between theoretical results and applications, thereby impeding the development of MBC theory. Modeling more precise high-order and complex nonlinear dynamic systems may result in yet another paradox. Controlled systems with a high order and a high degree of nonlinearity inherit controllers with a high order and a high degree of nonlinearity. These controllers are challenging to develop, operate, and maintain. Typically, either model reduction or controller reduction is required to reduce the control system's complexity. The DDC theory could be an alternative strategy for addressing these paradoxes. Many industrial processes prioritize low-cost, simple-to-install control techniques and automation equipment from the perspective of practical applications.

However, modeling a plant requires particular abilities and mathematical procedures. This type of work is beyond the capabilities of the majority of engineers, so high-level experts or researchers are required. Using the batch process as an example, modeling all batches for all products is impossible. A global model cannot be constructed for complex systems due to internal complexities and external disturbances. Even locally accurate modeling is difficult. Occasionally it is impossible. MBC theory is rarely applicable to industrial processes. In complex system control and management, large volumes of data and a paucity of knowledge are typical obstacles. The inability of the majority of control engineers in the majority of fields to deal with complex mathematics and identification theory is yet another barrier to the application of MBC theory. DDC theory and techniques are demanded by real-world needs.

2.5.2 Introduction to DDC

A model of the controlled system is frequently used to begin control design. The use of PID control and model-free adaptive control techniques are notable exceptions. It may be able to write down a model for a moderate-dimensional mechanical system (e.g., using the Lagrangian, Hamiltonian, or Newtonian formalism) and linearize the dynamics around a periodic orbit or fixed point. However, for modern systems of interest, such as finance, epidemiology, turbulence, climate, and neuroscience, there are often no simple models suited for control design.

Linear control theory provides ways for finding control oriented reduced order models for high-dimensional systems from data, however these techniques are confined to systems. Practical systems are frequently nonlinear, and linear approaches are ineffective for achieving the control goal. Nonlinear control remains an optimization problem with a non-convex, high-dimensional cost function landscape with several local minima. Machine learning is complimentary because it encompasses a growing collection of approaches that can be broadly characterized as nonlinear optimization in a high-dimensional space using data (Gill, Khalaf, Alotaibi, Alghamdi, & Alassery, 2022). Using the rising availability of high-quality measurement data, developing strategies that apply machine learning to regulate and characterize highly high-dimensional, multi-scale, and nonlinear systems were detailed in Section 2.3. Machine learning techniques can be used in a variety of situations. The below figure 2.2 show data driven approaches:

a) Model-based control can be used to characterize a system for subsequent usage,

b) Define a control law that interacts with a system in a direct manner.

The SVM model was used in the research to look at how machine learning may be used to directly identify controllers from input-output data and to discover nonlinear input-output models for control. Iterative learning control and reinforcement learning are two powerful strategies in this rapidly emerging subject.

The figure 2.2 show three ways data driven modeling approaches are used in control systems design and implementation:

1. Data Driven Model
2. Learn Controller
3. Sensor/Actuator placement

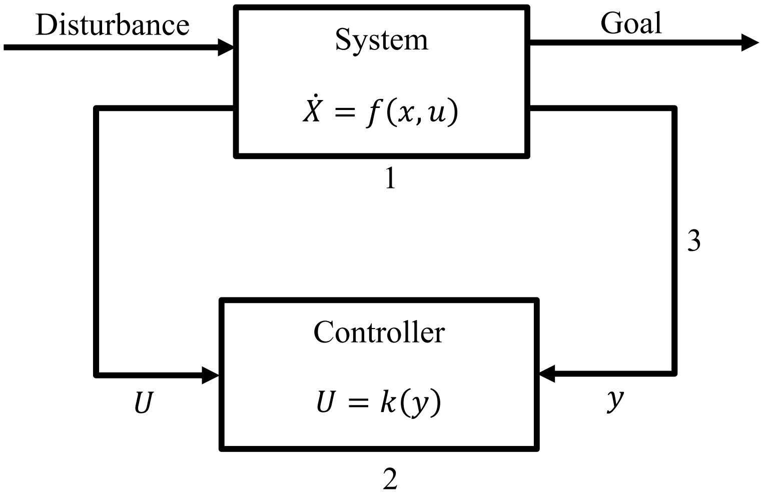


Figure 2.2: Data driven approaches

2.5.3 Model based control theory

Modern control theory, also known as model-based control, was born with the introduction of the parametric state-space model by Kalman in 1960, in conjunction with optimal control (MBC) (Hou & Wang, From model-based control to data-driven control: Survey, classification and perspective, 2013). There (Tang & Daoutidis, 2022) were numerous successful applications, particularly in the aerospace industry where precise models were available. Modern control theory includes both linear and nonlinear control theory. Typical methods for designing linear control systems include zero-pole assignment, LQR design, and robust control. Typical controller design methods for nonlinear systems include Lyapunov-based controller designs, back stepping controller designs, feedback linearization, etc. These methods of controller design are regarded as typical MBC system design.

In applications of MBC theory, the first step is modeling the plant or identifying the plant model, followed by the design of the controller based on the plant model obtained by applying the certainty equivalence principle in the belief that the plant model represents the actual system. MBC theory therefore requires the modeling and identification of the plant.

In order to model a plant using first principles, the parameters must be calibrated on-line or offline with measured data. Identification theory can be used to create a plant model within a set of models that either encompasses the actual system or approximates it in terms of bias and variance error on the identified model. Whether based on first principles or identification from data, modeling is an approximation of the real system, and error is inevitable. There are always unmodeled dynamics in the modeling process. Due to these unmodeled dynamics, the closed loop control system, which is based on supposedly unchangeable MBC approaches, is inherently less safe and less robust. To preserve the obvious benefits of MBC design while enhancing its robustness against model errors, significant effort has been devoted to the development of robust control theory.

A number of methods for describing model errors in the configuration of closed loop systems have been analyzed. These include additive and multiplicative descriptions and a priori noise or modeling error or uncertainty bounds. However, the model uncertainty descriptions upon which robust control design methods are based do not align with the methods delivered by physical mathematical modeling and identification modeling. Modeling by first principles and identification from data offer little in the way of explicit error quantification. Lack of adequate, practical uncertainty descriptions is the most significant obstacle to the application of model-based robust control design techniques. Obtaining a highly accurate model (including a model uncertainty set) for an unknown system through mechanism modeling or identification techniques, and then deriving a model-based robust controller from this model and its uncertainty set, is quite natural. To establish the ideal control theory, researchers must overcome both practical and theoretical obstacles. First, unmodeled dynamics and robustness are twin-born problems that cannot be solved simultaneously within the conventional MBC framework.

Second, the design of the control system requires more effort or expense the more precise the model. Up until now, there was no efficient way to create an accurate model of a plant. Control system design can be more challenging than accurate modeling. In addition, there are no widely accepted methods for addressing specific types of complexity, such as that observed in plants whose parameters vary rapidly or whose structures change over time.If the system dynamics are too complex, we cannot use it as a design model for a control system. Even if utilized as a design model for control systems, this would typically result in a controller with an excessively high order. In practice, high-order controllers are inapplicable, necessitating a reduction in model or controller order. Modeling an accurate high-order model to achieve high performance in a control system design, and then reducing the controller order or simplifying the model for a low order controller, seems paradoxical. Last but not least is the modeling condition of persistent excitation or persistently exciting inputs. Without the persistently exciting inputs, it is impossible to create an accurate model. Most model-based theoretical results of a closed loop control system scheme, such as stability and convergence, cannot be guaranteed when applied in practice without an accurate model.

2.5.4 Data-driven control theory and related topics

Chemical industry, metallurgy, machinery, electronics, electricity, transportation, and logistics processes have undergone significant changes as a result of the advancement of information science and technology. These industries have large-scale production technologies and equipment, and their production processes have become more intricate. Using first principles or identification to model processes has become more difficult. For this reason, traditional MBC theory is no longer applicable to control issues in organizations of this type. Moreover, numerous industrial processes generate and store vast quantities of process data at each and every second of each and every day, containing all the valuable state information of process operations and equipments. Using these data, both on-line and off-line, to directly design controllers, predict and assess system states, evaluate performance, make decisions, or even diagnose faults would be extremely important, particularly in the absence of accurate process models. For this reason, the theoretical and practical establishment and advancement of data-driven control theory (DDC) are urgent concerns (Tang & Daoutidis, 2022).

The term "data-driven" originated in computer science and has only recently entered the control community's lexicon. There have been a few DDC methods in the past, but they have different names, such as data-driven control, data-based control, modeless control, (Tang & Daoutidis, 2022)MFAC (model-free adaptive control), IFT (iterative feedback tuning), VRFT (virtual reference feedback tuning), and ILC (integrated linear control) (iterative learning control) (Hou & Wang, Information Sciences-2013 From model-based control to data-driven control., 2016). There are distinct distinctions between the terms data-driven control and data-based control. Data-driven control implies that the process is a closed loop control in which the starting point and final destination are both data, whereas data-based control indicates that the process is an open loop control in which the starting point is the only data-driven variable.

2.5.5 Control objects of Data Driven Control (DDC)

Data-driven control encompasses all control theories and methods in which the controller is designed by directly using on-line or off-line I/O data of the controlled system or knowledge from data processing, rather than explicit information from a mathematical model of the controlled process, and whose stability, convergence, and robustness are guaranteed by rigorous mathematical analysis under certain reasonable assumptions.

This definition emphasizes three key points. They include the direct use of measurement I/O data, data modeling as opposed to modeling from first principles or identified modeling, and the assurance of theoretical analysis results (Hou & Wang, From model-based control to data-driven control: Survey, classification and perspective, 2013). Simply put, it is a type of method that connects data to controller input.

The two primary components of a control system are the controlled object and the controller. Real-world controlled plants can be classified into four categories: First, those for which there are precise mathematical models derived from first principles or identification. Secondly, those for which first principles or identification-based mathematical models are approximatively precise with moderate uncertainty. Thirdly, those for which first principles or identification-based mathematical models are overly complex in terms of order and nonlinearity. Finally, those for which establishing first principles or identification-based mathematical models is difficult or impossible.

2.5.6 Fundamental differences between MBC and DDC approaches

MBC control and DDC control are the two components of control theory, and their ultimate goal should be the same, i.e., to design a controller that drives the output of the controlled plant to track the desired signal or meet the target (Hou & Wang, From model-based control to data-driven control: Survey, classification and perspective, 2013). The primary distinction between MBC and DDC is that one is a model-based control system design approach because a reliable mathematical model is available, whereas the other is a data-driven control system design approach because there is no such model. Due to this primary distinction, the DDC has many inherent characteristics:

In DDC approaches, the controller does not explicitly include any portion or the entire plant model. It has therefore eliminated the dependence on the plant model for the design of control systems. The stability and convergence conclusions of DDC approaches do not depend on the accuracy of the model, with the exception of the DDC control methods that implicitly use system dynamics and structure information, such as direct adaptive control, sub-space predictive control, etc., which is the primary obstacle for the applications of the MBC theory. The most remarkable aspect of DDC approaches is that the twinborn problem of un-modeled dynamics and robustness in conventional MBC theory does not exist within the DDC framework. The primary distinction between MBC and DDC is whether the controller is designed based on the system model or I/O data alone, or, alternatively, whether the system dynamic model is utilized in the controller design (Hou & Wang, From model-based control to data-driven control: Survey, classification and perspective, 2013). If the system model is involved in the controller, the method is MBC; otherwise, it is DDC. From this perspective, we can conclude that some neural-network-based control methods, fuzzy control methods, and many other intelligent control methods are DDC methods, such as the NN-based control methods in which the neural network serves as a controller directly approximating the system's inverse. Some are not, in which the neural network, fuzzy rule, or knowledge describing the system functions as a system model, and the neural network, fuzzy rule, or knowledge is involved in the controller.

2.6 Support Vector Machine (SVM)

The SVM is one of the most effective classification algorithms that has demonstrated current effectiveness in a variety of classification applications (Xu, Li, Li, & Deng, 2013). SVM is a classification algorithm that may be used on both linear and nonlinear data (Jiawei & Kamber,, 2006). SVM uses kernel functions to nonlinearly map information to a high-dimensional space (Mathar, Alirezaei, Balda, & Behboodi, 2020). Then next, in that high-dimensional space SVM tries to find the linear optimal hyper plane that separates information with maximum margin. Originally SVM was put forward for only 2-class problems, but for multi-class problem can be extended SVM using near-against-one or one-against-all strategies (Gosselin, Kleynen, Leemans, Destain, & Unay, 2010).

The LS-SVM solves a series of linear equations rather than a quadratic programming problem. LS-SVM was used for the automatic detection of browning degree on mango fruits in (Zheng & Lu, 2012). Suitable kernel function and optimal kernel parameters are of importance in LS-SVM classifier and hence, RBF kernel was applied as the kernel function due to its success and speed in training process.

Support Vector Machines (SVMs) are advanced classification technique based on Machine Learning (ML) theory (Patil, Shelokar, Jayaraman, & Kulkarni, Regression models using pattern search assisted least square support vector machines, 2005). SVMs have considerable advantages over other methods such as artificial neural networks, decision trees and Bayesian networks due of their exquisite mathematical tractability, direct geometric interpretation and high accuracy.

Further, they do not need a big number of samples for training to avoid over fitting (Li, Yang, & Wang, 2010). SVM makes the give and take between model complexity and generalization in order to realize the best generalization.

Table 2.3 Advantages and Disadvantages of Machine Learning Techniques
(Rashmi, Sapan, & Roma M., 2013)

Machine learning methods	Advantages	Disadvantages
LDA (Linear Discriminant Analysis) **(Gagan, 2022) (Jiménez Carvelo, Cruz, Cuadros-Rodríguez, & Koidis, 2022)**	LDA are suitable for the development of linear classification models.	i). Assumes Gaussian distribution of data. ii)..data over fitting problem

KNN (K- Nearest Neighbor) (Jiménez Carvelo, Cruz, Cuadros-Rodríguez, & Koidis, 2022)	i). Limited parameters to tune (distance metric and k) and is robust ii). Simple implementation. iii). Classes don't have to be linearly separable.	i). Sensitive to irrelevant or noisy data. ii). Long testing time because of calculation of distance to all known instances.
ANN (Artificial Neural Network)	i). User friendliness, robust and can handle noisy data. ii). Suitable to analyze complex data.	i). Difficult scalability. ii). Large number training samples. iii). Long processing time
Rule Based System (RBS) fuzzy system	i). Robust. ii). Little memory requirement. iii). Insensitive to the changing environment.	Problem with determination of membership function.
SVM (Support Vector Machines) (Mathar, Alirezaei, Balda, & Behboodi, 2020)	i). High classification accuracy compared to other traditional classification techniques. ii). Very robust, even with distorted training samples. iii). Suited to work with high dimensional data.	i). Difficult selection of kernel function and kernel parameters ii). Time consuming learning process.

Likened to other traditional learning methods, SVM can overcome convectional learning flaws, such as driven to local minimum and less-learning and over-learning.

For the situations when input samples cannot be separated in a linear space, SVM performs a nonlinear transform and modify this inseparable problem into a divisible question in a high dimensioned space and work out its optimal classification surface in this space. Without computation complexity increasing, classification can be done through inner product computation with SVM core function in a high dimensioned space. Table 2.3 enumerates the benefits and drawbacks of machine learning approaches

2.7 Research Gap

The purpose of the literature review was to give an overview of available computer vision methodology for food processing: the key principles are mainly image processing and abstract pattern recognition, which must function together in order to build entire computer vision systems. The techniques discussed above may have one or more of the following weaknesses:

i). Low accuracy and require better preprocessing algorithm

ii). Not best for colour image, due to the high correlation between R, G, and B component,

iii). Slow process and require high computation time,

iv). Methods use single feature extraction like colour, reducing their classification accuracy,

v). The techniques did not extend to sliced fruit classification,

vi). The recognition systems are not robust, require large number training samples, difficult scalability and are sensitive to irrelevant and noisy data.

With this background, the present work aims to advance a computer vision knowledge by developing an algorithm for automating a manual grading of pineapple slices based on colour, shape and texture. The choice of methodology is unique and addresses the challenge of unique feature extraction of almost similar pineapple slices.

Support Vector Machines (SVMs) have considerable advantages over other methods such as artificial neural networks, decision trees and Bayesian networks due of their exquisite mathematical tractability, direct geometric interpretation and high accuracy. Likewise, a hybrid feature extraction technique using texture, colour moment, colour histogram and shape features is better and more effective than using any single feature in classification of fruits. This hybrid technique improves the recognition performance. Colour histograms and colour moment are reasonably invariant with rotation and translation about the viewing axis. To reduce the number of features, PCA is used, by eliminating irrelevant features and preserving over 99% energy. This increases classification accuracy, reduce computation time and accelerate the algorithm remarkably. In addition, three different multiclass SVMs (DAG, MWV & WTA) are used for multiclass classification for performance validation.

CHAPTER THREE

METHODOLOGY

This chapter presents the approach adopted in developing the computer vision algorithm with the proposed method to solve the problem of pineapple slices sorting and grading problem.

3.1 Overview

In this study, a unique method for automating the manual grading of pineapple slices is developed using machine vision and the MWV-SVM with GRB kernel. Various pineapple slices images are taken with a camera at Del Monte cannery packing tables, and then the background of each image is removed using an Otsu's segmentation procedure. To create a feature space, colour moments, texture, colour histogram, and shape features of each pineapple slice image are retrieved. The dimension of the feature space is reduced when PCA is used; lastly, build three types of multiclass SVMs: MWV-SVM, WTA-SVM and DAG-SVM for performance comparison and validation. Also, three kinds of kernels are used: linear kernel, Homogeneous Polynomial kernel and Gaussian Radial Basis kernel. Then used 5-fold stratified cross validation to train the SVMs with the reduced feature space vector as input.

The methodology used is shown in flow chart in figure 3.1 below. The images taken are first preprocessed to remove the background and focus on slice image only. This form a data set of images. There are five categories of pineapple slices, fancy ¾, fancy ½, choice, broken and reject. Each category is made of 50 image, equally not to skew data in any way. Using a unique feature extraction technique developed, features are extracted from each image to form a feature space. Using PCA method, the features are ordered in the priority order and the most important one selected. The classifiers are trained using the reduced features space and performance validated.

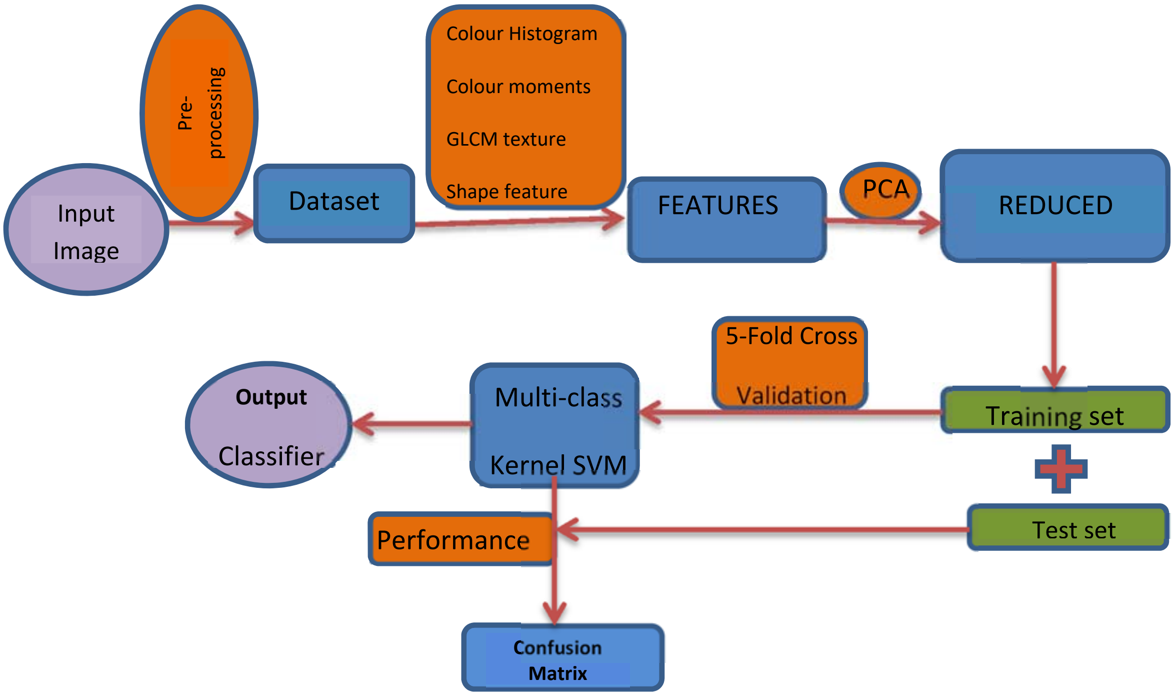


Figure 3.1: Methodology flow diagram

3.2 Image Segmentation by the Otsu's Method

The process of separating foreground and background pixels is known as thresholding. There are numerous methods for achieving optimal thresholding, one of which is called Otsu's method. The method is a variance-based technique for determining the threshold value with the smallest weighted variance between the foreground and background pixels. The key concept here is to iterate through all possible threshold values and measure the distance between pixels in the background and foreground. Then, identify the threshold with the smallest spread.

3.2.1 Otsu's method implementation

The algorithm seeks iteratively the threshold that minimizes the within-class variance, which is defined as the weighted sum of the variances of the two classes (background and foreground). Typically, grayscale colours range between 0 and 255. (0-1 in case of float).

Therefore, if we select a threshold of 100, all pixels with values less than 100 will become the image's background, while all pixels with values greater than or equal to 100 will become the image's foreground.

The formula for calculating the within-class variance at any given threshold t is:

$$\sigma^2(t) = \omega_{bg}(t)\sigma_{bg}^2(t) + \omega_{fg}(t)\sigma_{fg}^2(t) \tag{3.1}$$

Where $\omega_{bg}(t)$ and $\omega_{fg}(t)$ represent the probability of the number of pixels for each class at the threshold t and σ^2 represents the colour value variance. To grasp the significance of this probability, let's, P_{all} consist of the total number of pixels in an image, $P_{BG}(t)$ Be the number of background pixels at the threshold value t, $P_{FG}(t)$ Be the number of foreground pixels at the threshold value t.

Thus, the weights are determined by

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}} \tag{3.2}$$

$$\omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}} \tag{3.3}$$

The variance can be computed using the formula below:

$$\sigma^2(t) = \frac{\sum(x_i - \bar{x})^2}{N - 1} \tag{3.4}$$

where

x_i is the pixel value at position I in the group (bg or fg), $\bar{x}$ is the average pixel value within the group (bg or fg), N represents the number of pixels.

3.3 Feature Extraction and Reduction

The research used a hybrid classification algorithm based on colour histogram, colour moment, texture and shape i.e. appearance features of pineapple slices. Here, research supposes the pineapple slice images have been produced by Otsu's segmentation procedure (Juliana, Florencia, & Laura, 2020), (Nnamdi, Vincent, Eneh, Ogechukwu, & Ijeoma, 2022).

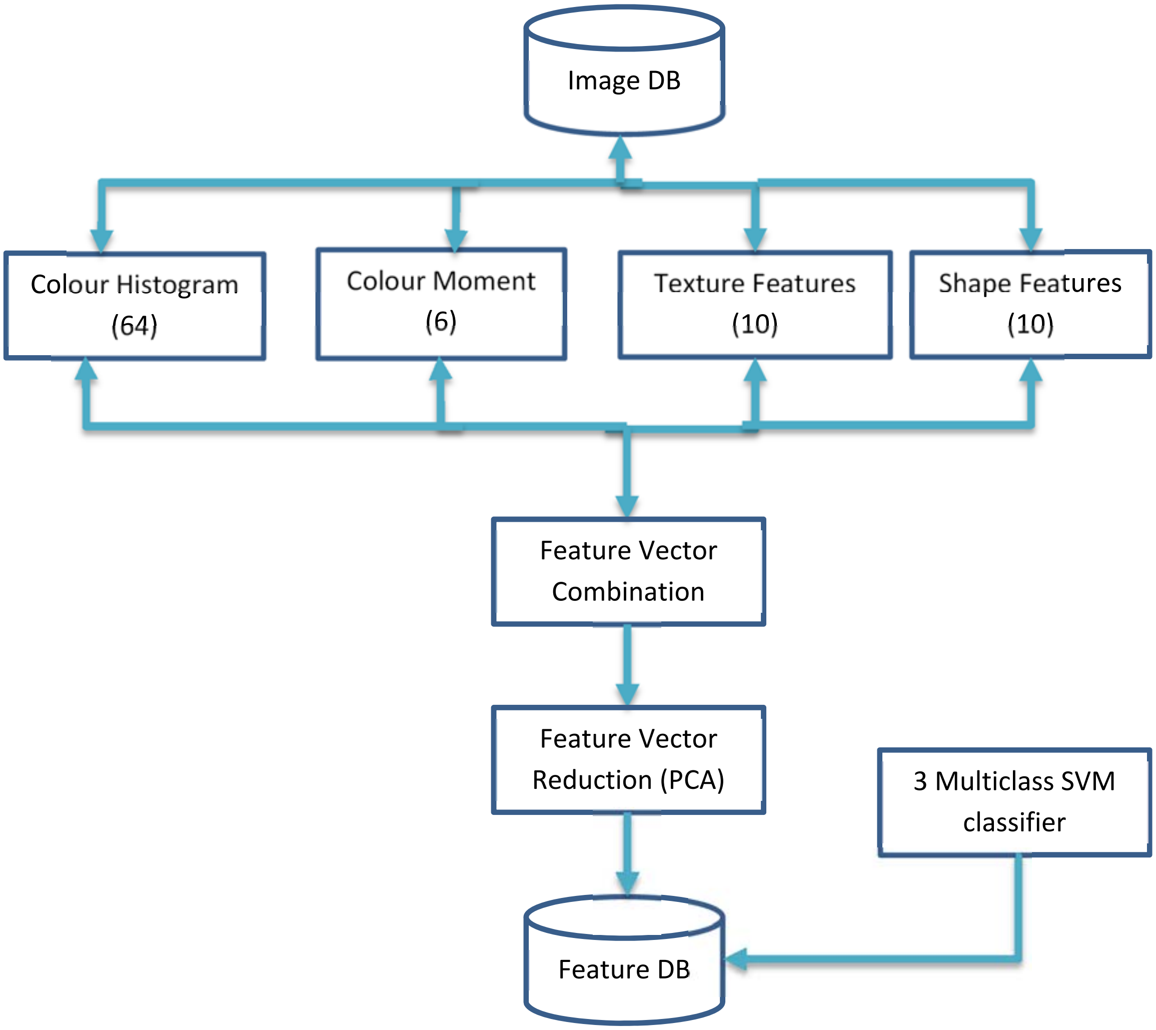


Figure 3.2: Feature Extraction overview

The figure below show the hybrid feature extraction technique employed by the thesis. A total of 90 features are extracted every single image of slice for the entire 250 image dataset to form the feature database. The features are them combined and reduced in dimension using PCA. Three multiclass SVM classifier are then employed for performance comparison.

The figure 3.2 below shows the feature extraction overview. From the image data base, four methods are used to extract the features and form a features vector. Features are then ordered using PCA and eliminate the irrelevant ones. The reduced feature vector is used to train the classifiers. By so doing, the algorithm is accelerated and high accuracy achieved.

3.3.1 Colour Histogram

The colour histogram is currently used to represent the colour distribution in an image (Siang & Mat, 2011). The amount of pixels with colour in a fixed list of colour ranges that span the image's colour space is defined as the colour histogram (Maitra & Chatterjee, 2008).

In the case of monochromatic photos, the set of possible colour values is sufficiently small that each of these colours may be placed on a single range; thus, the histogram is simply defined as the number of pixels that have each conceivable colour. For RGB colour images, the space is divided into a reasonable number of ranges, which are often organized as a regular grid and contain multiple similar colour values.

Pineapple slice image with RGB values from 0 to 255, so it will have a total of $256 \times 256 \times 256 = 2^{24}$ colour. The research chose to use four boxes to represent each colour component, Boxes 0, 1,2, 3 indicate intensities 0 to 63, 64 to 127, 128 to 191, 192 to 255, correspondingly, so there are in total $4 \times 4 \times 4 = 64$ colour. The four bins are chosen as found in literature as good representative reduction and summary.

The histogram offers a compact summary of the distributions of data in an image. An image's colour histogram is fairly rotation and translation invariant around the viewing axis. By matching the colour content of one image with the colour content of the other and equating the histogram signatures of two photos, the colour histogram is ideally suited to the task of recognizing an item of uncertain location and rotation inside a scene.

3.3.2 Colour moments

Colour moments are used to distinguish images based on their colour characteristics. This moment is used to determine the degree of colour similarity between two images. The assumption underlying colour moments is that the colour distribution in an image can be interpreted as a probability distribution. If the colour in an image follows a particular probability distribution, the moments of that distribution can be used to identify the image based on its colour.

Stricker and Orengo (Senthilkumaran & Rajesh, 2009) utilize three central points of the colour distribution of an image. Mean, standard deviation, and skewness are the three. A colour can be characterized by three or more values (Red, Green, and Blue). Moments are computed for each of these image channels. Therefore, an image is comprised of nine moments, three for each of the three colour channels. The i-th colour channel at the j-th image pixel will be denoted as Pij. The three colour moments can then be defined as:

MOMENT 1 - Mean
MOMENT 2 - Standard Deviation
MOMENT 3 – Skewness

In this research the first 6 colour moment were used as follow:

3.3.2.1 Moments of Colour Feature Extraction

Step1: open the image file.

Step 2: We calculate the mean using the following function.

$$E_i = \sum_{j=1}^{N} P_{ij} \tag{3.5}$$

Step 3: The following function yields the Standard Deviation value.

$$\sigma_i = \sqrt{\frac{1}{N}\left(\sum_{j=1}^{N}\left(P_{ij} - E_i\right)^2\right)} \tag{3.6}$$

Step 4: The final step is to store the mean and standard deviation values in a 1D array.

Step 5: Repeat steps 1-3 for each image in the database.

3.3.3 Texture feature

Local binary pattern and Unser's texture are very good texture descriptors, but are complicated to implement. In this research, the Gray level co-occurrence matrix (GLCM) is chosen as it shows better results for simple situations where the textures are visually easily separable. Furthermore, the GLCM algorithm is easy to implement and has been shown to give very good results in a large fields of applications. Texture is an important characteristic for the analysis of many types of images because it provides a rich source of information about the image. Also it provides a key to understand basic mechanisms that underlie human visual perception.

The GLCM is created from a gray-scale image. The GLCM determines how frequently a pixel with gray-level (grayscale intensity or Tone) value i appears horizontally, vertically, or diagonally to adjacent pixels with value j.

Several texture metrics that contain spatial information are based on the co-occurrence matrix, they also known as the spatial gray-level dependence matrix. Forming the co-occurrence matrices is an initial step that compiles spatial as well as statistical information for computing the texture metrics. The spatial information considered is the relative position of pairs of pixels, defined with distance d and orientation θ that describe the second pixel's position in relation to the first. A co-occurrence matrix is formed for each such position. In this manner, each co-occurrence matrix prepares the data to emphasize primarily structure or streaks in a given direction and a grain size that is at least as large as the selected distance. Typically, four values of θ, namely 0°, 45°, 90°, and 135°, cover the orientations, and the most common choice of distance is $d = 1$ when θ is 0° or 90°, and $d = \sqrt{2}$ when θ is 45° or 135°. The spatial relationships of pixels established by the array of offsets are shown in Figure-3.3, where D indicates the distance from the pixel of interest (Norton, Ozkan, Mert, & Senturk, 2008), (Benazir & Vijayakumar, 2012).

In 1973, Haralick (Haralick, Shanmugam, & Dinstein, 1973) introduced 14 statistical features. In the table 3.1 the 10 Haralick features are highlighted. These features are generated by calculating the features for each one of the co-occurrence matrices obtained by using the directions 0˚, 45˚, 90˚, and 135˚, then averaging these four values. The distance parameter can be selected as one or higher. A vector of these 14 statistical features is used for characterizing the co-occurrence matrix contents (Eleyan & Irel, 2011), only ten of them are defined here. The following notation are used: G is the number of gray levels used, µ is the mean value of P, $\sigma_{x\,,}\sigma_y,\ \mu_x\ and\ \mu_y$ are the standard deviations and means of $P_x and\ P_y$. $P_x(i)$ is the ith entry in the marginal probability matrix attained by summing the rows of $P(i,j)$. The table 3.1 below show Haralick features:

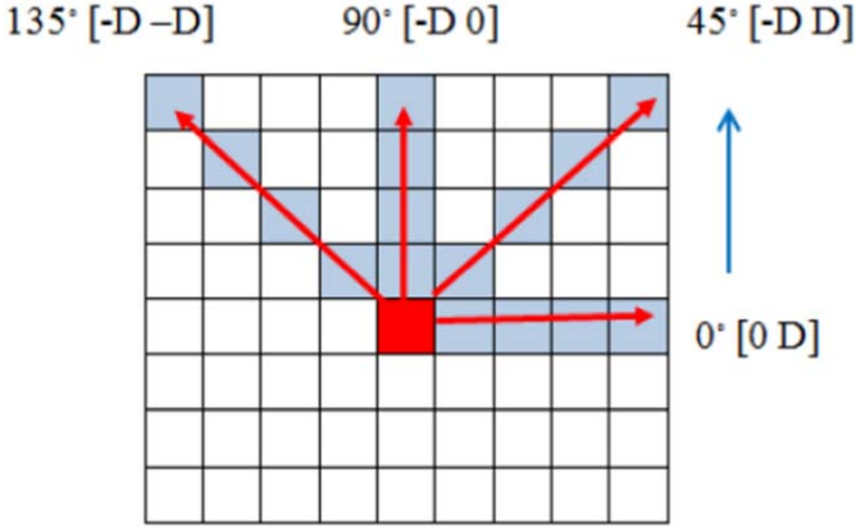


Figure 3.3: Spatial relationships of pixels

Table 3.1: Haralick Features (Haralick, Shanmugam, & Dinstein, 1973).

Measure	Formula
Mean	$\sum_{i=0}^{2G-2} iP_{x+y}(i)$
Contrast	$\sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=1}^{G} \sum_{j=1}^{G} P(i,j) \right\}, \qquad \lvert i-j \rvert = n$
Homogeneity	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i,j)$
Energy	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{P(i,j)\}^2$
Variance	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-\mu)^2 \, P(i,j)$
Correlation	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i \times j\} \times P(i,j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y}$
Entropy	$-\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i,j) \times \log(\, P(i,j))$
RMS	$\mathrm{\mu} = \sum_{i=1}^{G_{max}} \{i.h_i\}$
Kurtosis	$\mathrm{k} = \frac{1}{\sigma^4} \sum_{i=1}^{G_{max}} \{(i-\mu)^4.h_i\} - 3$
Skewness	$\mathrm{S} = \frac{1}{\sigma^3} \sum_{i=1}^{G_{max}} \{(i-\mu)^3.h_i\}$

3.3.4 Shape Features

Table 3.2: The Morphology based Measures (Lou, Jiang, & Scott, 2012).

Measure	Meaning
Area (A_r)	The number of pixels included within the object.
Perimeter (P_r)	surrounding the object boundary distance
Euler (E_l)	The Euler number of an object
Convex (C_n)	The number of pixels in a convex hull
Solidity (S_l)	percentage of area to convex hull
Minor length (M_n)	minor axis length of the ellipse
Major length (M_j)	major axis length of the ellipse
Eccentricity (E_c)	Ellipse eccentricity
Extent (E)	minimum bounding rectangles that are within specified parameters
Orientation (O)	the relative arrangements of points after a transformation

Based on mathematical morphology, ten measures were chosen in this research, which are listed in Table 3.2. There are five groups of measures to consider:

i). The object's area, Euler number, and perimeter can all be calculated directly;

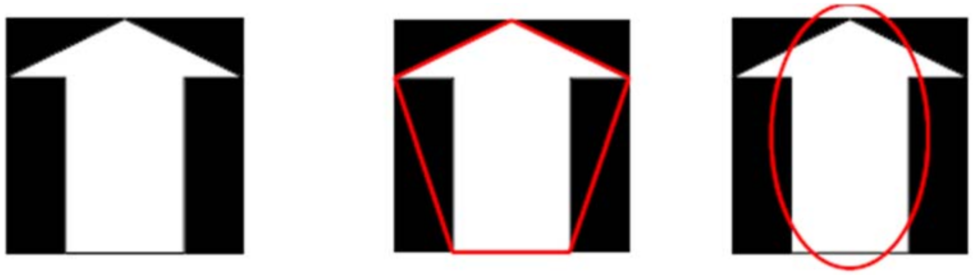
ii). Then, using the Graham Scan method, create a convex hull. Make the smallest convex polygon that covers the item to extract the convex area and solidity properties;

iii). Then, using the same second moments as the object, create an ellipse.

iv). After that, derive the major length, minor length, and eccentricity characteristics.

v). Finally we derive extent and orientation of the image

The figure 3.4 below show the illustration of the morphology measures (Lou, Jiang, & Scott, 2012).



(a) Original Image(s) (b) Convex Hull (c) Ellipse

Figure 3.4: Illustration of the morphology measures.

3.3.5 PCA (Principal Component Analysis)

A total of 90 features (64 colour features, 6 color moment, 10 shape features, and 10 texture features) were retrieved from a single image. Unnecessary features increase the amount of memory used for storage and the amount of time it takes to compute, which occasionally causes the classification process to decrease the classifier performance and even become more complicated (Kwak, 2008). A strategy to reduce the features number used in classification is essential. PCA is a useful approach for reducing the dimensionality of a data set with many interconnected variables while keeping the most essential changes (Kwak, 2008). It is accomplished by converting the data set into a new set of ordered variables based on the relevance or variance of the variables. This method has 3 effects:

i). The input vectors' components are orthogonalized in order to make them uncorrelated with one another,

ii). The orthogonal components that result are then sorted in order of biggest variation to smallest variation,

iii). The data set's components that contribute the least variation are subsequently deleted (Lipovetsky, 2009).

The figure 3.5 shown below shows using normalization before PCA. Before performing PCA, the input vectors are standardized to have unity variance and zero mean, which is shown in Figure 3.5. The process of standardization is a standard statistical procedure. An eigenvalues decomposition of the covariance matrix is used to perform principal component analysis. It can, however, be done using SVD (singular value decomposition) of the data matrix **X** (Jackson, 1991).

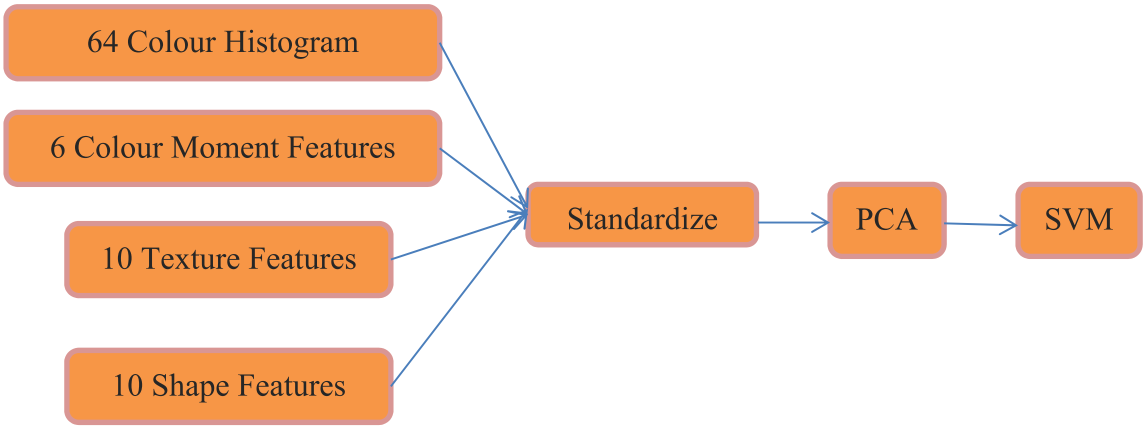


Figure 3.5: Normalization is used before PCA.

3.4 Multiclass Kernel SVMs

3.4.1 Kernel SVM

Convectional linear SVMs cannot separate complex distributed practical data. The kernel method is applied to SVMs so as to generalize it to nonlinear hyper plane (Acevedo, Maldonado, Lafuente, Siegmann, & López, 2009). The resulting method is formally identical since every dot product is replaced by a nonlinear kernel function. The kernel SVMs are used to fit the greatest margin hyperplane in a converted feature space. The transition could be nonlinear and have an altered higher dimensional space. The classifier may be nonlinear in the original input space even if it is a hyperplane in the higher dimensional feature space. Four mostly used kernels (Deris, Zain, & Sallehuddin, 2011) are listed in Table 3.3. For every kernel, there should be at least one adjusting parameter thus to make the kernel flexible and tailor itself to practical data.

SVMs were initially intended for binary classification. Many methods have been proposed for multi-class SVMs, and the prominent approach is to reduce the single multiclass problem into multiple binary classification problems (Maddipati, Nandigam, Kim,, & Venkatasubramanian, 2011). Four widely used types of methods are shown as follow.

Table 3.3: Popular Kernels (Deris, Zain, & Sallehuddin, 2011)**.**

Names	Formula (s)	Parameter (s)
Homogenous Polynomial (HPOL)	$k(x_i, x_j) = (x_i \cdot x_j)^d$	d
Inhomogeneous Polynomial	$k(x_i, x_j) = (x_i \cdot x_j + 1)^d$	d
Gaussian Radial Basis (GRB)	$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$	γ
Hyperbolic Tangent	$k(x_i, x_j) = \tanh(kx_i \cdot x_j + c)$	k, c

3.4.2 Winner-Takes-All (WTA) SVM

Given, there are totally *C* classes; where *C* > 2. A winner takes all strategy is used in the one versus all technique to classify new cases (Lingras & Butz, 2007). It is first trains *c* different binary support vector machine, each one was taught to distinguish data from a single class from data from all the other classes. All of the C classifiers are run on new test data, and the classifier with the highest value is picked. If two yield values are identical, WTA-SVM chooses the class with the smallest index. The following is the mathematical formula . Assume the following N-size *p*-dimensional training dataset:

$$\{(x_n, y_n) | x_n \in R^p, y_n \in \{1,2,\dots,C\}, n = 1,2,\dots,N\} \tag{3.7}$$

where x_n is a ρ-dimensional vector, and $y_n \in \{1,2,\dots,C\}$is the class label of each x_n . The classification function for ith individual binary WTA-SVM (Yudong & Lenan, 2012) can be defined as:

$$f_i(x) = \sum_{n=1}^{N} y_n^i \alpha_n^i \, k(x_n, x) - b_i, i = 1,2, \dots, C \tag{3.8}$$

$$y_n^i = \begin{cases} +1 & if\ x_n \in ith\ class \\ -1 & otherwise \end{cases} \tag{3.9}$$

Where *N* is the number of training data; *C* is the number of total classes; $y_n^i \in \{+1, -1\}$ depends on the class label of x_n, if $x_n$belongs to the *i*th class, $y_n^i = +1$, otherwise $y_n^i = -1$; $k()$is the predefined kernel function; $\alpha_n^i$is the Lagrange coefficient; and $b_i$is the bias term. α_n^i and b_i are obtained by training the *i*-th individual SVM. The *i*-th SVM-output is the sign function of its decision function, namely:

$$O_i(x) = sgn\big(f_i(x)\big) \tag{4.0}$$

If $f_i(x) > 0$, then the output $O_i(x)\ is + 1$, denoting x belongs to *i*-th class; otherwise output is -1, denoting $x$belongs to other classes.

3.4.3 Max Wins Voting (MWV) SVM

For the one vs one technique, a MWV strategy is used for classification. After generating a binary SVM for each pair of classes, you'll end up with C(C-1)/2 binary Support Vector Machines in total (Yudong & Lenan, 2012). When applied to fresh test data, each Support Vector Machine provides the winning class one vote, and the test data is labeled with the class with the most labels. If there are two identical votes, Max-Win-voting selects the class with the smallest index. The following is the mathematical formula. The *ij*-th (*i* = 1,2, …, *C*-1, *j* = *i* + 1, …, *C*) individual binary Support. To identify i-th class from j-th class, Vector Machine is trained with all data in the i-th class with +1 label and all data in the j-th class with 1 label. The decision function of *ij*-th SVM is:

$$f_{ij}(x) = \sum_{n=1}^{N_1+N_j} y_n^{ij} \alpha_n^{ij} \, k\big(x_n^{ij}, x\big) - b_{ij}, i = 1,2, \dots, C-1, j \\ = i+1, i+2, \dots, C \tag{4.1}$$

$$y_n^{ij} = \begin{cases} +1 & x_n^{ij} \in ith\ class \\ -1 & x_n^{ij} \in jth\ class \end{cases} \tag{4.2}$$

Where N_i and N_j represents the total number of i-th class and j-th class, respectively. $y_n^{ij} \in \{+1, -1\}$ depends on the class label of x_n^{ij}. If x_n^{ij} belongs to *i*th class, $y_n^{ij} = +1$; otherwise x_n^{ij} belongs to *j*th class, $y_n^{ij} = -1$. α_n^{ij} is the Lagrange coefficient; and b_{ij} is the bias term. α_n^{ij} and b_{ij} are attained by training the *ij*th individual Support Vector Machines. The output of *ij*th Support Vector Machines is the sign function of its decision function, namely:

$$O_{ij}(x) = sgn\left(f_{ij}(x)\right) \tag{4.3}$$

if $f_{ij}(x) > 0$, then the output $O_{ij}(x)\ is + 1$, denoting x belongs to i-th class; otherwise output is -1, denoting $x$belongs to j-th class.

3.4.4 Directed Acyclic Graph (DAG) SVM

A directed acyclic graph is one in which all of the edges have the same orientation and there are no cycles. Though the two methods create the individual SVM in the same way, the output of each individual Support Vector Machine in DAG (Yudong & Lenan, 2012) is explained differently than that of MWV-SVM. When Oij(x) is -1, it means x does not belong to the ith class, and when it is +1, it means x does not belong to the jth class. As a result, the ultimate decision cannot be made until the leaf node has been reached (Platt, Cristianini, Shawe, & Large, 2000). Figure 3.6 shows the Directed Acyclic Graph SVM for finding the best class out of 5 given classes. Here, the intermediate nodes and root node denotes the individual binary SVM, while the leaf nodes denote the output label. Individual binary Support Vector Machines are evaluated given a test sample *x* starting at the root node. The node is then exited to the right or left edge based on the evaluation outcome. The function of the following SVM is calculated until the leaf node is reached. Consequently, DAG-SVM faster in terms of computation time compared to MWV-SVM. In this case, the DAG-SVM only requires evaluating only four individual SVMs whereas the MWV-SVM requires covering all nodes of 10 individual SVMs, yet. The figure 3.6 below shows: The DAG-SVM for calculating out of five classes the best one.

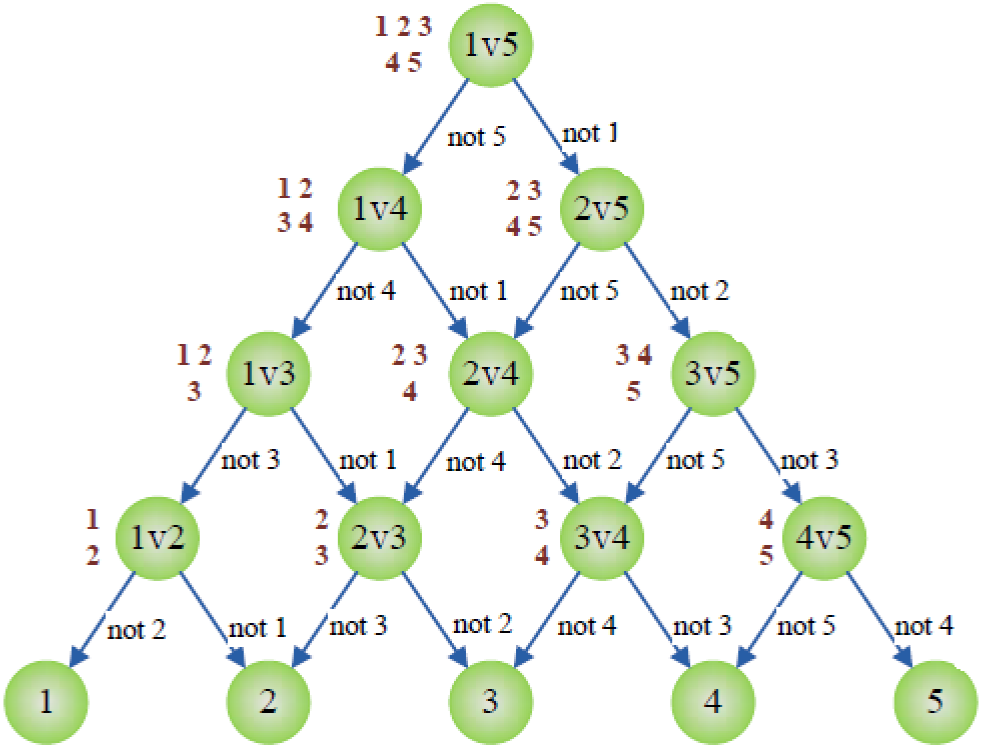


Figure 3.6: The DAG-SVM for calculating out of five classes the best one.

3.5 Stratified cross validation

Normally, a statistical model that works with intrinsic data variability is inferred from the training set database and used automatically by statistical learning machines to build classifiers (Yudong & Lenan, 2012). A model has a set of changeable parameters that are estimated using a set of examples during the learning phase. However, in order to achieve effective generalization, the learning machine must ensure that the parameters are accurately estimated. Generalization refers to the ability to respond correctly to unknown examples, such as correctly classifying new photos. As a result, the learning device must strike an efficient balance between its complexity, which is measured by several variables such as the feature input space dimension and the effective number of free parameters of the classifier, and the problem information provided by the training set, which is measured, for example, by the number of samples. Cross validation methods are frequently used to evaluate the statistical relevance of classifiers. It comprises of four types: *K*-fold cross validation, Random subsampling, Monte Carlo Cross-Validation and leave-one-out validation (Pereira, Reis, Saraiva, & Marques, 2011).

The K-fold cross validation is used for training and validation because of its basic and straightforward features. Making a K-fold partition of the entire dataset, repeating K times using K-1 folds for training and a left fold for validation, and then averaging the error rates of K experiments is the technique (Yudong & Lenan, 2012). The schematic figure below depicts the 5-fold cross validation, figure 3.7:

When *K* folds is purely random partitioned; some folds may have distributions that are significantly different from those of other folds. To avoid this, stratified K-fold cross validation is utilized, in which each fold's class distributions are almost identical (May, Maier, & Dandy, 2010). The mean response value in all of the folds is about equal in this procedure. In the case of a dichotomous classification, the fold contains nearly equal proportions of the two types of class labels.

Another difficulty encountered is determining the amount of folds. If K is set too high, the true error rate estimator's bias will be minor, but the estimator's variance will be big, making calculation time consuming. Alternatively, if K is set too small, the variance of the estimator will be modest, but the bias of the estimator will be considerable, and the calculation time will be reduced (Armand, Watelain, Roux, & Lepoutre, 2007). K was empirically determined to be 5 in this study using the trial-and-error method.

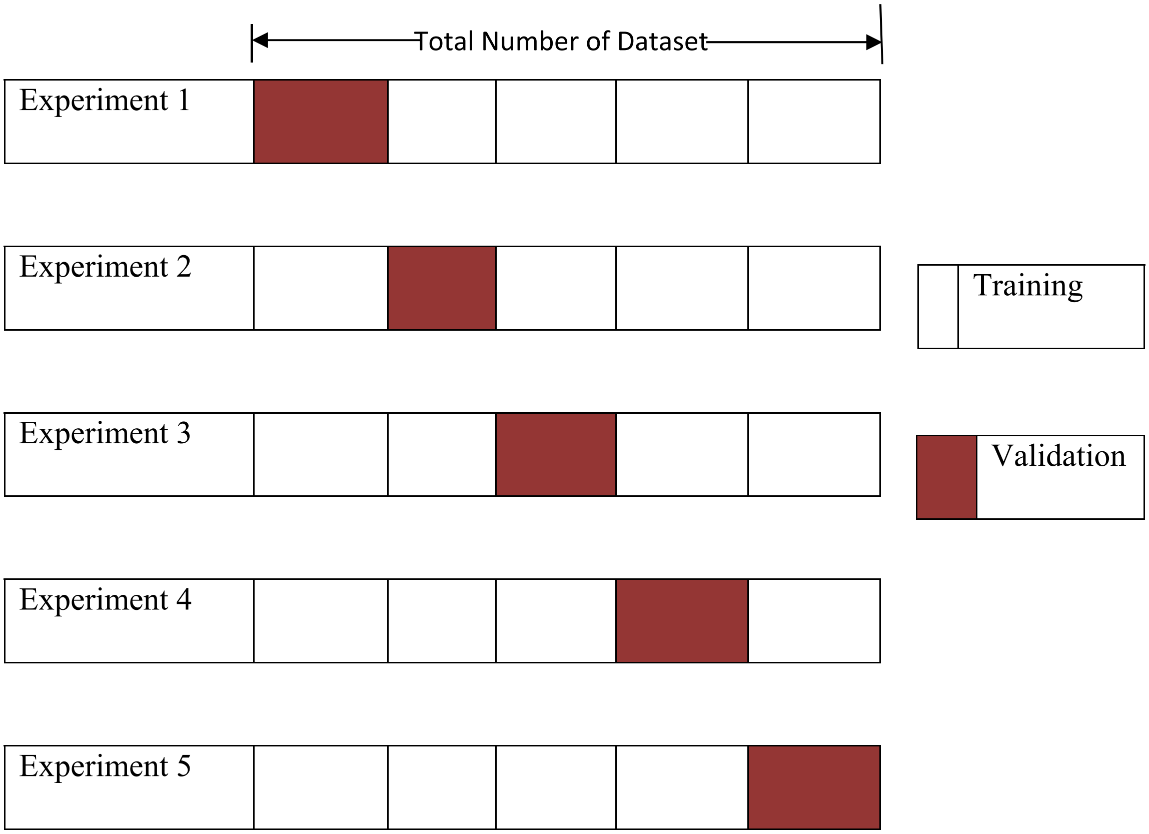


Figure 3.7: A 5-fold complexity cross Validation

3.6 Methodology implementation steps

The goal of this section is to develop methodology of the study step by step. The research was conducted using an HP 15 laptop with a core i7 1.8GHz base clock and 8GB of RAM running Microsoft Windows 10 64-bit (operating system). The algorithm is developed on the MATLAB 2018a software platform. The program can be tested or run on any computer/laptop platforms where MATLAB software version 2018a or newer is available.

3.6.1 The MATLAB experiment simulation

The experiment is done as follows: Assorted 332 photos of pineapple slices are taken at Del Monte cannery packing tables. The photos with distortion, blurred and pineapple slices image not centered are eliminated. The best 250 photos are selected for the experiment, 50 photos for each of five categories. The photos are then cropped to center the pineapple slices and then down sized to 256x256 pixel size using MS Paint.

The MATLAB script code is programmed as follows: The Otsu's code was built and tested with a photo dataset, and it was discovered to effectively remove the background. The feature extraction function code is then programmed and tested to be effective. The database of training features and training labels is then created. PCA algorithm is then used to reduce the feature space and it is found that at 12 principal components do preserved 99%. Various PCA plot are analyzed. The three multiclass SVMs algorithms are created and tested with the three kernel function and found to classify the images effectively. The SVM classifier weights are adjusted as the performance is validated through 5-fold cross validation. A Graphical User Interface GUI is then created to present the system working effectively.

To run the script, one open MATLAB software, open the folder named code in the CD and double click on PineappleSorting_GUI.m file, then click run. Once the GUI is running, one then load the image from the database by clicking on load image command, double click on database folder, double click on one of five folders and select one image. Once the image is loaded, one can click segment command, wait for 30 second on first run and 0.1 sec on subsequent run, to generate segmented output image. Two more figures are also generated. Then one click feature extraction command to generate the 90 features and displayed in GUI. Secondly, one chose the classifier, selects both the SVM and kernel, to use and then click the classification command. The system classifies the image and give the, a dialog box, computation time, class and the voting score. Thirdly, there is an optional command to show various PCA plot where further analysis can be done. Lastly, click accuracy command, wait for iteration to run, where 5-fold cross validation run to test accuracy of the classifiers. The classifier iterations correct rate, specificity, sensitivity, Positive Predictive Value and Negative Predictive Value are displayed in command window. Confusion matrix and best correct rate are shown for in the GUI.

3.6.2 Pineapple slices classification System

Below points is an explanation of the flowchart of the developed system shown in Figure 3.8. Numbers in this figure are achieved by experiments as follows: The input data is a database of 250 images consisting of 5 categories of pineapple slices, and each image is of size 256 × 256. The ninety features are extracted from each image of size 256 × 256. These ninety features contain six (6) colour moment features, ten (10) texture features, sixty four (64) colour histogram features and ten (10) shape features.

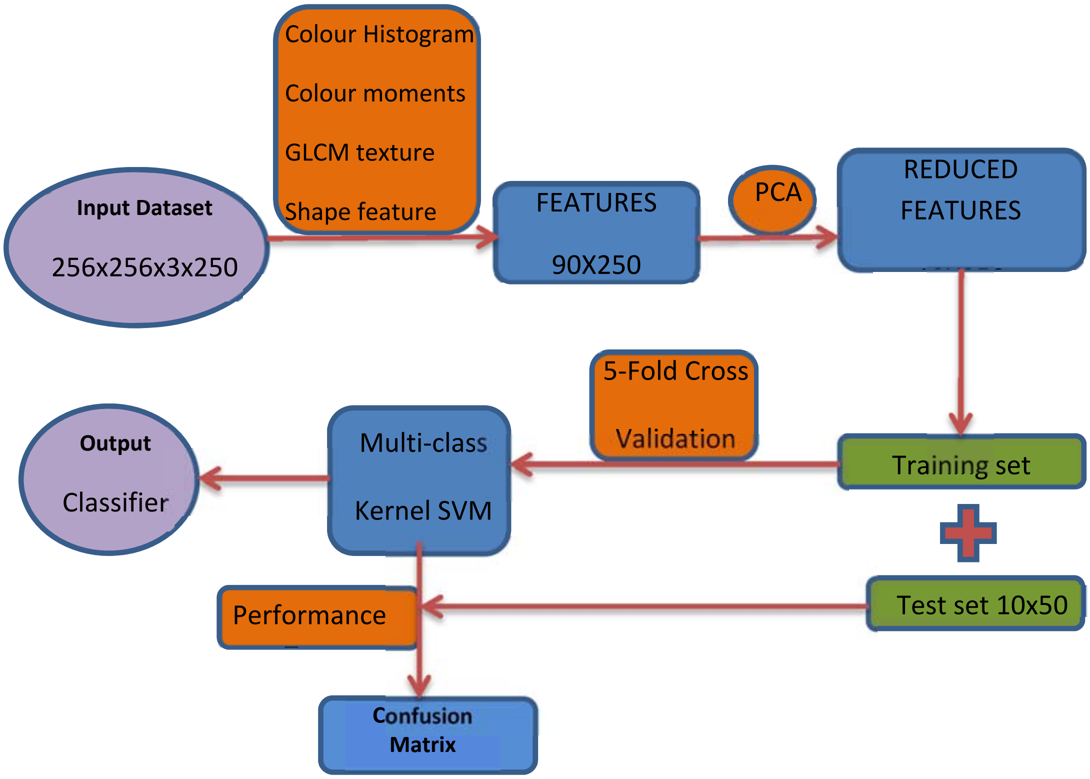


Figure 3.8: The flowchart of the developed pineapple slices classification algorithm.

The ninety (90) features are reduced to ten (12) features through PCA, and preserve 99% energy as the selection standard. The images are divided into two sets i.e. training set (200) and test set (50) in the ratio of 4:1. The training set is subsequently put through a -fold cross validation process. The training set is used to train the multiclass kSVM. The weights for kSVM are changed to achieve the minimum possible 5-fold cross validation average error. The test dataset is created by picking each group at random, and it is used to evaluate the classifier's performance and calculate the Confusion Matrix. If the classifier is sufficient, output it; otherwise, return to step 5 to retrain the SVM weights.

CHAPTER FOUR

RESULT AND DISCUSSION

The goal of this chapter is to present analysis and discussion of the results obtained from the study. The developed pineapple slices classification algorithm using a multi-class SVM and hybrid feature extraction technique is tested by simulation in MATLAB Software.

4.1 Dataset

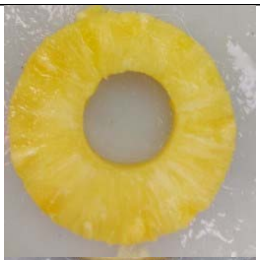
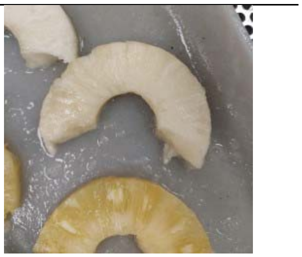
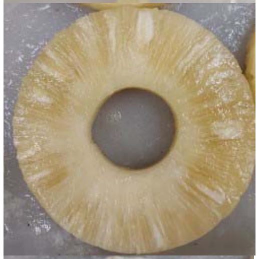
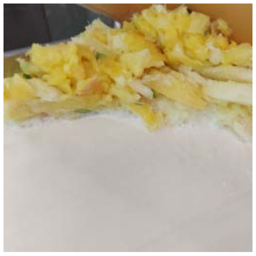
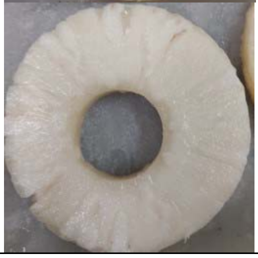
Table 4.1: Pineapple slices images database

Pineapple from both types i.e. Smooth Caen and MD2		
NO#	**Pineapple slice category**	**Quantity**
1	Fancy ¾ and full yellow	50
2	Fancy ½	50
3	Choice	50
4	Broken	50
5	Reject	50
In total, there will be 250 images		

The dataset of pineapple slice images below was result of on-site data collecting, at Del Monte, via mobile phone and digital camera. Background areas are removed using the Otsu's method, then the photos are cropped to leave the pineapple slice in the middle, and then down sampled to 256x256 pixels using MS Paint. All five categories were given equal 50 images so that the classification is not skewed in either way. The table 4.1 shows dataset comprises of five different categories of pineapple slices with each 50 samples.

The table 4.2 shows the samples of different types of slices images in the dataset. The sample of each of five categories is given. The striking similarities of the categories is shown and a unique method is required to extract the features that show the small differences to enable automatic classifications. The database of features is created by extracting 90 feature of each 250 images. A table of 250 rows (instances) and 90 columns (features) is made by using a creating algorithm below:

Table 4.2: Samples of pineapple slices dataset of five different categories

Pineapple types	Image Sample	Pineapple types	Image Sample
Fancy ¾ and full yellow		Broken	
Fancy ½		Reject	
Choice			

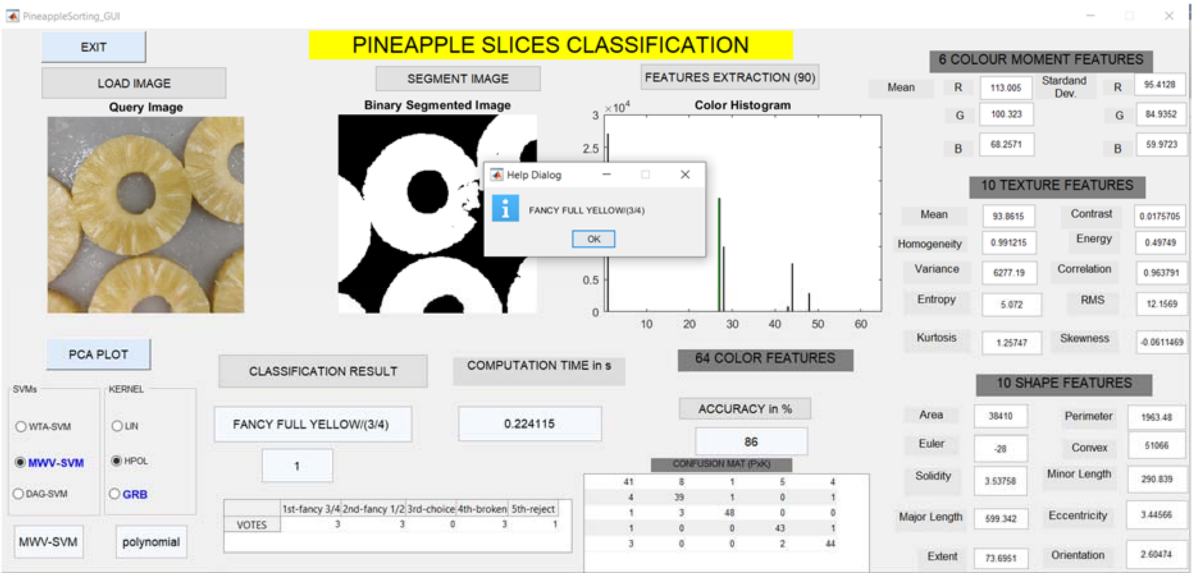
Features database creating algorithm below:

```
function DB_creating(Feature_Vector)
class=input('Enter the Class(Number from 1-5)');
try
   load Training_Data;
   Train_Feat=[Train_Feat; Feature_Vector];
   Train_Label=[Train_Label,class];
   save Training_Data.mat  Train_Feat Train_Label
catch
   helpdlg(' Error: Array mismatch for CAT or the first input');
   disp(' Error: Array mismatch for CAT or the first input');

   Train_Feat=[Feature_Vector];
   Train_Label=[class];
   save Training_Data2.mat  Train_Feat Train_Label
end
end
```

The function DB_creating (Feature_Vector) has an input argument Feature_Vector which contains 90 features extracted from the image loaded. [Feature_Vector]= Extract_FeaturesofSlices (bin_Image, RGB_Imag). The first 64 columns store histogram data, the next 6 columns store colour moment features, the next 10 columns store the texture features and the last 10 columns store shape features. All 250 images' features are stored in the database 250 rows. The first 50 rows form the fancy ¾ group, next 50 rows are from fancy ½ and it followed by choice, broken and reject in that order.

4.1.1 The MATLAB experiment Graphical User Interface

The figure 4.1 shows the GUI (Graphical User Interface) used to implement the pineapple slices classification system. A load command button is used to query the image from database, the image is then segmented using Otsu thresholding. The 90 features are then extracted and used to classify and automate the manual method of sorting pineapple slices. The performance of each classifies is validated and compared



Help Dialog

FANCY FULL YELLOW/(3/4)

SVMs: ○ WTA-SVM ● MWV-SVM ○ DAG-SVM

KERNEL: ○ LIN ● HPOL ○ GRB

MWV-SVM

polynomial

CLASSIFICATION RESULT: FANCY FULL YELLOW/(3/4)

1

COMPUTATION TIME in s: 0.224115

ACCURACY in %: 86

	1st-fancy 3/4	2nd-fancy 1/2	3rd-choice	4th-broken	5th-reject
VOTES	3	3	0	3	1

CONFUSION MAT (PxK)

41	8	1	5	4
4	39	1	0	1
1	3	48	0	0
1	0	0	43	1
3	0	0	2	44

6 COLOUR MOMENT FEATURES

Mean	R	113.005	Stardand Dev.	R	95.4128
	G	100.323		G	84.9352
	B	68.2571		B	59.9723

10 TEXTURE FEATURES

Mean	93.8615	Contrast	0.0175705
Homogeneity	0.991215	Energy	0.49749
Variance	6277.19	Correlation	0.963791
Entropy	5.072	RMS	12.1569
Kurtosis	1.25747	Skewness	-0.0611469

10 SHAPE FEATURES

Area	38410	Perimeter	1963.48
Euler	-28	Convex	51066
Solidity	3.53758	Minor Length	290.839
Major Length	599.342	Eccentricity	3.44566
Extent	73.6951	Orientation	2.60474

Figure 4.1: Pineapple Slices Classification GUI

4.2 Image segmentation result

The result of the segmentation is to remove the background and remain with the pineapple slice image. This will enable the features extracted to be form the pineapple slices only and not the background. Image preprocessing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and colour corrections. The actual step include read image, resize image, remove noise (denoise), segmentation and morphology (smoothing edges).

Contrast enhancement of colour images is typically done by converting the image to a colour space that has image luminosity as one of its components, such as the L*a*b* colour space. Contrast adjustment is performed on the luminosity layer L* only, and then the image is converted back to the RGB colour space. The first hole-filling step is used to fill in the holes in the depth image captured from camera. Hence a hole is an area of dark pixels surrounded by light pixels in gray images and black pixels surrounded by white pixels in binary image.

The figure 4.2 shows the segmented image and background removing. Otsu’s method is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background.

Opening and Closing are dual operations used in Digital Image Processing for restoring an eroded image. Opening is generally used to restore or recover the original image to the maximum possible extent. The background is removed by image masking. This is using masks or selective adjustments to isolate where an adjustment is taking place. The following steps are implemented using the following algorithm:

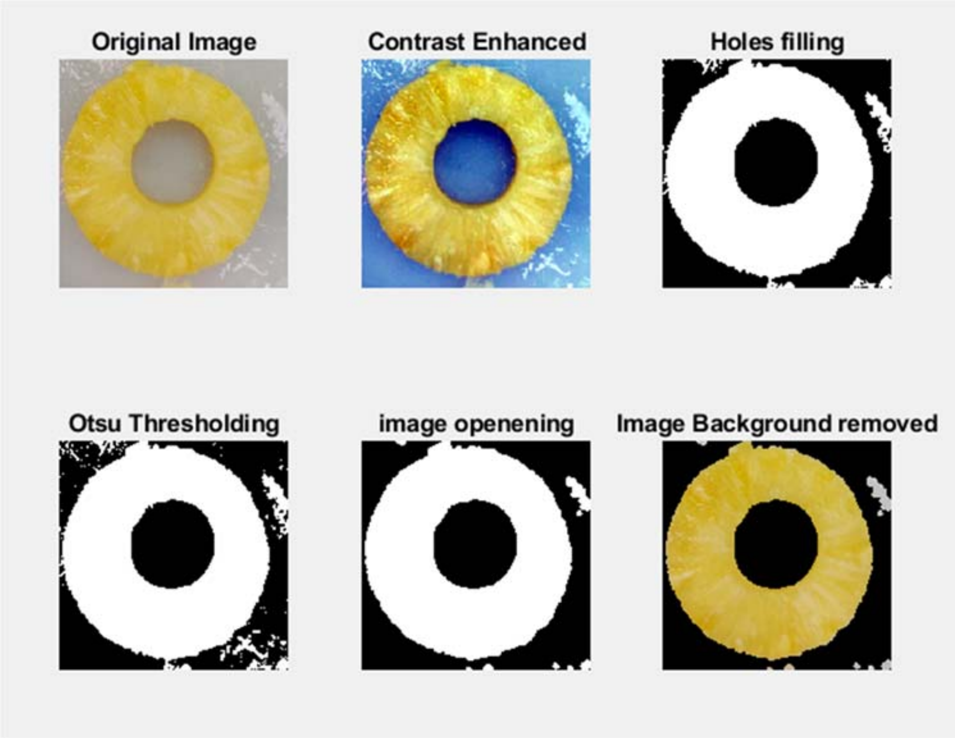


Figure 4.2: Background removing result

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 clc
[filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick a Pineapple slice Image File');
I = imread([pathname,filename]);
% I = imresize(I,[256,256]);
% I2 = imresize(I,[300,400]);
axes(handles.axes1);
imshow(I);title('Query Image');
ss = ones(300,400);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
imshow(ss);
handles.ImgData1 = I;
% Update GUI
guidata(hObject,handles);
% --- Executes on button press in pushbutton3.
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Enhance Contrast
I2 = handles.ImgData1;
I3 = imadjust(I2,stretchlim(I2));
% splitting RGB colour space
 rmat=I3(:,:,1);
 gmat=I3(:,:,2);
 bmat=I3(:,:,3);
 figure,
subplot(2,2,1),imshow(I3), title('Enhanced Image')
subplot(2,2,2),imhist(rmat), title('Red channel Histogram');
subplot(2,2,3),imhist(gmat), title('Green channel Histogram');
subplot(2,2,4),imhist(bmat), title('Blue channel Histogram');
% Otsu Segmentation
I_Otsu = im2bw(rmat,graythresh(I3));
%holes filling
Icfilled=imfill(~I_Otsu,'holes');
I_fill=~Icfilled;
% opening
se=strel('disk', 4);
I_open=imopen(I_fill,se);
% Create masked image.
maskedImage = I2;
%removing background
maskedImage(repmat(~I_open,[1 1 3])) = 0;
%  displaying otsu thresholding and Segmentation
figure,
subplot(2,3,1),imshow(I2), title('Original Image')
subplot(2,3,2),imshow(I3), title('Contrast Enhanced');
subplot(2,3,3),imshow(I_fill), title('Holes filling');
subplot(2,3,4),imshow(I_Otsu), title('Otsu Thresholding');
subplot(2,3,5),imshow(I_open), title('image openening');
subplot(2,3,6),imshow(maskedImage), title('Image Background removed');
%binary image
bin_Image=I_open;
handles.ImgData2=bin_Image;
%RGB image
RGB_Image=maskedImage;
handles.ImgData3=RGB_Image;
axes(handles.axes2);
imshow(bin_Image);title(' Binary Segmented Image ');
```

```
% Update GUI
guidata(hObject,handles);
% --- Executes on button press in pushbutton8.
```

4.3 Implementation of feature extraction:

The unique hybrid feature extraction method is extract the 90 features. The features are generated in a 1x90 vector and the classifier used the stored feature vector to classify the image and assig then appropriate class. The following algorithm was used to implement the feature extraction:

```
function feat_V = Extract_FeaturesofSlices(bin_I,RGB_I)
% Evaluate 90 pineapple slice features
colourFeat_64=rgbhist(RGB_I);
colourMoments_6=colourMoments(RGB_I);
textureFeat_10=textfeat(bin_I, RGB_I);
shapeFeat_10=shapefeat(bin_I);
feat_V= [colourFeat_64, colourMoments_6, textureFeat_10, shapeFeat_10];
```

4.3.1 Extracting the 64 colour features:

A colour histogram result show a unique signature of each image that can be used to classify the image uniquely. The code is used to split the image colour space into four bin and a colour histogram of 64 colour information generated. A colour histogram is a representation of the distribution of colours in an image in photography and image processing. A colour histogram is the number of pixels in a digital image that contain colours from a preset set of colour ranges that span the image's colour space, or the set of all possible colours. The following algorithm is used to implement the colour extraction:

```
%% RGBHIST: colour Histogram of an RGB image.
%Evaluate 64 colour features
 function H = rgbhist(I)
if (size(I, 3) ~= 3)
   error('rgbhist:numberOfSamples', 'Input image must be RGB.')
end
% nBins=4
nBins=4;
```

```
H=zeros([nBins nBins nBins]);
for i=1:size(I,1)
    for j=1:size(I,2)
        p=double(reshape(I(i,j,:),[1 3]));
        p=floor(p/(256/nBins))+1;
        H(p(1),p(2),p(3))=H(p(1),p(2),p(3))+1;
    end
end
% Un-Normalized histogram
H=H(:);
% return
 H=H.';
figure,
bar (H), title('Colour Histogram');
 end
```

The figure 4.3 below show the 64 colour histogram

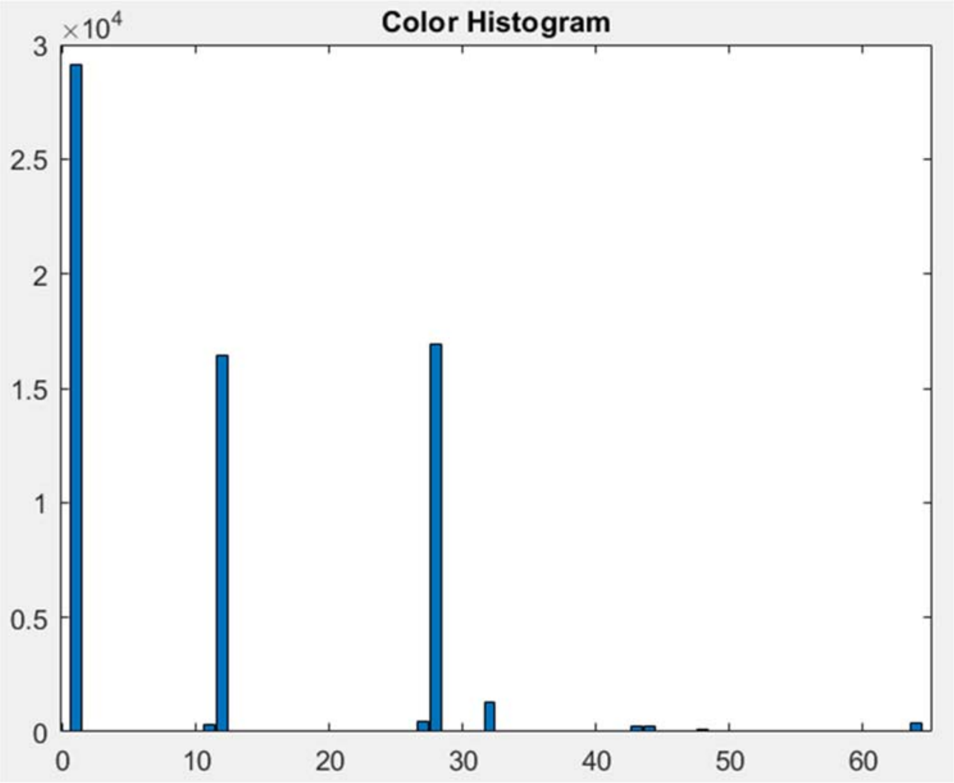


Figure 4.3: Colour histogram result

4.3.2 Colour moment

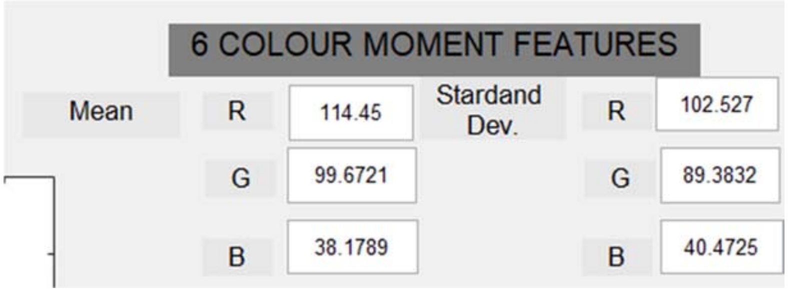
The first six colour moment result are show below. This features are unique for every image and are used in classification. Colour moments characterize the colour distribution in an image in the same way as moments of central tendency characterize a probability distribution. Colour moments are predominantly utilized for colour indexing as features in image retrieval applications to compare the similarity of two images based on colour. Typically, one image is compared to a database of digital photos with precomputed attributes to locate and get a comparable Image. Each image comparison yields a similarity score; the lower this value, the more similar the two photos are expected to be.

Colour moments are scale- and rotation-invariant. In image retrieval applications, typically only the first three colour moments are employed as features, as the majority of colour distribution information is stored in the low-order moments. Due to the fact that colour moments carry both shape and colour information, they are a useful feature under varying illumination conditions, but they cannot effectively handle occlusion. Moments of colour can be calculated for any colour model. Every channel, three colour moments are computed (e.g. 9 moments if the colour model is RGB). Moments of colour are computed in the same manner as moments of probability distributions. The below algorithm implement the colour moment extraction:

```
%% output: 1x6 vector containing the 2 first color momenst from each R,G,B channel
function colorMoments = colorMoments(image)
% input: image to be analyzed and extract 2 first moments from each R,G,B
% extract color channels
R = double(image(:, :, 1));
G = double(image(:, :, 2));
B = double(image(:, :, 3));
% compute 2 first color moments from each channel
meanR = mean( R(:) );
stdR  = std( R(:) );
meanG = mean( G(:) );
stdG  = std( G(:) );
meanB = mean( B(:) );
stdB  = std( B(:) );
% construct output vector
```

```
colorMoments = zeros(1, 6);
colorMoments(1, :) = [meanR stdR meanG stdG meanB stdB];
% clear workspace
%clear('R', 'G', 'B', 'meanR', 'stdR', 'meanG', 'stdG', 'meanB', 'stdB');
end
```

The below figure 4.4 show a sample 6 colour moment features results:



6 COLOUR MOMENT FEATURES

Mean	R	114.45	Stardand Dev.	R	102.527
	G	99.6721		G	89.3832
	B	38.1789		B	40.4725

Figure 4.4: colour moment result

4.3.3 Extracting the 10 texture features

GLCM is a statistical texture analysis approach of the second order. It analyzes the spatial relationship between pixels and defines the frequency with which a specific pixel combination appears in a picture along a given direction Θ and distance d. glcms = graycomatrix(I) generates a gray-level co-occurrence matrix (GLCM) from the given image. A gray-level co-occurrence matrix is also known as a gray-level spatial dependence matrix. Graycomatrix generates the GLCM by calculating the frequency with which a pixel with gray-level (grayscale intensity) value I is horizontally adjacent to a pixel with value j. Each element (i,j) in glcm specifies the number of occurrences of pixel I horizontally adjacent to pixel j.

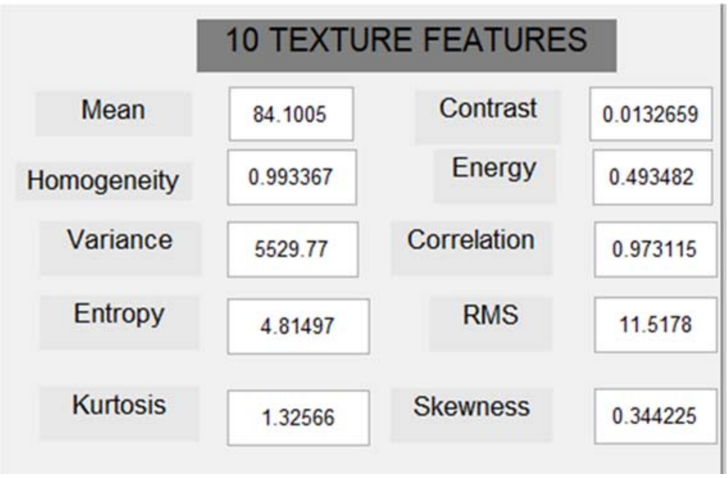
The ten texture features are extracted to form the feature space: The mean gives one an idea of what pixel colour to choose to summarize the colour of the complete image. The mean has the same dimension as your data (in case of pixels, think of intensity), while the variance has the dimension of your data squared (so intensity^2). Contrast is the difference in luminance or colour that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the colour and brightness of the object and other objects within the same field of view. The homogeneity of a region of an image depends on the intensities of the considered pixels. Any permutation of that pixels has the same gray values, but in different order, so the homogeneity should be the same. The energy is a measure of the image's localized change. The energy is referred to by a variety of names and in a variety of circumstances, but it all refers to the same entity. It's the pace at which the colour, brightness, and magnitude of pixels change over time in a given area. An image is a collection of data points on light intensity, variance of the image implies a gross measure of the imprecision/variation about the target value. variance has the dimension of your data squared (so intensity^2). Correlation is the process of moving a filter mask often referred to as kernel over the image and computing the sum of products at each location. Correlation is the function of displacement of the filter. The degree of randomness in an image is measured by its entropy, or average information. The entropy is useful in image coding because it sets a lower limit on the average coding length in bits per pixel that an optimal coding scheme can achieve without losing any information. The RMS value is defined as the square root of the squared function's mean value. Kurtosis is a measurement of the cumulative weight of a distribution's tails in relation to its center. Skewness is a measure of the asymmetry of the probability distribution around the mean of a real-valued random variable.

The below code is used to extract the 10 texture features:

```
%% Evaluate 10 texture features
function T =textfeat(I_b,I_r)
% Create the Gray Level Cooccurance Matrices (GLCMs)
glcms = graycomatrix(I_b);
%Evaluate 10 texture features
% Derive Statistics from GLCM
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
```

```
Homogeneity = stats.Homogeneity;
Mean = mean2(I_r);
Entropy = entropy(I_r);
Variance = mean2(var(double(I_r)));
% a = sum(double(I_r(:)));
% Smoothness = 1-(1/(1+a));
RMS = mean2(rms(I_r));
Kurtosis = kurtosis(double(I_r(:)));
Skewness = skewness(double(I_r(:)));
 T=[Mean, Contrast, Homogeneity, Energy, Variance, Correlation, Entropy, RMS,
Kurtosis, Skewness];
end
```

The below figure 4.5 show a sample texture features results



10 TEXTURE FEATURES

Mean	84.1005	Contrast	0.0132659
Homogeneity	0.993367	Energy	0.493482
Variance	5529.77	Correlation	0.973115
Entropy	4.81497	RMS	11.5178
Kurtosis	1.32566	Skewness	0.344225

Figure 4.5: Texture features

4.3.4 Extracting the 10 shape features

STATS = regionprops(L,properties) quantifies a collection of properties for each labeled region in the label matrix L. Positive integer elements of L represent several regions. For instance, the set of L elements equal to 1 belongs to area 1, the set of L elements equal to 2 refers to region 2, etc. STATS is a structural array with a length of max(L(:)). According to the characteristics, the fields of the structure array imply distinct measurements for each zone. Properties can be a comma-separated list of strings, a cell array of strings, the string 'all', or the string 'basic'. This table contains the valid property string values. There is no case sensitivity for property strings, and they can be truncated. Stats = *regionprops* (BW , properties) returns measurements for the set of properties for each 8-connected component (object) in the binary image, BW. The ten shape features are extracted to form the feature space. The table 4.3 shows regionprop properties.

The area of the objects in binary image is a scalar whose value corresponds roughly to the total number of on pixels in the image, however, because different patterns of pixels are weighted differently, they may not be precisely the same. Perimeter is the count of pixels of objects in the input image around it. The Euler number is a measure of the topology of an image. It is defined as the total number of objects in the image minus the number of holes in those objects. You can use either 4- or 8-connected neighborhoods. The research uses the later method. A binary image's convex hull is the set of pixels included in the smallest convex polygon that surrounds all white pixels in the input. Solidity (convexity) is image object area divided by its convex hull area. The two axes of the fitted ellipse to your object are major-/minor-length.

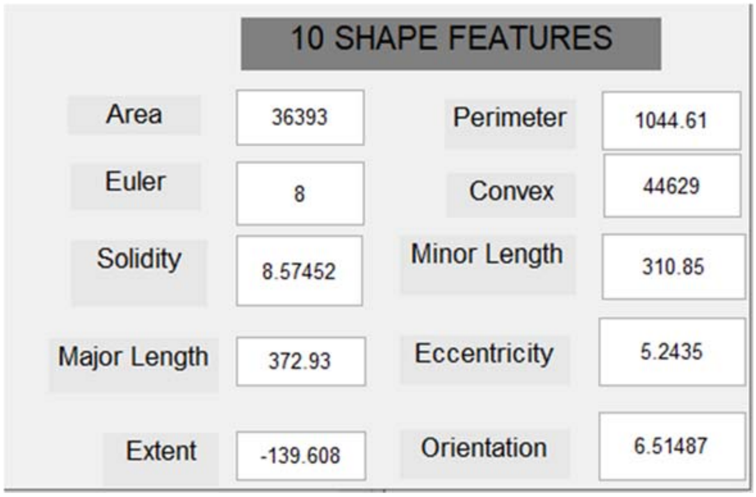
Table 4.3: Regionprop properties

'Area'	**'EquivDiameter'**	**'MajorAxisLength'**
'BoundingBox'	'EulerNumber'	'MinorAxisLength'
'Centroid'	'Extent'	'Orientation'
'ConvexArea'	'Extrema'	'PixelIdxList'
'ConvexHull'	'FilledArea'	'PixelList'
'ConvexImage'	'FilledImage'	'Solidity'
'Eccentricity'	'Image'	'SubarrayIdx'

Feret-Diameter is linked to an angle for an ellipse. You can calculate it either for an angle or for Min-/Max-Feret-Diameter. Eccentricity measures the shortest length of the paths from a given vertex v to reach any other vertex w of a connected graph. For a connected region of a digital image it is defined through its neighbourhood graph and the given metric. In mathematics, the eccentricity of a conic section is a non-negative real number that uniquely characterizes its shape. One can think of the eccentricity as a measure of how much a conic section deviates from being circular. In particular: The eccentricity of a circle is zero. The extent of a feature is also known as its envelope. It is defined as a minimum bounding rectangle with a width (x-value) and height (y-value). Orientation is the relative arrangements of points after a transformation or after traveling around a geometric figure. The following algorithm was used to extract 10 shape features:

```
%% Evaluate 10 Shape features


function S =shapefeat(I_b)
   bw=I_b;
 stats = regionprops('table',bw,'Area','Perimeter','EulerNumber','ConvexArea','Solidity','
MinorAxisLength', 'MajorAxisLength','Eccentricity','Orientation','Extent' );
  % Get shape features
     area=stats.Area;
     perimeter=stats.Perimeter;
     euler=stats.EulerNumber;
     convex=stats.ConvexArea;
     solidity=stats.Solidity;
     minor=stats.MinorAxisLength;
     major=stats.MajorAxisLength;
     eccentricity=stats.Eccentricity;
     orientation=stats.Orientation;
     extent=stats.Extent;
    Vectors=[area, perimeter, euler, convex, solidity, minor, major, eccentricity,
orientation, extent];
 if(size(Vectors,1))>=2
    S=sum(Vectors);
 else
     S=Vectors;
 end
end
```

The below figure 4.6 show a sample shape features results



10 SHAPE FEATURES

Area	36393	Perimeter	1044.61
Euler	8	Convex	44629
Solidity	8.57452	Minor Length	310.85
Major Length	372.93	Eccentricity	5.2435
Extent	-139.608	Orientation	6.51487

Figure 4.6: Shape features

4.4 Implementing multiclass kernel SVMs

The three SVMs are implemented as follows:

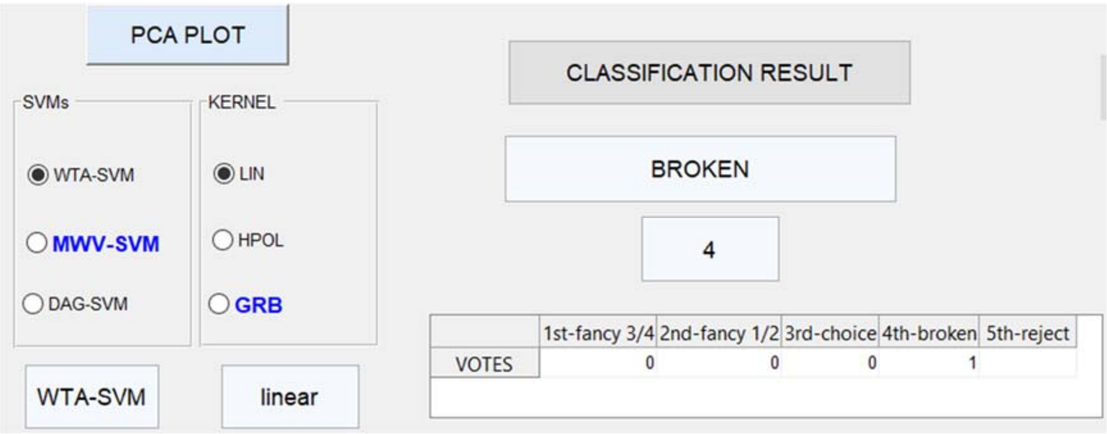
4.4.1 WTA-SVM

The Winner Take All SVM has only one vote for the winning class (see 4.1). The one vs all SVM algorithm evaluate whether the test sample fall in class 1 or the rest 4 classes. If the sample does not fall in the class 1, fancy ¾, the vote is zero for the class else the vote is one and loop terminate. In case of the zero vote, the first class is eliminated and the algorithm compare if the test sample fall in second class or the remaining 3 classes. The voting continue until the fourth class. If the sample fails the first 4 classes testing the algorithm assign it to fifth class, reject, by default. The figure 4.7 below shows the 4th class, broken, is chosen after scoring one votes. Classes 1, 2 and 3 fails and assigned zero score. The fifth class is not evaluated, as the algorithm run until a class win and the loop terminates. The following algorithm was developed to implement the WTA-SVM:

Winner-Takes-All SVM (WTA-SVM) algorithm

```
function [itrfin,votes] = multisvm_WTA( T,C,test,kernel )
%Inputs:  T=Training  Matrix,  C=Group,  test=Testing  matrix,  kernel=kernelfunction
Outputs: itrfin=Resultant class
% classification by a winner-takes-all (WTA) strategy i.e. one-versus-All approach.
kernel_fxn=kernel;
c4=[];
c3=[];
T=Tb;
Tb=T;
tst=test;
Cb=C;
C=Cb;
u=unique(C);
N=length(u);
votes=[];
    j=1;
    k=1;
    if(N>2)%testing (C > 2) classes.
        itr=1;
        classes=0;
        cond=max(C)-min(C);
        while((classes~=1)&&(itr<=length(u))&& size(C,2)>1 && cond>0)
        %This while loop is the winner-takes-all (WTA) strategy
            c1=(C==u(itr));
            newClass=c1;
            if kernel_fxn==1
            svmStruct = fitcsvm(T,newClass,'KernelFunction','linear','BoxConstraint',6.7);
% I am using linear kernel function
            classes = predict(svmStruct,tst);%return 0 0 1 for choice
            elseif kernel_fxn == 2
            svmStruct =
fitcsvm(T,newClass,'KernelFunction','polynomial','PolynomialOrder',1,'BoxConstraint',6
.7);   % I am using polynomial kernel function, polyorder =1
 classes = predict(svmStruct,tst);
elseif kernel_fxn == 3
   svmStruct = fitcsvm(T,newClass,'KernelFunction','rbf','KernelScale','auto',
'KernelScale',4.2,'BoxConstraint',6.7);
% I am using rbf kernel function
            classes = predict(svmStruct,tst);
            end
            % This is the loop for Reduction of Training Set
            for i=1:size(newClass,2)
  if newClass(1,i)==0;
                    c3(k,:)=T(i,:);
                    k=k+1;
                end
            end
        T=c3;
        c3=[];
        k=1;
            % This is the loop for reduction of group
            for i=1:size(newClass,2)
                if newClass(1,i)==0;
                    c4(1,j)=C(1,i);
                    j=j+1;
                end
            end
        C=c4;
        c4=[];
        j=1;
        cond=max(C)-min(C); % Condition for avoiding group
                            %to contain similar type of values
                            %and the reduce them to process
            % This condition can select the particular value of iteration
            % base on classes
            votes(itr)=classes;
            if classes~=1
                itr=itr+1;
            end
        end
    end
itrfin=u(itr);
end
%--------------------------------------------------------------------------%
```

The figure 4.7 below show a sample classification results and voting



SVMs

● WTA-SVM
○ MWV-SVM
○ DAG-SVM

KERNEL

● LIN
○ HPOL
○ GRB

WTA-SVM

linear

BROKEN

4

	1st-fancy 3/4	2nd-fancy 1/2	3rd-choice	4th-broken	5th-reject
VOTES	0	0	0	1	

Figure 4.7: WTA-SVM classification result

4.4.2 MWV-SVM

The Max Wins Voting SVM chooses the class with maximum votes. The one vs one SVM algorithm evaluate in which class the test sample fall in and give one vote to the wining class in the following 10 voting contest: 1v2, 1v3, 1v4, 1v5, 2v3, 2v4, 2v5, 3v4, 3v5 and 4v5. The algorithm is similar to group stages of soccer game. If the leading two classes have same voting score, the class with smaller index is chosen. The figure 4.8 shows the fourth class is chosen after scoring 4 votes, which is the maximum votes. The total votes are 10 (1+0+3+4+2=10), as they are result of ten contests. There is no tie in the first position. The following below algorithm to implement the MWV-SVM.

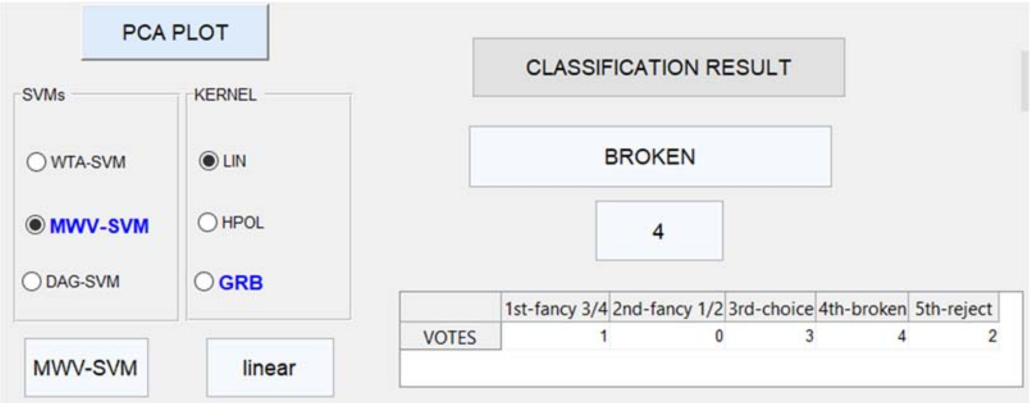
Max-Wins-Voting SVM (MWV-SVM) algorithm

```
function [itrfin,votes] = multisvm_MWV( T,C,test,kernel )
%Inputs: T=Training Matrix, C=Group, test=Testing matrix  kernel=kernel function
Outputs: itrfin=Resultant class
% Max-Wins-Voting SVM by one-versus-one approach
kernel_fxn=kernel;
Cb=C;
i=1;
Tb=T;
tst=test;
```

```
m=2;
u=unique(C);
k=1;
N=length(u);
p=1;
c3=[];
votes=[0 0 0 0 0];
valt=[];
classes=0;
for p=1:(N-1)%for m=1:4

   for m=p+1:N%for m=p+1:5
      valt=Cb==u(p)|Cb==u(m);%selecting labels for two classes
      label=Cb(valt==1);%100 labels of actual labels

      %selecting training dataset for respective two classes
         for i=1:size(valt,2)%for i:250
            if valt(1,i)==1;
               c3(k,:)=Tb(i,:);
               k=k+1;
            end
         end
      training=c3;
      c3=[];
      k=1;
         if kernel_fxn==1
          svmStruct = fitcsvm(training,label,'KernelFunction','linear','BoxConstraint',6.7);
% I am using linear kernel function
          classes = predict(svmStruct,tst);
         elseif kernel_fxn == 2
         svmStruct =
fitcsvm(training,label,'KernelFunction','polynomial','PolynomialOrder',1,'BoxConstraint'
,6.7);   % I am using polynomial kernel function, polyorder =1
          classes = predict(svmStruct,tst);
         elseif kernel_fxn == 3
         svmStruct = fitcsvm(training,label,'KernelFunction','rbf', 'KernelScale',
7.02229,'BoxConstraint',1);   % I am using rbf kernel function, sigma=4.2
          classes = predict(svmStruct,tst);
         end
% max-wins voting (MWV) strategy
  if classes==p
     votes(p)=votes(p)+1;
  else
     votes(m)=votes(m)+1;
```

```
 end
   m=m+1;
  end
   p=p+1;
end
   itrfin=min(u(votes==max(votes)));%min() for selecting class with the smallest index
                   %max() for labelling test with class
                   %with most labels
end
%---------------------------------------------------------------------------%
```

The figure 4.8 below show a sample classification results and voting by **MWV-SVM**



SVMs

○ WTA-SVM

● MWV-SVM

○ DAG-SVM

KERNEL

● LIN

○ HPOL

○ GRB

MWV-SVM

linear

BROKEN

4

	1st-fancy 3/4	2nd-fancy 1/2	3rd-choice	4th-broken	5th-reject
VOTES	1	0	3	4	2

Figure 4.8: MWV-SVM classification results

4.4.3 DAG-SVM

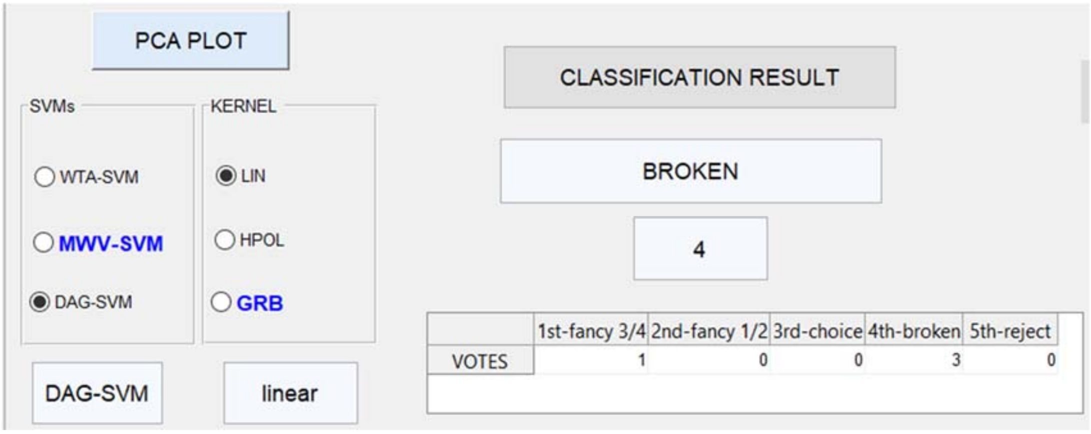
As MWV-SVM, the Directed Acyclic Graph SVM chooses the class with maximum votes. The one vs one SVM algorithm evaluates in which class the test sample fall in and give one vote to the wining class and the losing class is marked as failed and is eliminated and not tested again. The algorithm is similar to knock out stage of a soccer game. The path only takes four contest to arrive at conclusion. If leading two classes have same voting score, the class with smaller index is chosen. The figure 4.9 shows the fourth class is chosen after scoring 3 votes, which is the maximum votes. The total votes are 4 (1+0+0+3+0=4), as they are result of four contests i.e: 1v5, 1v4, 2v4, 3v4. Always the contest starts with the 1v5 contest; and in this case class 1 wins and class 5 fails and it is eliminated. In 1v4 contest, class 4 win and get 1 vote while class 1 fails and eliminated. In 2v4 and 3v4 contests, class 4 win and get a total of 3 votes. There is no tie in the first position. The below algorithm implemented the **DAG-SVM:**

Direct Acyclic Graph SVM (DAG-SVM) algorithm

```
function  [itrfin,votes] = multisvm_DAG( T,C,test,kernel )
%Inputs:  T=Training  Matrix,  C=Group,  test=Testing  matrix    kernel=kernelfunction
Outputs: itrfin=Resultant class
% Max-Wins-Voting SVM
% A Directed Acyclic Graph (DAG) by one-versus-one approach with elimination.
kernel_fxn=kernel;
Cb=C;
i=1;
Tb=T;
tst=test;
k=1;
u=unique(C);
p=1;
N=length(u);
m=5;
c3=[];
votes=[0 0 0 0 0];
valt=[];
classes=0;
itr=1;
while(itr<=(N-1)&&p~=m)%Directed Acyclic Graph (DAG) strategy
    valt=Cb==u(p)|Cb==u(m);%selecting labels for two classes
    label=Cb(valt==1);%100 labels of actual labels

    %selecting training dataset for respective two classes
      for i=1:size(valt,2)%for i:250
        if valt(1,i)==1;
          c3(k,:)=Tb(i,:);
          k=k+1;
        end
      end
    training=c3;
    c3=[];
    k=1;

      if kernel_fxn==1
```

```
             svmStruct = fitcsvm(training,label,'KernelFunction','linear');   % I am using
linear kernel function
             classes = predict(svmStruct,tst);%return 0 0 1 for choice
            elseif kernel_fxn == 2
            svmStruct =
fitcsvm(training,label,'KernelFunction','polynomial','PolynomialOrder',1);   % I am
using polynomial kernel function, polyorder =1
             classes = predict(svmStruct,tst);%return 0 0 1 for choice
            elseif kernel_fxn == 3
            svmStruct =
fitcsvm(training,label,'KernelFunction','rbf','KernelScale','auto','KernelScale',4.2,'BoxCo
nstraint',6.7);   % I am using rbf kernel function, sigma=4.2
             classes = predict(svmStruct,tst);%return 0 0 1 for choice
            end

 itr=itr+1;

  if classes==p
      votes(p)=votes(p)+1;
      m=m-1;
  else
      votes(m)=votes(m)+1;
      p=p+1;
  end
end
    itrfin=min(u(votes==max(votes)));%min() for selecting class with the smallest index
                                    %max() for labelling test with class
                                    %with most labels
end
```

The figure 4.9 below show a sample classification results and voting by DAG-SVM



SVMs: ○ WTA-SVM ○ MWV-SVM ● DAG-SVM

KERNEL: ● LIN ○ HPOL ○ GRB

CLASSIFICATION RESULT: BROKEN

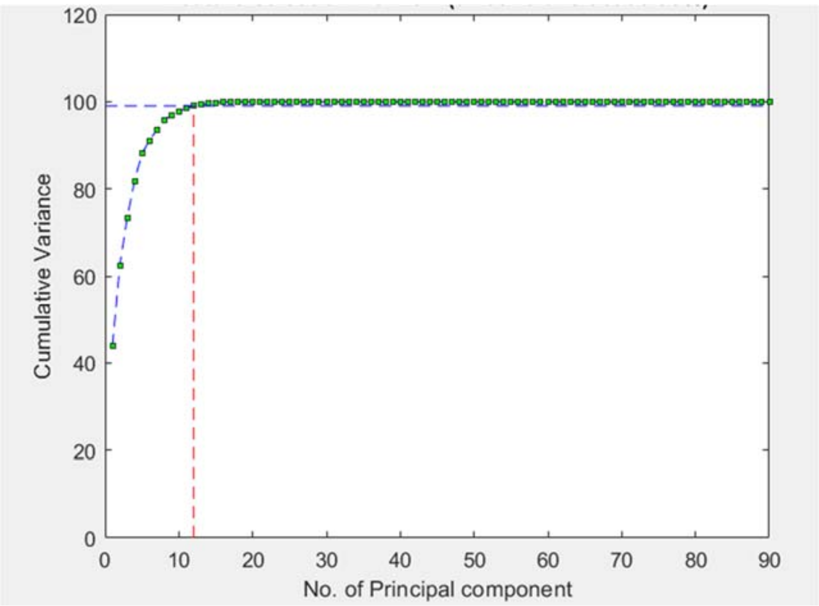
4

	1st-fancy 3/4	2nd-fancy 1/2	3rd-choice	4th-broken	5th-reject
VOTES	1	0	0	3	0

Figure 4.9: DAG-SVM classification results

4.5 PCA Results

Figure 4.10 shows a graph of cumulative sum of variance vs. number of reduced vectors via PCA. The figure shows the findings of the comprehensive data analysis of cumulative variance. It demonstrates that only twelve features can preserve 99 percent of energy. The reduced features cost only 13.3 percent (12/90) of the RAM required for the original 90 features. Subsequently, the algorithm can be accelerated significantly. Ninety (90) features are not a problem to the latest computers. But by working with 12 features, one can accelerate the test and training speed, and at the same time removing extra features will improve the classification accuracy.



.

Figure 4.10: Feature selection through PCA (threshold at 99%).

The table 4.4 below, show the PCA cumulative variances of transformed new features. The percentage of variance accounted for by the first n components is given in the Cumulative percentage column. The sum of the percentages of variation for the first and second components, for example, is the cumulative percentage for the second component. The first 12 PCs explain 99% of variance

Table 4.4: The PCA cumulative variances of transformed new features

Dimension	1	2	3	4	5	6	7
Variance (%)	43.84917	62.41931	73.39663	81.86644	88.10705	91.07444	93.48092
Dimension	8	9	10	11	12	13	14
Variance (%)	95.80852	96.86223	97.77846	98.46545	98.99664	99.34492	99.58282
Dimension	15	16	17	18	19	20	21
Variance (%)	99.77514	99.89002	99.97795	99.98983	99.99341	99.99575	99.99758
Dimension	22	23	24	25	26	27	28
Variance (%)	99.99926	99.99966	99.99979	99.9999	99.99997	99.99998	99.99999
Dimension	29	30	31	32	33	34	35
Variance (%)	99.99999	100	100	100	100	100	100
Dimension	36	37	38	39	40	41	42
Variance (%)	100	100	100	100	100	100	100
Dimension	43	44	45	46	47	48	49
Variance (%)	100	100	100	100	100	100	100
Dimension	50	51	52	53	54	55	56
Variance (%)	100	100	100	100	100	100	100
Dimension	57	58	59	60	61	62	63
Variance (%)	100	100	100	100	100	100	100
Dimension	64	65	66	67	68	69	70
Variance (%)	100	100	100	100	100	100	100
Dimension	71	72	73	74	75	76	77
Variance (%)	100	100	100	100	100	100	100
Dimension	78	79	80	81	82	83	84
Variance (%)	100	100	100	100	100	100	100
Dimension	85	86	87	88	89	90	

Variance (%)	100	100	100	100	100	100

In multivariate statistics, a scree plot is a line plot of the eigenvalues of factors or principal components in an analysis. Figure 4.11 depicts a scree plot that is used to assess how many factors to keep in an exploratory factor analysis (FA) or how many principal components to keep in a principal component analysis (PCA) (principal component analysis).

A scree plot depicts the amount of variance captured by each principal component in the data. The scree plot is a sharp curve that bends fast and flattens out if the first two or three PCs are sufficient to capture the substance of the data. It's a diagnostic tool for determining whether PCA works well with your data.PC1 records the most variation, PC2 the second, and so on.

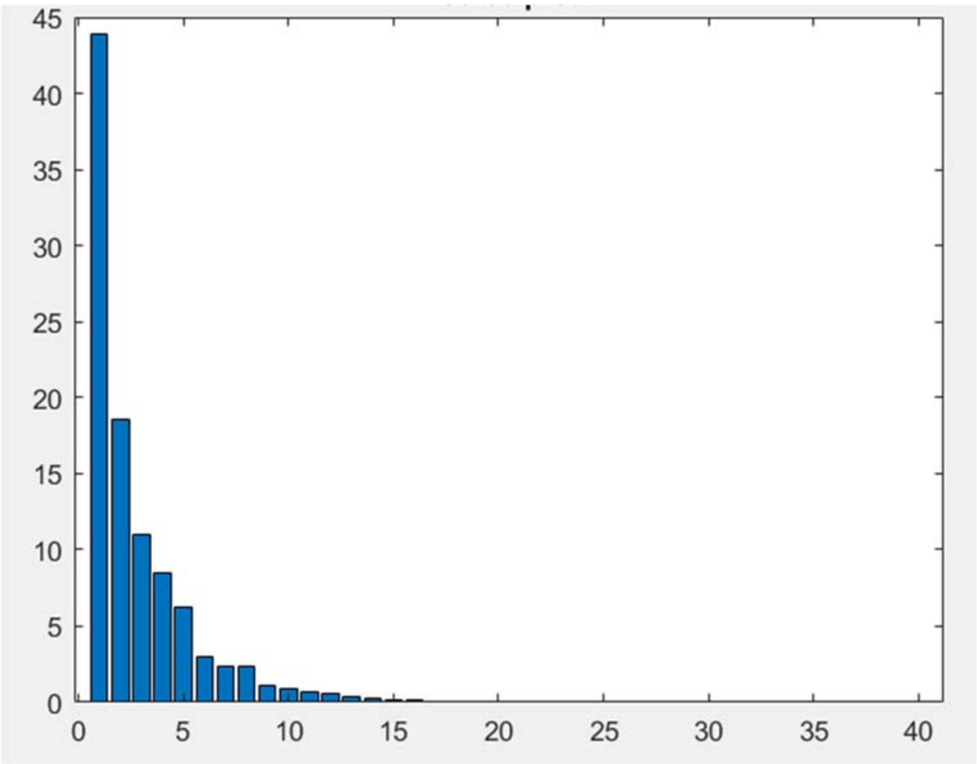


Figure 4.11: Scree Plot of the first 40 principal components

Each one adds to the data's information, and there are as many principal components as there are features in a PCA. Information is lost when PCs are left out.The loading plot on first two principal component is shown in Figure 4.12, from which one can evaluate the subsequent role of 90 features for classification. This will lead to identification of the relationship between samples and features and to select features more effectively.

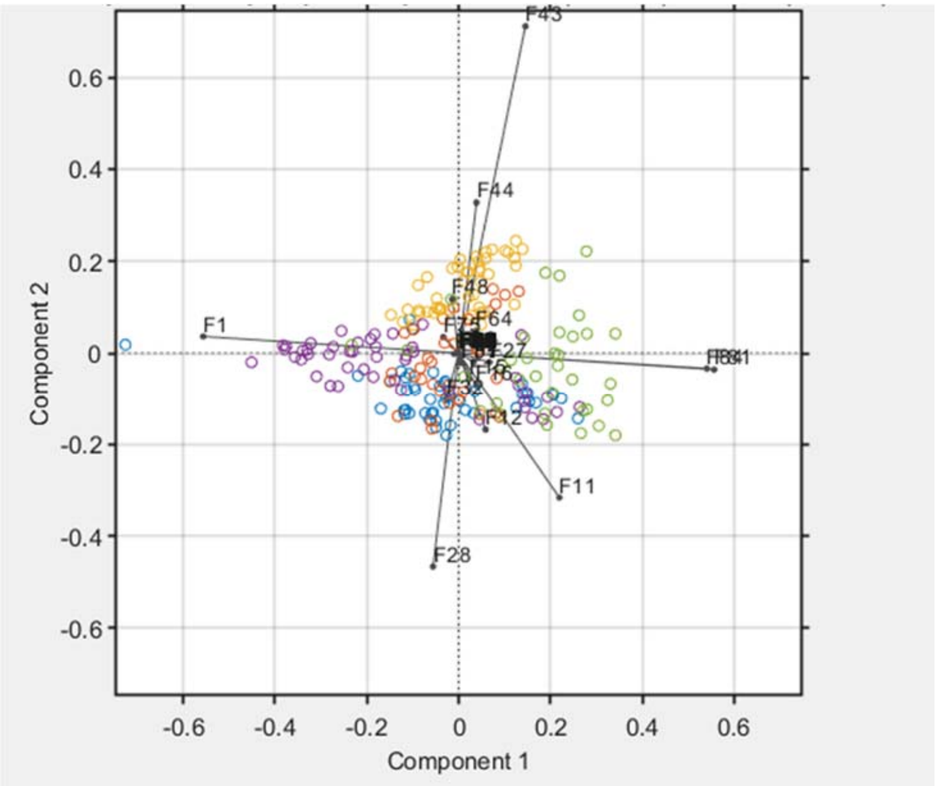


Figure 4.12: The biplot of two principal components (lines represent the 90 original features and dots represent the samples).

A PCA biplot displays both variable loadings (vectors) and sample PC scores (dots). The greater the influence of these vectors on a PC, the further they are from its origin. Loading plots also reveal how variables are related to one another: a big angle indicates negative correlation, a small angle shows positive connection, and a 90° angle indicates no association. The biplot in figure 4.12 shows two principal components PC1 with 43.85% contribution and PC2 with 18.57% contribution.

The PCA result are generated through a MATLAB code as follows:

```
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton9.
Train_Feat=handles.ImgData7;
Train_Label=handles.ImgData8;
groups=Train_Label.';
labels = {'F1' 'F2' 'F3' 'F4' 'F5' 'F6' 'F7' 'F8' 'F9' 'F10' 'F11' 'F12' 'F13' 'F14' 'F15' 'F16'
'F17' 'F18' 'F19' 'F20' 'F21' 'F22' 'F23' 'F24' 'F25' 'F26' 'F27' 'F28' 'F29' 'F30' 'F31' 'F32'
'F33' 'F34' 'F35' 'F36' 'F37' 'F38' 'F39' 'F40' 'F41' 'F42' 'F43' 'F44' 'F45' 'F46' 'F47' 'F48'
'F49' 'F50' 'F51' 'F52' 'F53' 'F54' 'F55' 'F56' 'F57' 'F58' 'F59'  'F60' 'F61' 'F62' 'F63' 'F64'
'F65' 'F66' 'F67' 'F68' 'F69' 'F70' 'F71' 'F72' 'F73' 'F74' 'F75' 'F76' 'F77' 'F78' 'F79' 'F80'
'F81' 'F82' 'F83' 'F84' 'F85' 'F86' 'F87' 'F88' 'F89' 'F90'};
[COEFF, SCORE, LATENT, TSQUARED, EXPLAINED] = pca(Train_Feat);
% figure, cdfplot(EXPLAINED);
% Display an empirical cumulative distribution function.
 figure, plot(cumsum(EXPLAINED),'--
bs','MarkerFaceColor','g','MarkerSize',4,'MarkerEdgeColor','k'), ylabel('Cumulative
Variance'), xlabel('No. of Principal component');
 title('Feature selection via PCA (threshold is set as 99%)');
 hold on
 plot([12 12],[0 99],'--r');
 hold on
 plot([0 90],[99 99],'--b');
% xlswrite('PCA.xlsx', cumsum(EXPLAINED));
figure;
%  biplotG(loadings, scores, 'Groups', groups, 'VarLabels', labels)
biplotG(COEFF, SCORE,'Groups', groups, 'VarLabels', labels);
title('The biplot of two pricipal components PC1(43.85%) and PC2(18.57%)');
grid on;
figure;
bar(EXPLAINED(1:40,:), 0.8);
title('scree plot');
clc;
% Update GUI
guidata(hObject,handles);
```

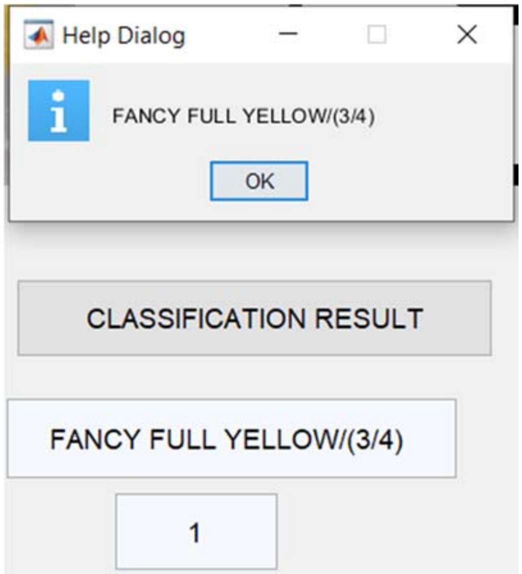
4.6 SVM Results

The multiclass SVMs (WTA, MWV, and DAG) is used to classify the pineapples slice into five categories as follows: fancy ¾, fancy ½, choice, broken and reject. The classifies are first tuned with the parameter the give the highest accuracy. An IF function is used to choose the result of classification. The code used is as follows:

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB handles
structure with handles and user data (see GUIDATA)

% Load Test Features
Feature_Vector=handles.ImgData4;
% Load All The Features
load('Training_Data.mat')
handles.ImgData7 = Train_Feat;
handles.ImgData8=Train_Label;
% cat the matrixes
Train_Feat=[Train_Feat; Feature_Vector];
% standardize to have unity variance and zero mean
Train_Feat=zscore(Train_Feat);
% reducing dimesionality
[COEFF, SCORE] = pca(Train_Feat);
% retaining atleast 99% energy
Train_Feat=SCORE(:,1:12);
% Put the test features into variable 'test'
test=Train_Feat(end,:);
% Put the training features into variable 'Train_Feat'
Train_Feat=Train_Feat(1:end-1,:);
kernel=handles.ImgData5;
svm=handles.ImgData6;
if svm==1
tic
[result,votes] = multisvm_WTA(Train_Feat,Train_Label,test,kernel);
exec_time=toc;
elseif svm==2
```

```
 tic
[result,votes] = multisvm_MWV(Train_Feat,Train_Label,test,kernel);
exec_time=toc;
elseif svm==3
    tic
[result,votes] = multisvm_DAG(Train_Feat,Train_Label,test,kernel);
exec_time=toc;
end
% display execution time
set(handles.edit3,'string',exec_time);
set(handles.edit21,'string',result);
set(handles.uitable1,'data',votes);
% disp(result);
% Visualize Results
if result == 1
     R1 = 'FANCY FULL YELLOW/(3/4) ';
    set(handles.edit2,'string',R1);
    helpdlg(' FANCY FULL YELLOW/(3/4) ');
    disp(' FANCY FULL YELLOW SLICE');
elseif result == 2
    R2 = 'FANCY HALF YELLOW';
    set(handles.edit2,'string',R2);
    helpdlg(' FANCY HALF YELLOW ');
    disp('FANCY HALF YELLOW SLICE');
elseif result == 3
    R3 = 'CHOICE';
    set(handles.edit2,'string',R3);
    helpdlg(' CHOICE ');
    disp('CHOICE SLICE  ');
elseif result == 4
    R5 = 'BROKEN';
    set(handles.edit2,'string',R5);
    helpdlg(' BROKEN ');
    disp(' BROKEN SLICE');
elseif result == 5
    R6 = 'REJECT';
    set(handles.edit2,'string',R6);
    helpdlg('REJECT ');
    disp(' REJECT SLICE ');
  end
% Update GUI
guidata(hObject,handles);
```

The classification of the images is displayed in a dialog box and text box as shown in figure below. Three multiclass SVMs (WTA, MWV, and DAG) are trained and tested using 5-fold cross validation and using the reduced feature vectors. Then, one choses the GRB (Gaussian Radial Basis) kernel, (HPOL) *d*th Homogeneous Polynomial kernel, and LIN (Linear) kernel as listed in Table 3.3. Hundreds of simulations are performed to determine the best kernel function parameters, such as the scaling factor γ in the GRB kernel and the order d in the HPOL kernel. The figure 4.13 show SVM classification results.



FANCY FULL YELLOW/(3/4)

FANCY FULL YELLOW/(3/4)

1

Figure 4.13: SVM classification Results in dialog box

The code used for 5-fold cross validation for performance is as follows:

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% Evaluate Accuracy
% clear all;
load('Training_Data.mat')
label=Train_Label.';
Train_Feat=zscore(Train_Feat);
[COEFF, SCORE] = pca(Train_Feat);
Train_Feat=SCORE(:,1:12);
itr = 5;
hWaitBar = waitbar(0,'Evaluating Max Accuracy using 5-Fold cross validation and 5
iterations');
for k = 1:itr
    indices = crossvalind('Kfold',label,5);
    cp = classperf(label);
    for i = 1:5
        test = (indices == i);
        TestingFoldSample = Train_Feat(test,:);
        N=size(TestingFoldSample,1);
        TrainingFoldSample = Train_Feat(indices ~= i,:);
        TrainingFoldLabel=label(indices ~= i,:);
        TrainingFoldLabel_R=TrainingFoldLabel.';
        kernel=handles.ImgData5;
        svm=handles.ImgData6;
        for j=1:N
            if svm==1
                [result,votes] =
multisvm_WTA(TrainingFoldSample,TrainingFoldLabel_R,TestingFoldSample(j,:),ker
nel);
                class(j,:)=result;
                %            set(handles.uitable1,'data',votes);
            elseif svm==2
                [result,votes] =
multisvm_MWV(TrainingFoldSample,TrainingFoldLabel_R,TestingFoldSample(j,:),ker
nel);
                class(j,:)=result;
                %            set(handles.uitable1,'data',votes);
            elseif svm==3
                [result,votes] =
multisvm_DAG(TrainingFoldSample,TrainingFoldLabel_R,TestingFoldSample(j,:),kern
el);
                class(j,:)=result;
                %            set(handles.uitable1,'data',votes);
            end
        end
        outlabel(indices==i,1)= class;
        classperf(cp,class,test);
    end
    % cp.ErrorRate
    class=[];
    waitbar(k/itr);
    Acc_F=cp.CorrectRate*100;
    Acc(k)=Acc_F;
    if Acc_F>=max(Acc)
        outlabel_max=outlabel;
    end
end
% Max_Acc2=sum(grp2idx(outlabel)==grp2idx(label))/length(label)
fold_correct_rate=Acc
Max_Acc=max(Acc);
set(handles.edit4,'string',Max_Acc);
C = confusionmat(outlabel_max,label);
set(handles.uitable2,'data',C);
outlabel_max=[];
delete(hWaitBar);
Sensitivity=cp.Sensitivity
Specificity=cp.Specificity
PPV=cp.PositivePredictiveValue
NPV=cp.NegativePredictiveValue
guidata(hObject,handles);
```

The performance of nine set of m-SVMs is tested and result tabulated as below in table 4.5 and 4.6. The two tables show the accuracy for classification and computation time in 5-fold cross-validation for those SVMs with optimized parameters, respectively. The research main objective accuracy is above 85%. Both MWV and WTA did achieve the objective and therefore recommended for use. MWV performed much better in all the kernels, and therefore the best method. The classification accuracies for both WTA SVM and MWV SVM, using LIN kernel, are above 85%, higher than the DAG SVM of 60.8%. WTA SVM and MWV SVM using the HPOL kernel have a classification accuracy of above 85%, compared to 60.4% for DAG SVM.

Finally, the classification accuracy of MWV SVM, using the GRB kernel, is 92.8%, slightly higher than WTA SVM of 88% and much higher than DAG SVM of 55.2%.

Therefore, the best results are attained using the Gaussian Radial Basis kernel MWV SVM with a classification accuracy of 92.8%.

Table 4.5: Classification Accuracy of SVMs

	LIN	HPOL	GRB
WTA-SVM	85.6%	85.2%	88%
MWV-SVM	87.2%	87.6%	**92.8%**
DAG-SVM	60.8%	60.4%	55.2%

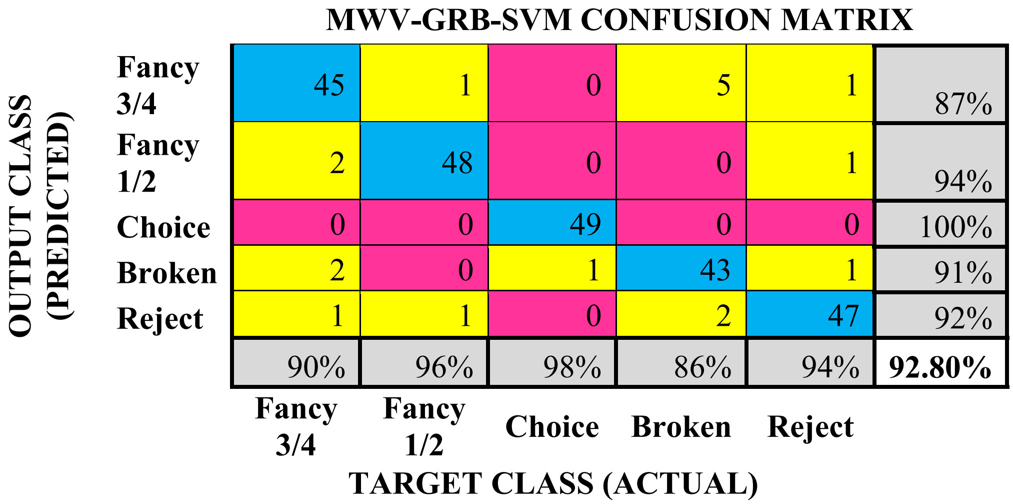
Table 4.6: Computation Time of SVMs

	LIN	HPOL	GRB
WTA-SVM	0.0788	0.0968	0.0493
MWV-SVM	0.0721	0.0668	0.0457
DAG-SVM	0.0202	0.0234	0.0210

The WTA SVM is the slowest in term of classification speed, the dataset needed for training is relatively large since it uses one-*versus*-all strategy. Since MWV SVM uses one-versus-one strategy, it is faster than WTA-SVM, for in MWV it is that every individual binary SVM only takes about 2/5 portion of the data. The DAG SVM is yet twice faster than MWV SVM. This is due to the fact that the MWV SVM requires all 10 individual SVMs to achieve a final conclusion, whereas the DAG SVM only requires 4 binary SVMs to perform the same task.

4.7 Confusion Matrix

Figure 4.14 shows the MWV SVM confusion matrix using the GRB kernel. Each column of the matrix represents the instances in the target class, whereas each row represents the instances in the output class (predicted class) (actual class). Samples whose target is the *j*th class that is classified as *i*th class are represented by the number in the *i*th row and *j*th column. Highlighted in yellow are all the misclassification cases.



MWV-GRB-SVM CONFUSION MATRIX

OUTPUT CLASS (PREDICTED)	Fancy 3/4	Fancy 1/2	Choice	Broken	Reject	
Fancy 3/4	45	1	0	5	1	87%
Fancy 1/2	2	48	0	0	1	94%
Choice	0	0	49	0	0	100%
Broken	2	0	1	43	1	91%
Reject	1	1	0	2	47	92%
	90%	96%	98%	86%	94%	**92.80%**

TARGET CLASS (ACTUAL)

Figure 4.14: MWV-GRB-SVM Confusion matrix with total classification accuracy of 92.8%.

By observing at the bottom line of Figure 4.14, one finds that none of the class is all recognized successfully. Nevertheless, only a few types of pineapple slices are misclassified. The SVM classification for the broken performs the worst i.e. 86%. In the database, there are 50 different pictures of Fancy ¾ class, however, two of them are misclassified as Fancy ½ class, two as Broken class and another one of them is misclassified as Reject class, so the rest 45 are classified correctly leading to a 90% correct rate. There are fifty samples in the database for the Fancy ½ class, but one are misclassified as Fancy ¾ class and one is misclassified as reject class, leaving 48 samples correctly classified, resulting in an 96% accuracy rate. There are fifty samples in the database for the Choice class, one of which is incorrectly classified as Broken class, leaving a 98% accuracy rating. In the Broken class, there are fifty samples in the test dataset, two of which are misclassified as Reject class and five of which are misclassified as Fancy ¾ class, resulting in an 86% accurate rate. With fifty samples in the database for the Reject class, one is misclassified as Fancy ½ class, one is misclassified as Fancy ¾ class, and another is misclassified as Broken class, resulting in a 94% accurate rate. A motivation for our future work is to solve above misclassification as all classes are observed not distinct in the view of SVM.

Table 4.7: The confusion matrices for the nine SVM models

LINEAR KERNEL				
WTA-LIN-SVM CONFUSION MATRIX (85.6%)				
39	7	0	2	1
8	37	0	2	1
0	4	48	1	1
2	1	1	44	1
1	1	1	1	46
MWV-LIN-SVM CONFUSION MATRIX (87.2%)				
37	8	0	3	1
8	42	1	1	1
0	0	48	1	0
4	0	1	45	2
1	0	0	0	46
DAG-LIN-SVM CONFUSION MATRIX (60.8%)				
45	40	18	5	1
1	10	18	0	2
0	0	12	0	0
4	0	2	39	1
0	0	0	6	46

4.7 (a): Linear Kernel

HOMOGENEOUS POLYNOMIAL KERNEL				
WTA-HPOL-SVM CONFUSION MATRIX (85.2%)				
37	4	0	2	1
10	41	0	1	3
0	4	49	2	1
3	1	0	44	3
0	0	1	1	42
MWV-HPOL-SVM CONFUSION MATRIX (87.6%)				
38	4	0	7	2
6	45	1	0	0
0	0	48	1	1
5	0	1	42	1
1	1	0	0	46
DAG-HPOL-SVM CONFUSION MATRIX (60.4%)				
45	41	21	3	2
1	8	13	0	0
0	0	13	0	0
2	0	3	39	2
2	1	0	8	46

4.7 (b): Homogenous Kernel

The table 4.7 shows the confusion matrices for the 9 SVM models. The leading diagonal gives the accuracy of the class. i.e 92.8% for MWV-GRB-SVM as (45+48+49+43+47)/250

GAUSSIAN RADIAL BASIS KERNEL				
WTA-GRB-SVM CONFUSION MATRIX (88%)				
39	5	0	2	1
6	43	1	2	1
0	1	48	0	1
4	0	1	44	1
1	1	0	2	46
MWV-GRB-SVM CONFUSION MATRIX (92.8%)				
45	1	0	5	1
2	48	0	0	1
0	0	49	0	0
2	0	1	43	1
1	1	0	2	47
DAG-GRB-SVM CONFUSION MATRIX (55.2%)				
45	40	32	4	0
2	9	13	0	1
0	0	2	0	0
1	0	3	33	0
2	1	0	13	49

4.7 (c): Gaussian Radial Basis Kernel

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The main goal of the research is to develop a pineapple slices classification algorithm using a multi-class SVM and hybrid feature extraction technique to automate the manual of sorting and grading of slices. At the end of the research work, through experiments, the following conclusions arrived at:

i). An Otsu's segmentation algorithm has been developed and preprocessed high quality images for classification. The images delivered clearly identifiable features to the SVMs classify and enabled classification of the pineapple slices into five categories.

ii). An improved SVM recognition performance has been achieved by use of a hybrid feature extraction method. The combination of colour moment, shape, colour histogram and GLCM texture features, are found to be more effective than any single kind of feature in classification of pineapple slices.

iii). PCA algorithm has been used to reduce dimension of feature space from 90 features to 12 features and maintain 99% energy as selection standard. By working with 12 features, the test and training speed are accelerated and at the same time removing irrelevant extra features improved the classification accuracy.

iv). Using PCA reduced features, three different multiclass SVMs (DAG-SVM, MWV-SVM, and WTA-SVM) have been built and tested through 5-fold cross validation and with three kernels; Gaussian Radial Basis, *d*th Homogeneous Polynomial, and linear in the database of 250 pineapple slice images. The highest overall classification accuracy results of about 93% was achieved.

v). The pineapple slices grading and sorting method develop achieved accuracy of about 93% which is a higher accuracy than J. I. Asnor, 2013, pineapple classification using RGB colour that achieved 75% accuracy.

In summary, a pineapple slices classification algorithm with accuracy of above 85% is developed and tested. The experimental results showed that the MWV-GRB-SVM achieves classification accuracy of about 93% which is the highest accuracy overall.

5.2 Recommendations

The research recommends the following:

i). Take in additional features to increase the classification accuracy, fractal dimension (FD), more textures and many more;

ii). Refine algorithm to find distinguishable features for Fancy ¾ and broken;

iii). Build hardware prototype to implement the algorithm;

iv). Find ways to accelerate the algorithm, like using modified Gaussian kernel that run fast hence accelerating algorithm.

v). Extending this research to multiple fruits sale at supermarket.

vi). From the research it was found that it’s challenging to calibrate the classifier daily to capture daily variabilities of pineapple slices received from plantation. From this a paper was published to solve the problem.

REFERENCES

Acevedo, J. R., Maldonado, S. B., Lafuente, S. A., Siegmann, P., & López, F. F. (2009). Computational load reduction in decision functions using support vector Machines. *Signal Process*, 89, 2066–2071.

Armand, S., Watelain, E., Roux, E. M., & Lepoutre, F.-X. (2007). Linking clinical measurements and kinematic gait patterns of toe-walking using fuzzy decision trees. *Gait Posture*, 25, 475–484.

Bangare, S., Dubal, A., Bangare, P., & Patil, S. (2015). Reviewing Otsu's Method For Image Thresholding. *International Journal of Applied Engineering Research*, 10. 21777-21783.

Benazir, K., & Vijayakumar. (2012). Fingerprint Matching by Extracting GLCM Features. *International Conference & Workshop on Recent Trends in Technology (TCET), International Journal of Computer Applications (IJCA)* (pp. 30-34). New York: Association for Computing Machinery.

Davies, E. R. (1984). Design of cost-effective systems for the inspection of certain food products during manufacture. *Proceedings of the 4th Conference on Robot Vision and Sensory Controls, 9–11 October* (pp. 4. 37–46). London: In Pugh, A. (ed.).

Davies, E. R. (2012). *Computer and Machine Vision: Theory, Algorithms, Practicalities, 4th edition,.* Oxford, UK: Academic Press.

Deris, A. M., Zain, A. M., & Sallehuddin, R. (2011). Overview of Support Vector Machine in Modeling-Machining Performances. *Procedia Eng.,*, 24, 308–312.

Diwash, P., Ankit, B., P., S., & S., Y. (2022, September 1). *HSV Selector for Stickers and Crown Detection.* Retrieved from researchgate: https://www.researchgate.net/publication/363539471_HSV_Selector_for_Stickers_and_Crown_Detection

Eleyan, A., & Irel, H. D. (2011). Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turk J Elec Eng & Comp Sci*, 19(1):97-107.

Gagan, B. (2022). Topic modeling by using lda. *Interantional Journal of scientific research in engineering and management*, 06.

Gill, H., Khalaf, O., Alotaibi, Y., Alghamdi, S., & Alassery, F. (2022). Fruit Image Classification Using Deep Learning. *Computers, Materials & Continua*, 71. 5135-5150.

Gosselin, B., Kleynen, O., Leemans, V., Destain, M. F., & Unay, O. D. (2010). Automatic grading of Bi-coloured apples by multispectral machine vision. *Computers and Electronics in Agriculture*.

Haralick, R., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions On Systems, Man and Cybernetics*, SMC-3(6):610-21.

Heinemann, P., Varghese, Z. A., Morrow, C. T., Sommer III, H. J., & Crasswelle, R. (1995). Machine Vision Inspection of 'Golden Delicious' Apples. *Applied Engineering in Agriculture*, 11(6),901-906.

Hou, Z., & Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235. 3–35.

Hou, Z., & Wang, Z. (2016, September 1). *Information Sciences-2013 From model-based control to data-driven control.* Retrieved from Researchgate: https://www.researchgate.net/publication/307959874_Information_Sciences-2013_From_model-based_control_to_data-driven_control

Jackson, J. E. (1991). *A User's Guide to Principal Components.* Hoboken, NJ, USA: John Wiley & Sons.

Jiawei, H., & K. M. (2006). Data mining: concepts and techniques. *morgan kaufmann*, 54.

Jiménez Carvelo, A., Cruz, C., Cuadros-Rodríguez, L., & Koidis, T. (2022). Machine learning techniques in food processing. In A. P. Ayon Tarafdar, *Current Developments in Biotechnology and Bioengineering: Advances in Food* (pp. 333-351). Amsterdam : Elsevier.

Juliana, G., Florencia, V., & Laura, C. (2020). *OTSU's thresholding.* Retrieved from Researchgate: https://www.researchgate.net/publication/345685905_OTSU%27s_thresholding

Kwak, N. (2008). Principal Component Analysis Based on L1-Norm Maximization. *IEEE Trans. Patt. Anal. Mach. Int*, 30, 1672–1680.

Li, D., Yang, W., & Wang, S. (2010). Classification of foreign fibers in cotton lint using machine vision and multi-class support vector machine. *Comput. Electron. Agric*, 74, 274–279.

Lind, R., & Murhed, A. (2012). Computer vision in food processing: an overview. *Computer vision technology in the food and beverage industries*, 130-149.

Lingras, P., & Butz, C. (2007). Rough set based one-v-one and one-v-r approaches to support vector machine multi-classification. *Inform. Sci.*, 177, 3782–3798.

Lipovetsky, S. (2009). PCA and SVD with nonnegative loadings. *Pattern Recognit. Lett.,* , 42, 68–76.

Lou, S., Jiang, X., & Scott, P. (2012). Algorithms for morphological profile filters and their comparison. *Precis. Eng.*, 36, 414–423.

Maddipati, S., Nandigam, R., K. S., & Venkatasubramanian, V. (2011). Learning patterns in combinatorial-protein libraries by Support Vector Machines. *Comput. Chem.Eng.*, 35, 1143–1151.

Maitra, M., & Chatterjee, A. (2008). A hybrid cooperative comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Syst. Appl.*, 34, 1341–1350.

Mathar, R., Alirezaei, G., Balda, E., & Behboodi, A. (2020). Support Vector Machines. In R. Mathar, *Fundamentals of Data Analytics, With a View to Machine Learning* (pp. 83-105). Cologne: Springer; 1st ed. 2020 edition.

May, R., Maier, H., & Dandy, G. (2010). Data splitting for artificial neural networks using SOM-based stratified sampling. *Neural Netw*, 23, 283–294.

Nakano, V. (1997). Application of neural networks to the colour grading of apples. *Computers and Electronics in Agriculture, Elsevier*, 105-116.

Narendra, V. .., & Hareesh, K. S. (2010). Prospects of computer vision automated grading and sorting systems in agricultural and food products for quality evaluation. *Int J Comp App*, 1: 0975 – 8887.

Nnamdi, E., Vincent, C., Eneh, N., Ogechukwu, I., & Ijeoma, E. A. (2022). Segmentation of Lung Nodules in Computed Tomography Images Using Modi ed Otsu's Technique. *SUSTAINABLE ENGINEERING AND INDUSTRIAL TECHNOLOGY* (pp. J9,1-6). NSUKKA NIGERIA: UNIVERSITY OF NIGERIA.

Noah, K. (2005). Colour moments. *School Of Informatics, University Of Edinburgh*, 3-6.

Norton, O. I., Ozkan, D., Mert, U. Y., & Senturk, O. (2008). Estimation of tree size diversity using object–oriented texture analysis and ASTER imagery. *sensors*, 8:4709-4724.

Patil, N. S., Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2005). Regression models using pattern search assisted least square support vector machines. *Chem. Eng. Res. Des.*, 83, 1030–1037.

Patil, N. S., Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2005). Regression models using pattern search assisted least square support vector machines. *Chem. Eng. Res. Des.*, 83, 1030–1037.

Pereira, A., Reis, M., Saraiva, P., & Marques, J. (2011). Madeira wine ageing prediction based on different analytical techniques. *UV–vis, GC-MS, HPLC-DAD. Chemometr. Intel. Lab. Syst.*, 105, 43–55.

Platt, J. C., Cristianini, N., Shawe, J. T., & Large, M. (2000). DAGs for multiclass classification. *Adv. Neural. Inform. Process. Syst.*, 12, 547–553.

Rashmi, P., Sapan, N., & Roma M. (2013). Image Processing and Machine Learning for Automated Fruit Grading System: A Technical Review. *International Journal of Computer Applications*, 0975 – 8887.

Senthilkumaran, N., & Rajesh, R. (2009). Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches. *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, May.

Siang, K. T., & Mat, N. A. (2011). Colour image segmentation using histogram thresholding—Fuzzy C-means hybrid approach. *Pattern Recognit. Lett*, 44, 1–15.

Sindhuri, M., & Nallapareddy, A. (2022). Text Separation in Document Images through Otsu's Method. *EEE WiSPNET 2016 conference* (pp. 2395-2399). Chennai : Anusha Nallapareddy.

Sun, D. W., & Brosnan, T. (2003). Improving quality inspection of food products by computer vision: a review. *J Food Engineering*, 61: 3-16.

Tang, W., & Daoutidis, P. (2022). Data-Driven Control: Overview and Perspectives. *American Control Conference (ACC)* (pp. 1048-1064). Atlanta: IEEE.

Wangwe, S. (1995). *Exporting Africa: technology, trade and industrialization in Sub-Saharan Africa.* New York: United Nations University/Routledge.

Xu, Y., Li, L., Li, X., & Deng, Y. H. (2013). A feature-selection algorithm based on Support Vector Machine-Multiclass for hyperspectral visible spectral analysis. *Journal of Food Engineering*.

Yudong, Z., & Lenan, W. (2012). Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine. *Sensors*, 1424-8220.

Zayas, I., Pomeranz, Y., & Lai, F. S. (1989). Discrimination of wheat and non-wheat components in grain samples by image analysis. *Cereal Chemistry*, 66: 233- 237.

Zheng, H., & Lu, H. (2012). A least-squares Support Vector Machine (LS-SVM) based on fractal analysis and CIELab parameters for the detection of browning degree on mango. *Computers and Electronics in Agriculture*, 83: 47-51.

Zou, X. B., Zhao, J. W., & Li, Y. M. (2010). In-line detection of apple defects using three colour cameras system. *Computers and Electronics in Agriculture*, Vol.70:129-134.

APPENDICES

Appendix 1: List of Journal Publications and Conference Papers

1. John N. Kamau, “A Step-By-Step Approach to Retrofit and Automation of PCB Machines Using PLC”, European International Journal of Science and Technology, Vol. 7, No. 3, May 2018.
2. J.N. Kamau, “A Step-By-Step Approach to Retrofit and Automation of PCB Machines Using PLC”, Proceedings of the Sustainable Research and Innovation Conference, held at JKUAT Main Campus, Kenya, on 2 - 4 May, 2018
3. J.N. Kamau, P.K. Hinga, S.I. Kamau, “Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification” Published in International Research Journal of Innovations in Engineering and Technology - IRJIET, Volume 6, Issue 9, pp 1-8, September 2022. Article DOI

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 9, pp 1-8, September-2022

https://doi.org/10.47001/IRJIET/2022.609001

Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification

[1]*J.N. Kamau, [2]P.K. Hinga, [3]S.I. Kamau

[1,2,3]Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, P.O. Box 62000 – 00200 Nairobi, Kenya
Authors Email: [1]*jnyutu@gmail.com, [2]pkhinga@gmail.com, [3]skamau@eng.jkuat.ac.ke

Abstract **- Sorting pineapple can be automated with use of computer vision. The unique challenge with the pineapple slices is variability of the fruit slices color, ripeness and texture due to varying environmental parameters and fruit types. The most common types of pineapple fruit are smooth Caen and MD2. Currently the pineapple industries sort the slices manually using casual workers. Before commencement of a typical production shift, there is start-up shift where machine are cleaned, prepared and calibrated for the production. Fruit slices are also sampled and processed to simulated actual production. A mock sorting is done to help guide the worker for the expected sorting for the five categories i.e: fancy ¾, fancy ½, choice, broken and reject. To achieve a fully automated sorting process there is a need to calibrate machine model and capture the day to day variability of fruit color, ripeness and texture. In this paper we propose to use an analytical method to calibrate the Support Vector Machine (SVM) with Gaussian radial basis function (RBF) for optimal sigma and box constraint (C). A compelling feature of the proposed algorithm is that it does not require an optimization search, making the selection process simpler and more computationally efficient. The proposed algorithm achieves the highest accuracy when used with the Gaussian multiclass SVM, as demonstrated by experimental results on three real-world datasets.**

Keywords: Gaussian Radial Basis Function, Sigma, Support Vector Machine, Class Separability, Computer Vision, Box Constraint.

I. DEFINITIONS

Notation	**Meaning**
C	Box contraint/ regularization parameter in SVM
ω	the weight vector in two dimensions for W and B
$\kappa(\mathbf{x}i,\mathbf{x}j)$	the kernel function
$\Phi(\mathbf{x}i)$	the kernel-space instance
W, B	separability within and between classes, accordingly
W', B'	The mean distance inside and between classes, respectively.
σ	Sigma parameter for kernel function

II. INTRODUCTION

Calibration is the process of configuring an instrument to produce acceptable results for a sample. Fundamental to the design of instruments is the elimination or minimization of factors that lead to inaccurate measurements. The purpose of pineapple classifier calibration is to reduce classification error by ensuring the precision of test equipment. Calibration quantifies and controls classification process errors and uncertainties to an acceptable level.

A human operator sorts the sample of pineapple slices into five categories i.e. fancy ¾, fancy ½, choice, broken and reject. The sample size holds at least 50 slices of each category. The samples are then passed into a machine to capture the day's pineapple slices variability. The machine is designed to accept the first 50 slices as the fancy ¾, the second 50 slices as fancy ½, then choice, broken and reject in that order. The machines then run the proposed algorithm to determine the optimal sigma and regularization parameters for the day. Here we describe how to develop the calibration algorithm.

The evaluation of SVM performance is a crucial step in ML design: precision, CPU time, and consistency. The kernel parameter is crucial for maintaining the SVM's high performance. Since's domain of definition ranges from zero to infinity, exhaustive search for parameter selection of is intractable. For instance, if the parameter is close to zero, SVM tends to over fit because all training instances are used as support vectors. Under-fitting occurs in SVM if the parameter tends toward infinity because all training instances are treated as a single instance. The vast majority of prior research on this topic is based on optimization search algorithms that result in a high computational load and are extremely slow. A simple but effective analytical algorithm is proposed for locating a good value of σ.

In the field of machine learning, the support vector machine (SVM) is a critical technique for supervised learning. SVM utilizes the principle of structural risk minimization to locate an optimal hyper plane in which training instances of distinct classes are linearly separable. Due to its numerous desirable properties and promising empirical performance [i, ii], SVM quickly gained attention from researchers who applied it to a variety of fields, including science and engineering, such as condition monitoring and fault diagnosis [iii,iv]. Among the kernels available in SVM, the RBF kernel is the most widely used due to its attractive properties [1, 2], *e.g.* the property of structure-preserving. The Gaussian RBF kernel has a structure similar to that of $k(X_i, X_j) = \exp(-\frac{\|X_i - X_j\|^2}{2\sigma^2})$ where σ is the only parameter named by width of features.

We propose an algorithm for determining the optimal σ based on the maximization of between-class separability and minimization of within-class separability, and calibrate our classifier for optimal pineapple classification. An attractive feature of the proposed algorithm is that it does not require an optimization search, making the selection procedure simpler and more computationally efficient. Since the maximizer can be derived analytically, the proposed method avoids the optimization search process, resulting in a significant improvement in computation load for parameter selection. After the optimal σ has been determined, searching for the box constraint parameter is a simple iterative process. The experimental results on three real-world datasets indicate that the proposed algorithm provides the highest accuracy for the Gaussian multiclass SVM.

III. REPORTED WORK

3.1 Manufacturing Calibration

Manufacturing calibration ensures the precision and consistency of measuring/classification instruments by comparing them to reference calibrating equipment and making any necessary adjustments. The primary significance of calibration is that it preserves classification precision, standardization, and repeatability, thereby ensuring the reliability of benchmarks and results.

Without regular calibration, equipment can deviate from specifications, provide inaccurate classification, and jeopardize quality, safety, and longevity[v]

In terms of the quality and performance of a procedure or finished product, manufacturing precision is a crucial metric.

If the pineapple slices are not classified to the customer's required specifications, which are determined by the product's intended use, there will likely be negative consequences.

The significance of precision in manufacturing highlights the significance of calibration, as manufacturing equipment must be properly calibrated to meet specifications. Without properly calibrated equipment, it is impossible to meet ISO quality standards and achieve the required level of accuracy.

The quality and safety of the finished product is the most essential reason why precision is a crucial manufacturing standard. Inaccurate components, or those that fall outside of the required tolerance, cannot be used because their likelihood of functioning properly is extremely low.

Calibration plays a significant role in preventing the production of inaccurate products; therefore, the benefits of calibrating your machinery and measuring equipment cannot be overstated.

There are numerous facets of calibration in the manufacturing industry, but calibration is generally significant in these two key areas:

- Calibration of classification equipment ensures that your quality control processes are precise and that you do not accept defective slices.
- Calibration creates a more efficient manufacturing process by ensuring that equipment is operating as expected. Inadequately calibrated equipment will lead to unpredictable manufacturing outcomes and inaccurate slices classification.

Here the SVM classifier is a virtual machine, developed to sort pineapple slices into five categories. The two parameters of Gaussian kernel required to be calibrated are sigma and regularization parameter. In this paper we propose to use method that entirely uses derived formulae to arrive to optimal parameter without rigorous iterative process.

3.2 Gaussian radial basis function (RBF) kernel

Without knowing the mapping function Φ explicitly, kernel methods map data from the feature space to the kernel space.

Using the kernel method, SVM can locate a hyper plane in the kernel space, resulting in a non-linear separation of features in the feature space.

The Gaussian radial basis function (RBF) kernel is one of the most widely used kernels in SVM due to its attractive characteristics, such as the structure-preserving property. The kernel of the Gaussian RBF has the form $k(X_i, X_j) = \exp(-\frac{\|X_i - X_j\|^2}{2\sigma^2})$where σ is the only specified parameter by width of features. Sigma (σ) is determined by a default value, such as σ=1.It has been reported, however, that is crucial for the robust performance of SVM, whereas an arbitrary value of

ISSN (online): 2581-3048

Volume 6, Issue 9, pp 1-8, September-2022

https://doi.org/10.47001/IRJIET/2022.609001

σ cannot guarantee performance. Grid search is a simple and intuitive method. By defining a finite set, grid search applies a criterion to each possible solution (node) within the set. As the optimal value of σ, the node with the highest score on the criterion is selected. The grid search strategy is adopted, and the Support Vector Machine's (SVM) classification accuracy is typically used as the selection criterion. Grid search has two shortcomings;

1. CPU time increases exponentially with the number of nodes in the set, making it a time-consuming operation;
2. The optimal σ cannot be determined if the set is improperly defined.

This could occur due to a lack of prior knowledge. To determine the optimal value σ of, intelligent optimization techniques such as simulated annealing algorithm[vi], genetic algorithm [vii],gradient descent algorithm[viii]and particles warm optimization algorithm were utilized[ix]. Typically, classification accuracy is regarded as the objective function. However, the classification accuracy of SVM is affected by other parameters, such as the regularization parameter, in addition to σ. Li's method [x] identifies the optimal σ using gradient search. Using intelligent optimization search algorithms, the reviewed parameter selection methods may require less computation time than grid search. Nevertheless, they increase the complexity of selection algorithms, which is likely why the parameter σ in numerous applications is frequently set to a default value. To improve the efficiency of the selection procedure, a simple but effective analytical algorithm for determining a good value σ of is proposed in the present work. Introducing both within-class and between-class separability, we define the objective function of class separability. This class separability measure is in fact a function with respect to the parameter σ [xi].

3.3 Max Wins Voting (MWV) SVM

Initially, SVMs were designed for binary classification. Numerous methods for multi-class SVMs have been proposed, the most prevalent of which is to divide the multiclass problem into multiple binary classification problems [xii]. Three types of methods are frequently used: Winner-Takes-All (WTA) SVM, Max-Wins-Voting (MWV) SVM, and Directed Acyclic Graph (DAG) SVM. MWV has the highest accuracy in image classification among the three methods [xiii] and we select this method to optimize its accuracy.

Classification is accomplished using a strategy for the one versus one method in max wins voting (MWV). After constructing a binary SVM for each pair of classes, one will obtain C(C-1)/2 binary SVMs in total. When applied to new test data, each SVM assigns one vote to the winning class, and the test data is labeled with the class with the most labels. If there are two identical votes, MWV chooses the class with the smallest index. The following is the mathematical formula. The *ij*th (*i*= 1,2, …, *C*-1, *j* = *i*+ 1, …, *C*) individual binary SVM is trained with all data in the *i*th class with +1 label plus all data of the *j*th class with −1 label, so as to distinguish *i*th class from *j*th class. The decision function of *ij*th SVM is:

$$f_{ij}(x) = \sum_{n=1}^{N_1+N_j} y_n^{ij} \alpha_n^{ij} k(x_n^{ij}, x) - b_{ij}, i = 1,2, \dots, C-1, j = i+1, i+2, \dots, C$$

$$y_n^{ij} = \begin{cases} +1 & x_n^{ij} \in ith\ class \\ -1 & x_n^{ij} \in jth\ class \end{cases}$$

WhereN_i and N_j denotes the total number of *i*th class and *j*th class, respectively.$y_n^{ij} \in \{+1, -1\}$ depends on the class label of x_n^{ij}. If x_n^{ij} belongs to*i*th class, $y_n^{ij} = +1$; otherwise x_n^{ij} belongs to *j*th class, $y_n^{ij} = -1$. α_n^{ij} is the Lagrange coefficient; and b_{ij} is the bias term. α_n^{ij} and b_{ij} are obtained by training the *ij*th individual SVM. The output of *ij*th SVM is the sign function of its decision function, namely:

$$O_{ij}(x) = sgn\left(f_{ij}(x)\right)$$

if $f_{ij}(x) > 0$, then the output $O_{ij}(x)\ is + 1$, denoting x belongs to *i*th class; otherwise output is -1, denoting *x*belongs to *j*th class.

IV. METHODOLOGY

To develop an analytical algorithm to determine the optimal σ and C and achieve the best calibration accuracy for Gaussian m-SVM, the following steps are used:

- Define the objective function, which is a class separability function with respect to the parameter σ.
- Analytically define the maximizer of the objective function
- Select the weight vector through intuitive and simple method.
- Develop a MATLAB algorithm implementing the maximizer function with selected weights.
- Test the code on three real world datasets and evaluate their prediction performance through 5-fold cross validation

4.1 Defining the objective function

Gramian matrix (also known as kernel matrix) is a matrix that contains all the dot product values of a training subset. SVM training relies on the dot product. In other words, the Gramian matrix contains all information that the SVM can learn about training instances, along with the label information.

Given a data set U and a kernel function, the Gramian matrix is denoted by

$$G = \begin{bmatrix} K_{11} \ldots K_{1L} \\ \vdots \quad \ddots \quad \vdots \\ K_{L1} \ldots K_{LL} \end{bmatrix} = \exp\left(-\frac{12}{\sigma^2} D\right), D = \begin{bmatrix} K'_{11} \ldots K'_{1L} \\ \vdots \quad \ddots \quad \vdots \\ K'_{L1} \ldots K'_{LL} \end{bmatrix}$$

D is known as the Euclidean distance matrix. The Gramian matrix has a relationship with both σ and D. Since D is fixed for a dataset, the only parameter that can be modified is σ. Class separability is a traditional concept for describing the scattering of instances in the feature space. Class separability takes the following two principles into account:

1. *Principle I*: Instances of the same class ought to be as comparable as possible;
2. *Principle II*: Different class instances should be as dissimilar as possible.

Typically, the within class separability (W) and the between-class separability (B) are used to evaluate adherence to these two principles. We first define two distance-based scalars to estimate W and B in the feature space. The relationship between the distance similarity in the feature space and that in the kernel space is depicted in the following equation [7].

$$\left\|\Phi(X_i) - \Phi(X_j)\right\|^2 = 2 - 2\exp\left(-\frac{\left\|X_i - X_j\right\|^2}{2\sigma^2}\right)$$

Given this relationship, two corresponding scalars are derived to estimate W and B in the kernel space. The optimal σ is the one that simultaneously minimizes W and maximizes B in the kernel space. It is assumed that datasets are Gaussian distributed so that the mean distance can be used to accurately estimate class separability. The following relationship exists between the within-class mean distance (W'), the between-class mean distance (B'), and the total mean distance (T') in the feature space:

$$T' = \left(\sum_{i=1}^{L} N_i^2/N^2\right) W' + \left(1 - \sum_{i=1}^{L} N_i^2/N^2\right) B'$$

W and *B* in the kernel space can be estimated respectively using:

$$W = 2 - 2\exp\left(-\frac{12}{\sigma^2} W'\right)$$

$$B = 2 - 2\exp\left(-\frac{12}{\sigma^2} B'\right)$$

The objective function of class separability is established by

$$J(\sigma) = \omega^T \begin{bmatrix} -W \\ B \end{bmatrix}$$
$$= \omega_w \left(2\exp\left(-\frac{12}{\sigma^2} W'\right) - 2\right)$$
$$+ \omega_B \left(2 - 2exp\left(-\frac{12}{\sigma^2} B'\right)\right)$$

Where ω, ω = [ω_W, ω_B]^T, is the weight vector with a constraint of ω_W+ ω_B= 1. And we consider cases of W' < B' to be distinguishable while the other case is not.

4.2 Defining the maximizer of objective function

The optimal is the one that maximizes class separability, or the maximizer of the objective function that is twice differentiable. If the first derivative of J(σ) is equal to zero and the second derivative is negative, the maximizer is determined. The stationary point in the following equation is the maximizer and optimal σ we are seeking.

$$\sigma^* = \sqrt{\frac{B' - W'}{2 \times log\left(\omega_B B'/\omega_W W'\right)}}$$

4.3 Selecting the weight vector

The proposed method proposes that the Gramian matrix is obtained by transforming the Euclidean matrix. A dataset's Euclidean matrix is fixed. The only factor that determines this transformation is the parameter σ. The selection of weight ω depends on the situation.

In this section, we provide two simple and intuitive options for selecting the weight vector. First, we must identify the constraints of ω in maximize Eq. Because of the application condition of W'< B', the denominator in the square root must be positive. In addition to the constraint ω_W + ω_B = 1, the constraints for two elements ω of are as follows:

$$0 < \omega_w < \frac{B'}{B'+W'} \qquad \frac{B'}{B'+W'} < \omega_b < 1$$

If we choose ω_W= ω_B= 0.5, it is clear that the two conditions hold. Under this selection, the optimal *σ* is calculated by:

$$\sigma_1^* = \sqrt{\frac{B' - W'}{2 \times log\left(B'/W'\right)}}$$

If we choose ω_W= *W*'/(*W*' + *B*') and ω_B= *B*'/(*W*' + *B*'), the two conditions are also satisfied. The optimal *σ* is calculated by:

$$\sigma_2^* = \sqrt{\frac{B' - W'}{4 \times log\left(\frac{B'}{W'}\right)}}$$

ISSN (online): 2581-3048
Volume 6, Issue 9, pp 1-8, September-2022
https://doi.org/10.47001/IRJIET/2022.609001

4.4 Developing a MATLAB algorithm

The algorithm was developed on the MATLAB 2018a (The Mathworks ©) platform. A function is coded using Euclidean Matrix Sum to return as follows:

function [optSigma, B_bar, W_bar] =OptimalSigma(feaVal,species,method)

The code is then tested on three real world datasets and evaluate their prediction performance through 5-fold cross validation to get optimal values for calibrating the classifier.

Optimal parameters are extracted with following steps:

Our first step on optimal accuracy selection is to select the database and load it in to the memory. There are three dataset to select: Ionosphere, Fisher iris and Pineapple Slices. The database details i.e. Number of classes, number of features and numbers of instances are displayed.

Our second step is to selected method of approach. Chose mode between default, proposed method 1 and proposed method 2. The method chosen is displayed on the text box.

Our third step is to run the objective function and generate the optimal parameters. These are sigma, between class separability (B), within class separability (W) and ration of B/W which supposed to be above one if classes are separable.

Our final step is to test for accuracy where test and training accuracy are evaluated through 5 fold cross validation and the best box constraint evaluated. The algorithm also output confusion matrix for the best class that yielded optimal box constraint using the optimal sigma.

The last step is to click Exit command or run for another database where from first to final steps.

After obtaining optimal parameter, the figure 1 below shows system overview.

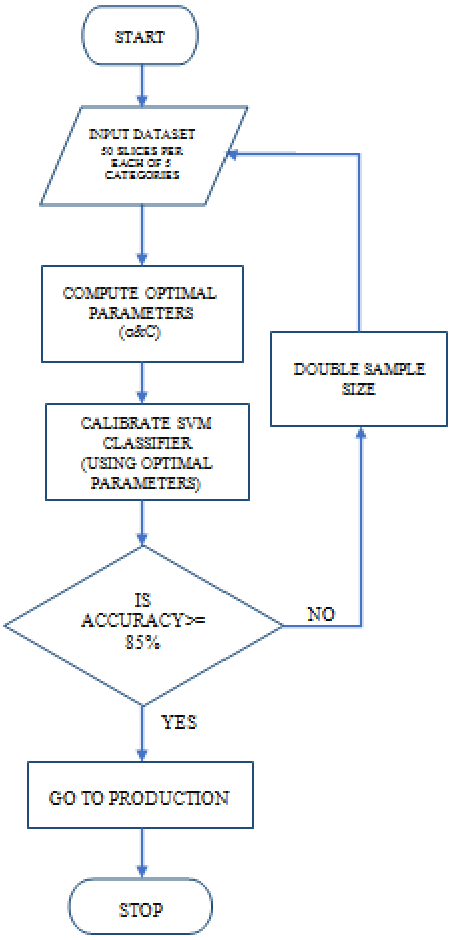
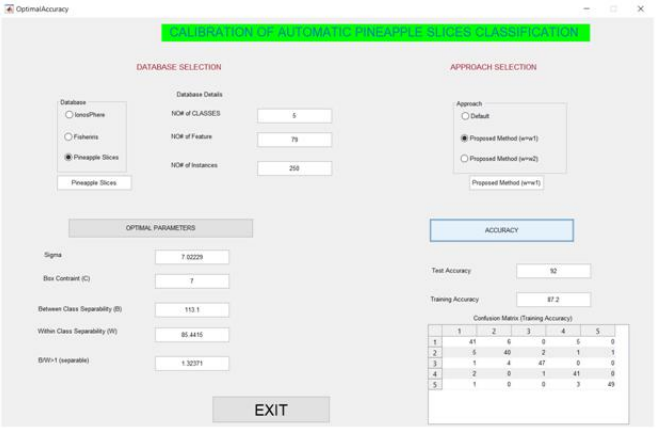

START
INPUT DATASET 50 SLICES PER EACH OF 5 CATEGORIES
COMPUTE OPTIMAL PARAMETERS (σ&C)
CALIBRATE SVM CLASSIFIER (USING OPTIMAL PARAMETERS)
IS ACCURACY>= 85%
NO
DOUBLE SAMPLE SIZE
YES
GO TO PRODUCTION
STOP

Figure 1: System flow diagram

The figure 2 below is the Graphical User Interface (GUI) of the experiment.



Database
○ IonosPhere
○ Fisheriris
● Pineapple Slices

Pineapple Slices

Database Details

NO# of CLASSES: 5

NO# of Feature: 79

NO# of Instances: 250

Approach
○ Default
● Proposed Method (w=w1)
○ Proposed Method (w=w2)

Proposed Method (w=w1)

Sigma: 7.02229

Box Contraint (C): 7

Between Class Separability (B): 113.1

Within Class Separability (W): 85.4415

B/W>1 (separable): 1.32371

Test Accuracy: 92

Training Accuracy: 87.2

Confusion Matrix (Training Accuracy)

	1	2	3	4	5
1	41	6	0	5	0
2	5	40	2	1	1
3	1	4	47	0	0
4	2	0	1	41	0
5	1	0	0	3	49

Figure 2: The Experiment GUI

V. RESULT AND DISCUSSION

This section compares the classification accuracy of two selection methods, the default and proposed method. Following is a description of the two approaches. The first method specifies σ and C with their default values, σ= C = 1. The proposed method represents the second strategy. Using two distinct subsets, we assess the proposed method of **ω**: **ω**1 = [0.5, 0.5]^T; **ω**2 = [*W*'/(*W*' +*B*'), *B*'/(*W*' + *B*')]^T.A SVM classifier is used to evaluate the performance of the two approaches. Three real-world datasets were used to evaluate the two approaches.

Table 1: Summary of the three Datasets

No	Dataset	Number of Classes	Number of Features	Number of Instances
1	Ionosphere	2	34	351
2	Fisheriris	3	4	150
3	Pineapple Slices	5	79	250

The classification accuracy is used to compare the performance of the two approaches. The classification accuracy is defined as Nc/(Nc+Nf)100%, where Nc represents the number of instances correctly classified and Nf represents the number of instances incorrectly classified. The most important performance indicator is accuracy. Due to the fact that the parameter C (box constraint) affects classification accuracy, it is necessary to select C in order to evaluate the two approaches accurately. In the first approach, the default value C is specified.

The second method for C selection combines grid search with σ selection. Table 2 summarizes the selected values of σ and C.

Table 2: Selected parameters for three datasets

Dataset	Approach		σ	C	B'	W'	B'/W'
Ionosphere	Default		1	1	78.1863	55.2614	1.41484
	The proposed Method	$\omega = \omega_1$	5.74727	10			
		$\omega = \omega_2$	4.06393	10			
Fisheriris	Default		1	1	10.8171	2.20571	4.90416
	The proposed Method	$\omega = \omega_1$	1.64556	1			
		$\omega = \omega_2$	1.16358	1			
Pineapple Slices	Default		1	1	90.0723	67.9907	1.32477
	The proposed Method	$\omega = \omega_1$	6.26557	7			
		$\omega = \omega_2$	4.43043	10			

Once the optimal values for σ and C have been determined, the SVM model is trained on the training subset and hence calibrated for optimal performance. Each dataset is subjected to twenty independent iterations of each methodology. Using K-fold cross-validation (K = 5), the training and test accuracy for every run is estimated. The results are provided in Table 3.

Table 3: Experimental results of the three Dataset

Dataset	Method		Test Accuracy	Training Accuracy
Ionosphere	Default		68.5714	77.1429
	The proposed Method	$\omega = \omega_1$	94.2857	94.4857
		$\omega = \omega_2$	87.1429	94.8857
Fisheriris	Default		93.3333	95.3333
	The proposed Method	$\omega = \omega_1$	93.333	96
		$\omega = \omega_2$	100	96
Pineapple Slices	Default		50	53.6
	The proposed Method	$\omega = \omega_1$	88	88.6857
		$\omega = \omega_2$	88	89.08

As demonstrated in Table 3, training accuracy is typically higher than test accuracy for two approaches. This demonstrates that both approaches are effective at empirically minimizing risk in SVM.

We must evaluate the test accuracy generalizability of the two approaches. The first method, which utilizes default values for C and σ, is the least efficient of the three. In other words, the first method is often associated with low test accuracy. The second strategy has the potential to significantly improve test accuracy for the vast majority of datasets. The test accuracy of the first approach varies substantially across datasets, indicating a lack of generalization ability. Only when the optimal value of σ is close to the default value of σ, as is the case with the Fisheriris dataset, is the first method comparable to the others.

The ionosphere dataset demonstrates that the first approach's test accuracy suffers significantly and even tends toward over fitting otherwise. As a result, it is strongly advised that the Gaussian SVM not be executed with the default value of σ.

The proposed method has a high degree of generalization, allowing it to achieve a high level of test precision and best calibration parameters. On the pineapple slices dataset, the approach performs slightly worse than the default method. This is primarily due to the fact that when training with a small sample size, class separability is underestimated.

VI. CONCLUSIONS

The algorithm developed was used to calibrate the SVM classifier and the result from the three dataset showed higher accuracy in proposed calibration method than using default method. The accuracy achieved in the three dataset used were above 85% which is suitable for an online classification implementation as it exceed human accuracy based on 75%.

The analytical method developed in the paper is shown to be a fast, consistent and robust in parameter selection for calibrating the Gaussian radial basis kernel. The method resulted to test and training accuracies above the default. This will ensure the slices classification achieves optimal accuracy every time and capture the daily fruits variability in terms of color, ripeness and texture.

ACKNOWLEDGMENT

I'd like to express my gratitude to both of my supervisors for their constructive comments and helpful suggestions.

REFERENCES

[1] W. Wang, Z. Xu, W. Lu, and X. Zhang, “Determination of the spread parameter in the Gaussian kernel for classification and regression,” Neurocomputing, Vol. 55, 2003, pp. 643-663.

[2] Z. Xu, M. Dai, and D. Meng, “Fast and efficient strategies for model selection of Gaussian support vector machine,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 39, 2009, pp. 1292-1307.

[3] J. Qu and M. J. Zuo, “Support vector machine based data processing algorithm for wear degree classification of slurry pump systems,” Measurement, Vol. 43, 2010, pp. 781-791.

[4] A.Widodo and B.-S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis,” Mechanical Systems and Signal Processing, Vol. 21, 2007, pp. 2560-2574.

[5] LAND VICTORIA, 2002. EDM Calibration Handbook. Edition 7. Land Victoria, Department of NaturalResources and Environment, State of Victoria, Australia.

[6] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, “Parameter determination of support vector machine and feature selection using simulated annealing approach,” Applied Soft Computing, Vol. 8, 2008, pp. 1505-1512.

[7] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, “Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes,” Expert Systems with Applications, Vol. 38, 2011, pp. 5197-5204.

[8] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” Machine Leaning, Vol. 46, 2002, pp. 131-159.

[9] J. Peng and S. Wang, “Parameter selection of support vector machine based on chaotic particle swarm optimization algorithm,” in Proceedings of World Congress on Intelligent Control and Automation, 2010, pp. 1654-1657.

[10] C.-H. Li, C.-T. Lin, B.-C. Kuo, and H. S. Chu, “An automatic method for selecting the parameter of the RBF kernel function to support vector machines,” in Proceedingsof IEEE International Conference on Geoscience and Remote Sensing Symposium, 2010, pp. 836-839.

[11] Z. Liu, M. J. Zuo, and H. Xu, “A Gaussian radial basis function based feature selection algorithm,” in Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2011, pp. 1-4.

[12] S. Maddipati, R. Nandigam, S. Kim and V. Venkatasubramanian, “Learning patterns in combinatorial-protein libraries by Support Vector Machines”, Comput. Chem. Eng., 35, 1143–1151, 2011.

[13] John N. Kamau, P.K.Hinga and S.I.Kamau, “Optimal Accuracy Selection for Gaussian Multiclass SVM through optimization of kernel scale and box constraint”, Proceedings of the Sustainable Research and Innovation Conference, held at JKUAT Main Campus, Kenya, on 6 - 7 October, 2021.

AUTHORS BIOGRAPHY

John Kamau is a Msc. Student in the Electrical and Electronic department. His main research area is in control engineering.

Peterson K. Hinga is an associate professor in the Electrical and Electronic department. His main research area is in power electronic engineering.

Stanley I. Kamau is currently an Associate Professor in the Department of Electrical and Electronic Engineering at Jomo Kenyatta University of Agriculture and Technology (JKUAT) in Kenya. His research interests include hybrid control systems, real-time control and renewable energy development.

Citation of this Article:

J.N. Kamau, P.K. Hinga, S.I. Kamau, "Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 6, Issue 9, pp 1-8, September 2022. Article DOI https://doi.org/10.47001/IRJIET/2022.609001

****

IRJIET

ISSN(online): 2581-3048
Impact Factor : 5.87

CERTIFICATE OF PUBLICATION

INTERNATIONAL RESEARCH JOURNAL OF INNOVATIONS IN ENGINEERING AND TECHNOLOGY

Is Hereby Awarding this Certificate to

J.N. Kamau

Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, P.O. Box 62000 – 00200 Nairobi, Kenya

In Recognition of the Publication of Manuscript Entitled

Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification

Published in International Research Journal of Innovations in Engineering and Technology (IRJIET)

Volume 6, Issue 9, pp 1-8, September 2022

https://doi.org/10.47001/IRJIET/2022.609001

Manuscript ID : IRJIET609001
Date of Issue : September 12, 2022

INTERNATIONAL JOURNAL
IRJIET
INDIA

Editor-In-Chief
IRJIET

Managing Editor
IRJIET

Mail us at: editor@irjiet.com / irjietjournal@gmail.com
Journal Website : www.irjiet.com

IRJIET

ISSN(online): 2581-3048
Impact Factor : 5.87

CERTIFICATE OF PUBLICATION

INTERNATIONAL RESEARCH JOURNAL OF INNOVATIONS IN ENGINEERING AND TECHNOLOGY

Is Hereby Awarding this Certificate to

P.K. Hinga

Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, P.O. Box 62000 – 00200 Nairobi, Kenya

In Recognition of the Publication of Manuscript Entitled

Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification

Published in International Research Journal of Innovations in Engineering and Technology (IRJIET)

Volume 6, Issue 9, pp 1-8, September 2022

https://doi.org/10.47001/IRJIET/2022.609001

Manuscript ID : IRJIET609001
Date of Issue : September 12, 2022

INTERNATIONAL JOURNAL
IRJIET
INDIA

Editor-In-Chief
IRJIET

Managing Editor
IRJIET

Mail us at: editor@irjiet.com / irjietjournal@gmail.com
Journal Website : www.irjiet.com

IRJIET

ISSN(online): 2581-3048
Impact Factor : 5.87

CERTIFICATE OF PUBLICATION

INTERNATIONAL RESEARCH JOURNAL OF INNOVATIONS IN ENGINEERING AND TECHNOLOGY

Is Hereby Awarding this Certificate to

S.I. Kamau

Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, P.O. Box 62000 – 00200 Nairobi, Kenya

In Recognition of the Publication of Manuscript Entitled

Support Vector Machine Kernel Model Calibration for Optimal Accuracy in Automatic Pineapple Slices Classification

Published in International Research Journal of Innovations in Engineering and Technology (IRJIET)

Volume 6, Issue 9, pp 1-8, September 2022

https://doi.org/10.47001/IRJIET/2022.609001

Manuscript ID : IRJIET609001
Date of Issue : September 12, 2022

INTERNATIONAL JOURNAL
IRJIET
INDIA

Editor-In-Chief
IRJIET

Managing Editor
IRJIET

Mail us at: editor@irjiet.com / irjietjournal@gmail.com
Journal Website : www.irjiet.com