

**An Integrated Framework for
Processing of Fruits and Vegetables
using Image Analysis and Machine
Learning Techniques:
A De Novo Approach**

Thesis submitted by
Susovan Jana

Doctor of Philosophy (Engineering)

Department of Production Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India
2021

JADAVPUR UNIVERSITY
KOLKATA – 700032, INDIA

INDEX NO. D-7/E/282/18

1. Title of the thesis: An Integrated Framework for Processing of Fruits and Vegetables using Image Analysis and Machine Learning Techniques: A De Novo Approach

2. Name, Designation & Institution of the Supervisors:

- A. Prof. Bijan Sarkar, Department of Production Engineering, Jadavpur University, Kolkata, India
- B. Prof. Ranjan Parekh, School of Education Technology, Jadavpur University, Kolkata, India

3. List of Publication:

A. Journals (2)

- i. S. Jana, B. Sarkar, and R. Parekh, “An Integrated Framework for the Performance Evaluation of Fruits and Vegetable Store Located in a Supermarket,” Trends in Sciences, vol. 19(3), p. 2073, 2022. <https://doi.org/10.48048/tis.2022.2073>
- ii. S. Jana, R. Parekh, and B. Sarkar, “A De novo approach for automatic volume and mass estimation of fruits and vegetables,” Optik, vol. 200, p. 163443, Jan. 2020. <https://doi.org/10.1016/j.ijleo.2019.163443>

B. Book Chapters (2)

- i. S. Jana, R. Parekh, and B. Sarkar, “Detection of Rotten Fruits and Vegetables using Deep Learning,” Computer Vision and Machine Learning in Agriculture, pp. 31–49, 2021. https://doi.org/10.1007/978-981-33-6424-0_3
- ii. S. Jana, R. Parekh, and B. Sarkar, “Automatic Classification of Fruits and Vegetables: A Texture-Based Approach,” Algorithms in Machine Learning Paradigms, pp. 71–89, 2020. https://doi.org/10.1007/978-981-15-1041-0_5

C. International Conferences (6)

- i. S. Jana, R. Parekh, and B. Sarkar, “Automated Sorting of Rotten or Defective Fruits and Vegetables using Convolutional Neural Network”, International Conference on Computational Intelligence, Data Science and Cloud Computing, pp 43-55, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-4968-1_4
- ii. S. Jana, B. Sarkar, and R. Parekh, “A Holistic Framework for Quality Evaluation of Fruits and Vegetables Suppliers”, International Conference on Computational Intelligence, Data Science and Cloud Computing, pp 155-168, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-4968-1_13
- iii. S. Jana, R. Parekh, and B. Sarkar, “Volume estimation of non-axisymmetric fruits and vegetables using image analysis,” International Conference on Computing, Power and Communication Technologies, pp. 628-633, IEEE, Sept. 2019. <https://ieeexplore.ieee.org/document/8940598>
- iv. S. Jana, R. Parekh, and B. Sarkar, “An Approach Towards Classification of Fruits and Vegetables Using Fractal Analysis,” International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 167–180, Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-8581-0_14
- v. S. Jana and R. Parekh, “Shape-based Fruit Recognition and Classification,” International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 184–196, Springer, Singapore, 2017. https://doi.org/10.1007/978-981-10-6430-2_15
- vi. S. Jana, S. Basak, and R. Parekh, “Automatic fruit recognition from natural images using color and texture features,” Devices for Integrated Circuit (DevIC), pp. 620-624, IEEE, Mar. 2017. <https://doi.org/10.1109/devic.2017.8074025>

4. List of Patents: None

5. List of Presentations in National/International/Conferences/Workshops (5):

- A. Springer- International Conference on Computational Intelligence, Data Science and Cloud Computing (IEM-ICDC), Department of Information Technology, Institute of Engineering and Management, Kolkata, India, 25- 27 September, 2020.
 - Automated Sorting of Rotten or Defective Fruits and Vegetables using Convolutional Neural Network
- B. Springer- International Conference on Computational Intelligence, Data Science and Cloud Computing (IEM-ICDC), Department of Information Technology, Institute of Engineering and Management, Kolkata, India, 25- 27 September, 2020.
 - A Holistic Framework for Quality Evaluation of Fruits and Vegetables Suppliers
- C. IEEE- International Conference on Computing, Power and Communication Technologies (GUCON-2019), Radisson Blu Hotel & Galgotias University, Greater Noida, NCR New Delhi, India, 27-28 September, 2019.
 - Volume estimation of non-axisymmetric fruits and vegetables using image analysis
- D. Springer- International Conference on Computational Intelligence, Communications, and Business Analytics (CICBA-2018), Kalyani Government Engineering College, Kalyani, West Bengal, India, 28 – 29 July, 2018.
 - An Approach Towards Classification of Fruits and Vegetables Using Fractal Analysis
- E. Springer- International Conference on Computational Intelligence, Communications, and Business Analytics (CICBA-2017), Calcutta Business School, 24 Pargana (S), West Bengal, India, 24 – 25 March, 2017.
 - Shape-based Fruit Recognition and Classification

PROFORMA – 1

“Statement of Originality”

I Susovan Jana registered on 23 May 2018 do hereby declare that this thesis entitled “An Integrated Framework for Processing of Fruits and Vegetables using Image Analysis and Machine Learning Techniques: A De Novo Approach” contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the “Policy on Anti Plagiarism, Jadavpur University, 2019”, and the level of similarity as checked by iThenticate software is 5%.

Signature of Candidate: Susovan Jana

Date: 18 June 2021

Certified by Supervisors:

(Signature with date, seal)

1. Bijan Sankar

18 June 2021

Professor
Engineering Department
Jadavpur University
Kolkata - 700 032

2. R. Parekh. 18/6/2021

Professor
School of Education Technology
Jadavpur University
Kolkata - 700 032

PROFORMA - 2

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled "An Integrated Framework for Processing of Fruits and Vegetables using Image Analysis and Machine Learning Techniques: A De Novo Approach" submitted by Shri Susovan Jana, who got his name registered on 23 May 2018 for the award of Ph. D. (Engineering) degree of Jadavpur University is absolutely based upon his own work under the supervision of Prof. Bijan Sarkar and Prof. Ranjan Parekh and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

1. Bijan Sarkar

Prof. Bijan Sarkar
Professor & Former Head
Department of Production Engineering
Jadavpur University, Kolkata, India

Professor
Production Engineering Department
Jadavpur University
Kolkata - 700 032

2. R. Parekh

Prof. Ranjan Parekh
Professor & Director
School of Education Technology
Jadavpur University
Kolkata - 700 032
Jadavpur University, Kolkata, India

Professor
School of Education Technology
Jadavpur University
Kolkata - 700 032

Acknowledgment

Many individuals have contributed directly and indirectly during my journey of Ph.D. (Engineering) research. Here, I would like to take the opportunity to acknowledge them.

At first, I would like to express my sincere gratitude to my respected supervisors Prof. Bijan Sarkar and Prof. Ranjan Parekh. I am fortunate to have both of them as my supervisor. They extended their help in solving critical problems and challenges in my research. They have always been inspirational to overcome all the hurdles. I am grateful to them for their valuable suggestions, motivations, guidance, and advice throughout the journey of my Ph.D.

I would like to extend my sincere gratitude to Prof. Matangini Chattopadhyay, Director of the School of Education Technology, Jadavpur University. She introduced the world of research to me.

I am greatly indebted to Prof. Biplab Ranjan Sarkar, Head of the Production Engineering Department, Jadavpur University. He extended his help to overcome every phase in my Ph.D.


I am thankful to Mr. Hindol Bhattacharya, Mr. Saikat Basak, Mr. Selim Sekh, and Dr. Jyotismita Chaki. I would like to appreciate their impactful technical as well as non-technical help in my research.

I am also thankful to the support staff of the Department of Production Engineering as well as the School of Education Technology for their help in different contexts.

I would like to thank my friend Ms. Nibedita Bhaumik for accompanying and motivating me in critical times during my Ph.D.

Finally, I would like to acknowledge my parents and my uncle Dr. Biswajit Jana. My parents always motivated me for doing higher studies from my childhood days. My uncle guided me in every aspect of decision-making throughout my long academic journey. It would not be possible without their encouragement and support to take a fearless decision of doing a full-time Ph.D.

Thanks & Regards,


(Susovan Jana)

This thesis is dedicated to
My Parents
and
My Uncle (Dr. Biswajit Jana)

Contents

Abstract.....	i
List of Tables.....	iii
List of Figures.....	iv
List of Abbreviation.....	vii
Chapter 1: Introduction	
1.1 Background.....	001
1.2 Aim.....	004
1.3 Objectives.....	004
1.4 Challenges and Motivations.....	006
1.5 Contributions.....	009
1.6 Summary.....	010
Chapter 2: Literature Survey	
2.1 Introduction.....	012
2.2 Segmentation and Detection from Natural Environment.....	013
2.3 Sorting of Fresh and Non-fresh.....	016
2.4 Classification of Fruits and Vegetables.....	019
2.5 Volume and Mass Estimation.....	024
2.6 Supplier Selection.....	027
2.7 Summary.....	028
Chapter 3: Segmentation and Detection from Natural Environment	
3.1 Introduction.....	030
3.2 Proposed Method.....	031
3.2.1 Fruits and Vegetables Segmentation.....	031
3.2.2 Region of Interest (ROI) Detection.....	036
3.3 Dataset.....	037

3.4	Experimentations & Results.....	038
3.5	Summary.....	041
Chapter 4: Sorting of Fresh and Non-fresh		
4.1	Introduction.....	043
4.2	Sorting Parameters.....	044
4.3	Proposed Method.....	045
4.3.1	Convolutional Neural Network.....	045
4.3.2	Proposed Convolutional Neural Network Architecture.....	047
4.3.3	AlexNet Architecture.....	049
4.4	Dataset.....	051
4.5	Experimentations & Results.....	054
4.6	Analysis.....	056
4.7	Summary.....	061
Chapter 5: Classification		
5.1	Introduction.....	062
5.2	Proposed Framework.....	063
5.2.1	Components of Training Phase.....	064
5.2.1.1	Imaging Module (IM).....	064
5.2.1.2	Image Repository (IR).....	065
5.2.1.3	Feature Extraction Module (FEM).....	065
5.2.1.4	Feature Database (FD).....	066
5.2.1.5	Classification Module (CM).....	066
5.2.2	Shape Feature Extraction Module (SFEM).....	067
5.2.2.1	Pre-processing.....	067
5.2.2.2	Measurement of Basic Shape Parameters.....	069
5.2.2.3	Shape Feature Computation.....	070
5.2.3	Color Feature Extraction Module (CFEM).....	071

5.2.4	Texture Feature Extraction Module (TFEM).....	073
5.2.4.1	Fractal Dimension Computation.....	073
5.2.4.2	GLCM features.....	078
5.2.5	Classification Module (CM).....	079
5.2.6	Testing Phase.....	080
5.2.6.1	Imaging Module (IM).....	081
5.2.6.2	Feature Extraction Module (FEM).....	081
5.2.6.3	Classification Module (CM).....	081
5.2.6.4	Final Prediction Module (FPM).....	081
5.3	Dataset.....	082
5.4	Experimentations and Results.....	085
5.5	10-fold Cross-validation.....	095
5.6	Analysis.....	097
5.7	Summary.....	099

Chapter 6: Volume and Mass Estimation

6.1	Introduction.....	101
6.2	Proposed Method.....	102
6.2.1	Pre-processing.....	103
6.2.2	Splitting the image.....	104
6.2.3	Boundary Extraction.....	105
6.2.4	Approximating Polynomial for Outer Boundary.....	107
6.2.5	Volume Estimation.....	108
6.2.6	Mass Estimation.....	109
6.3	Dataset.....	109
6.4	Experimentations & Results.....	111
6.5	Analysis.....	115
6.6	Summary.....	118

Chapter 7: Supplier Selection

7.1	Introduction.....	120
7.2	Preliminaries.....	121
7.2.1	Principal Component Analysis (PCA).....	121
7.2.2	Fuzzy Set Theory.....	123
7.2.3	Technique for Order of Preference by Similarity to Ideal Solution...124	
7.3	Proposed Framework.....	127
7.4	Identification of Criteria.....	128
7.5	Numerical Example.....	130
7.6	Summary.....	134

Chapter 8: Conclusions & Future Scopes

8.1	Conclusions.....	136
8.2	Future Scopes.....	138

References.....	139
-----------------	-----

Abstract

Fruits and vegetables are necessary for our daily life as it contains many important nutrition constrains. Hence, fruits and vegetables are the most profitable agricultural product. The appropriate selection of suppliers is very important for a retail store of fruits and vegetables. The fruits and vegetable processing includes lots of tasks from harvesting to reach the customer's hand. The tasks are segmentation, sorting, classification, grading, etc. The manual execution of those tasks is very time-consuming and requires a huge number of expert resources. Hence, this research aims to propose a framework for automating those tasks using image analysis and machine learning techniques. This work proposes an end-to-end framework for fruits and vegetable processing in agricultural industries as well as in the supermarket.

The fruits and vegetable segmentation is mandatory for further processing. Segmentation and detection are also important for harvesting fruits and vegetables from the natural environment. A graph-based segmentation technique was used to segment the foreground fruit or vegetable object from the natural background. A region of interest detection technique is also proposed here. The experimentation shows that the proposed technique performs better than the popular Otsu thresholding technique in this context.

The non-fresh i.e. rotten or defective fruits and vegetables are very harmful to fruits and vegetables inventory or store. The non-fresh fruits and vegetables need to be detected and removed as early as possible. The computer vision system has to rely only on the visual features to label fruit and vegetable as rotten or defective. Another challenge is that the pattern of rot and defect is different for different types of fruits or vegetables. A convolutional neural network architecture has been proposed for the classification of fresh and non-fresh fruit and vegetables. The test accuracy and f1 score of the proposed network architecture are 97.74% and 98.43% respectively.

There is an almost infinite number of fruits and vegetable species in the world. Hence, the classification of fruits and vegetables using visual features is a very difficult task. A novel framework has been proposed here for the classification of fruits and vegetables using image analysis and machine learning techniques. The shape, color, and texture features are combined to classify fruits and vegetables. An exploratory analysis has been done on the performance of different classification algorithms in this context. The experimentations are done on 35 classes

of edible fruits and vegetables. The proposed framework achieves at most 99.96% classification accuracy.

The grading is very necessary to get the proper market value of fruits and vegetables. There are different grading parameters i.e. shape, size, color, volume, mass, etc. Volume and mass estimation is more challenging using computer vision than the estimation of other parameters. The challenge increases when the shape of the fruit or vegetable is irregular or non-axisymmetric. Here, a novel split and merge technique has been proposed to estimate the mass and volume of fruit and vegetable from a single image. The experimentation has been done on both regular and axisymmetric as well as irregular and non-axisymmetric fruit and vegetable. The results have been validated with water displacement and digital balance for volume and mass respectively. The overall correlation for volume and mass estimations are 0.96 and 0.97 respectively.

The selection of an appropriate supplier is very important to reduce the complexity in further processing of fruits and vegetables. A holistic framework for fruit and vegetable supplier selection is proposed in this thesis. The proposed framework integrates principal component analysis and technique for order of preference by similarity to ideal solution. The proposed approach reduces the number of criteria as well as correlations among the criteria. It also generates weights through the principal component analysis for each of the newly generated criteria. The framework finally ranks among the suppliers.

List of Tables

Tables	Page No.
Table 4.1. Distribution of training and testing images of different datasets	53
Table 4.2. The prediction performance using the proposed CNN model on different datasets	55
Table 4.3. The prediction performance using the fine-tuned AlexNet model on different datasets	55
Table 4.4. The prediction performance using Grayscale Histogram with SVM on different datasets	56
Table 4.5. The prediction performance using GLCM Features with SVM on different datasets	56
Table 4.6. The prediction performance using Bag of Features with SVM on different datasets	56
Table 4.7. The prediction performance on dataset 5 and dataset 6 using trained CNN & AlexNet	60
Table 5.1. Class labels and number of images for the corresponding class	83
Table 5.2. Class-wise accuracy from the top view using different classifiers	93
Table 5.3. Class-wise accuracy from the side view using different classifiers	94
Table 5.4. Class-wise accuracy using different classifiers in the complete proposed framework	94
Table 5.5. Distribution of images over multiple folds	95
Table 5.6. Class-wise final accuracy on different folds using the proposed approach	96
Table 5.7. The details of the previous approaches implemented for comparison with the proposed approach	98
Table 5.8. Class-wise prediction accuracy using separate camera view and previous approaches	98
Table 6.1. Volume and mass estimation results of potato samples using standard technique, proposed approach, and Venkatesh et al. approach	113
Table 6.2. Volume and mass estimation results of citrus samples using standard technique, proposed approach, and Venkatesh et al. approach	114
Table 6.3. Volume and mass estimation results of tomato samples using standard technique, proposed approach, and Venkatesh et al. approach	114
Table 6.4. Comparison of absolute percentage of error for volume and mass estimation where the proposed approach overcomes the limitations of the Venkatesh et al. approach	118
Table 7.1. Linguistic rating, fuzzy number, and defuzzified crisp value	124
Table 7.2. Details of the experts in the committee (Explicit Knowledge)	127
Table 7.3. Hierarchical presentation of criteria	129
Table 7.4. The defuzzified and normalized decision matrix	131
Table 7.5. Decision matrix with uncorrelated principal components	132
Table 7.6. Normalized weights of principal components	132
Table 7.7. Weighted normalized decision matrix with principal components	133
Table 7.8. DPIS, DNIS, CC, and Rank	133

List of Figures

Figures	Page No.
Fig. 1.1. Euler diagram for the overlaps of fruits and vegetables	2
Fig. 1.2. Types of fruits and vegetables produced in different states of India	2
Fig. 1.3. State-wise fruits and vegetables cultivation area in India	3
Fig. 1.4. State-wise fruits and vegetables production in India	3
Fig. 1.5. The complete end to end framework of fruits and vegetable processing	5
Fig. 1.6. Challenges of segmentation and detection of fruits and vegetables from natural background (a) Complex background (b) Occlusion by leaf (c) Similar object and background color (d) Multiple appearances of fruits	7
Fig. 1.7. Challenges of detecting the non-fresh fruits and vegetables	7
Fig. 1.8. (a) Various classes of fruits and vegetables (b) Different growth cycles of tomato fruit	8
Fig. 1.9. (a)(b) The inter-class similarity between capsicum and orange (c)(d) Intra-class dissimilarities of apple	8
Fig. 1.10. The cauliflower from different viewing angles	8
Fig. 1.11. Geometrical Transformations (a) Actual Image (b) Translation (c) Rotation (d) Scaling	8
Fig. 1.12. Challenges of volume and mass estimation of fruits and vegetables (a) Single view (b) Irregular shape (c) Non-axisymmetric	9
Fig. 2.1. Simplified process flow diagram of Jidong et al. (2016) approach	13
Fig. 2.2. Simplified process flow diagram of Xiang et al. (2013a) approach	14
Fig. 2.3. Simplified process flow diagram of Xiang et al. (2013b) approach	14
Fig. 2.4. Simplified process flow diagram of Chandini et al. (2018) approach	17
Fig. 2.5. Simplified process flow diagram of Karakaya et al. (2019) approach	17
Fig. 2.6. Simplified process flow diagram of Seng et al. (2009) approach	19
Fig. 2.7. Simplified process flow diagram of Roomi et al. (2012) approach	20
Fig. 2.8. Simplified process flow diagram of Cornejo et al. (2016) approach	20
Fig. 2.9. Simplified process flow diagram of Venkatesh et al. (2015) approach	24
Fig. 2.10. Simplified process flow diagram of Uluişik et al. (2018) approach	25
Fig. 3.1. Fruits under natural environment	31
Fig. 3.2. Process flow diagram of the proposed technique	31
Fig. 3.3. Process flow of segmentation (GrabCut)	32
Fig. 3.4. Graph of pixels	35
Fig. 3.5. Simplified steps of GrabCut	36
Fig. 3.6. Sample images from the dataset	38
Fig. 3.7. The flow of segmentation and ROI detection	39
Fig. 3.8. The final output after segmentation and ROI detection using the proposed approach	40
Fig. 3.9. Comparison of segmentation performance between the proposed approach and Otsu thresholding	41
Fig. 4.1. A demonstration of the convolution process	46
Fig. 4.2. An example of max pooling	47
Fig. 4.3. The architecture of the proposed Convolutional Neural Network model	48
Fig. 4.4. The architecture of the fine-tuned AlexNet	50

Fig. 4.5. Samples from all the categories (a) Fresh Apple (b) Rotten Apple (c) Fresh Banana (d) Rotten Banana (e) Fresh Orange (f) Rotten Orange	51
Fig. 4.6. (a) A sample image and it's appearance after applying different augmentation techniques (b) 15° rotation, (c) 30° rotation, (d) 45° rotation, (e) 60° rotation, (f) 75° rotation, (g) Salt and pepper noise, (h) Translation, and (i) Vertical flip	52
Fig. 4.7. Samples from dataset 5 (a), (b) Fresh apple (c), (d) Non-fresh apple	53
Fig. 4.8. Samples from dataset 6 (a)(b) Fresh fruits and vegetables (c)(d) Non-fresh fruits and vegetables	53
Fig. 4.9. Training vs validation accuracy of the proposed CNN model on dataset 4	57
Fig. 4.10. Training vs validation loss of the proposed CNN model on dataset 4	57
Fig. 4.11. The extracted features by final convolution layer in (row 1) Proposed CNN model (row 2) Fine-tuned AlexNet	58
Fig. 4.12. Comparison of overall accuracy using different approaches on different datasets	58
Fig. 4.13. Comparison of recall using different approaches on different datasets	59
Fig. 4.14. Comparison of precision using different approaches on different datasets	59
Fig. 4.15. Comparison of f1 score using different approaches on different datasets	59
Fig. 4.16. Performance of CNN and fine-tuned AlexNet trained by Dataset 1	61
Fig. 4.17. Performance of CNN and fine-tuned AlexNet trained by Dataset 4	61
Fig. 5.1. A top-level view of the proposed framework for fruits and vegetable classification	64
Fig. 5.2. The detailed architecture of the training phase	64
Fig. 5.3. Structure of the imaging setup	65
Fig. 5.4. The process flow in shape feature extraction module	66
Fig. 5.5. (a) Input color (RGB) image (b) Grayscale image (c) Segmented binary image (d) Complemented binary image (e) After removing background noises (f) Rotated Image (g) After filling inner holes	68
Fig. 5.6. Shape parameters for a sample	69
Fig. 5.7. Process flow in color feature extraction module	71
Fig. 5.8. Process flow of texture feature extraction module	72
Fig. 5.9. First four iterations of Koch Snowflake	73
Fig 5.10. (a) Input RGB color image (b) Grayscale image	74
Fig. 5.11. Binary images generated from Fig. 5.10(b) using Eq. (5.28) with $N_t = 4$	75
Fig. 5.12. Binary images generated from Fig. 5.10(b) using Eq. (5.29) with $N_t = 4$	76
Fig. 5.13. Border images generated from the images in Fig. 5.11	77
Fig. 5.14. Border images generated from the images in Fig. 5.12	77
Fig. 5.15. The detailed architecture of the testing phase	80
Fig. 5.16. The detailed flow of the final prediction module	82
Fig. 5.17. Sample images for each class from the top view	84
Fig. 5.18. Sample images for each class from the side view	85
Fig. 5.19. Plot for sf_1	86
Fig. 5.20. Plot for sf_2	87
Fig. 5.21. Plot for sf_3	87
Fig. 5.22. Plot for sf_4	87
Fig. 5.23. Plot for sf_5	87
Fig. 5.24. Plot for sf_6	88
Fig. 5.25. Plot for sf_7	88

Fig. 5.26. Plot for sf_8	88
Fig. 5.27. Plot for σ_R	88
Fig. 5.28. Plot for σ_G	89
Fig. 5.29. Plot for σ_B	89
Fig. 5.30. Plot for D_1	89
Fig. 5.31. Plot for D_2	89
Fig. 5.32. Plot for D_3	90
Fig. 5.33. Plot for D_4	90
Fig. 5.34. Plot for D_5	90
Fig. 5.35. Plot for D_6	90
Fig. 5.36. Plot for D_7	91
Fig. 5.37. Plot for Ct_{0°	91
Fig. 5.38. Plot for Cn_{0°	91
Fig. 5.39. Plot for Ey_{0°	91
Fig. 5.40. Plot for Hy_{0°	92
Fig. 5.41. Overall accuracy using different classifiers on different flows of the proposed framework	92
Fig. 5.42. Class-wise prediction accuracy comparison among previous approaches on top view images and the proposed approach	99
Fig. 5.43. Class-wise prediction accuracy comparison among previous approaches on the side view images and the proposed approach	99
Fig. 6.1. Process flow of the proposed technique	102
Fig. 6.2. (a) Input image (b) Grayscale version (c) Binary version (d) Complemented binary version (e) After the exclusion of noise regions (f) After filling holes	104
Fig. 6.3. Alignment of the major axis along the horizontal plane	105
Fig. 6.4. After splitting the I_r into two parts (a) upper part (b) lower part	105
Fig. 6.5. Valid boundary points (blue) and approximated polynomial (red) for the (a) upper part and (b) lower part	107
Fig. 6.6. A cross-sectional view along major-axis	109
Fig. 6.7. The imaging setup	110
Fig. 6.8. Samples from the dataset	110
Fig. 6.9. (a) Digital Calipers (b) Digital Balance	111
Fig. 6.10. The scatter plot between the proposed approach and standard technique for (a) Volume and (b) Mass	115
Fig. 6.11. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the potato samples	117
Fig. 6.12. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the citrus samples	117
Fig. 6.13. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the tomato samples	118
Fig. 7.1. Fruits and vegetable supply chain in India	121
Fig. 7.2. Fuzzy membership functions of linguistic terms for expressing the rating	124
Fig. 7.3. The proposed framework for supplier selection of fruits and vegetable	126
Fig. 7.4. Scree plot of the eigenvalues	132
Fig. 7.5. Variation of closeness coefficients with a different combination of weights	134

List of Abbreviations

Abbreviations	Full-Form
ROI	Region of Interest
RHT	Randomized Hough Transform
MLP	Multilayer Perceptron
RBPNN	Radial Basis Probabilistic Neural Network
HSV	Hue Saturation Value
GLCM	Gray-level co-occurrence matrix
SIFT	Scale Invariant Feature Transform
ANN	Artificial Neural Network
VGG-16	Visual Geometry Group-16
HOG	Histogram Oriented Gradient
LBP	Local Binary Pattern
PCA	Principal Component Analysis
SVM	Support Vector Machine
kNN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
PROMETHEE	Preference Ranking Organization Method for Enrichment Evaluation
SWOT	Strengths, Weaknesses, Opportunities, and Threats
GRA	Grey Relational Analysis
AHP	Analytic Hierarchy Process
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
GMM	Gaussian Mixture Model
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
FR	Foreground Region
RA	Region Area
IM	Imaging Module
IR	Image Repository
FEM	Feature Extraction Module
FD	Feature Database
CM	Classification Module
TVC	Top View Camera
SVC	Side View Camera
TVIR	Top View Image Repository
SVIR	Side View Image Repository
TI	Top View Image
SI	Side View Image
SFEM	Shape Feature Extraction Module
CFEM	Color Feature Extraction Module
TFEM	Texture Feature Extraction Module
SFV	Shape Feature Vector

CFV	Color Feature Vector
TFV	Texture Feature Vector
SFD	Shape Feature Database
CFD	Color Feature Database
TFD	Texture Feature Database
CLM	Classification Model
CL	Class Label
BSP	Basic Shape Parameters
GF	GLCM Features
FD	Fractal Dimension
TPL	Top View Prediction Level
TPS	Top View Prediction Score
SPL	Side View Prediction Level
SPS	Side View Prediction Score
FPL	Final Predicted Level
CF	Conversion Factor
COG	Center of Gravity
PIS	Positive Ideal Solution
NIS	Negative Ideal Solution
CC	Closeness co-efficient

Introduction

1.1 Background

Fruits are the most important part of a plant. The fruit is formed from the ovary after flowering. The fruits are usually edible, fleshy, tasty. Most of the fruit contains one or more seeds inside. Nowadays there is a demand for seedless fruit, which can be produced through the parthenocarpy process. Vegetables can be a complete plant or part of a plant including flowers, fruits, stems, leaves, roots, and seeds, which are used as food, such as cauliflower, potato, or bean, etc. Vegetables can be eaten either raw or cooked. Some agricultural products are biologically fruit but people consume them as vegetables i.e. tomato, brinjal, capsicum, etc. Some agricultural products are not biologically fruit but people consume them as fruit i.e. pineapple, strawberry, etc. Fig. 1.1 shows the overlapping between fruits and vegetables. There are many species of fruits and vegetables. All are not consumable. Consumable fruits and vegetables play a very important role in human nutrition. It gives us many important vitamins and minerals directly.

India ranked second for fruits and vegetable production in the world as per the National Horticulture Database in 2015-16 [NHB, 2016]. India ranks first for the production of fruits like Mango, Papaya, and Banana, etc., and ranks second for the production of necessary

vegetables like potato, brinjal, cabbage, and onion, etc. Fig. 1.2 shows the different types of fruits and vegetables produced in different states of India.

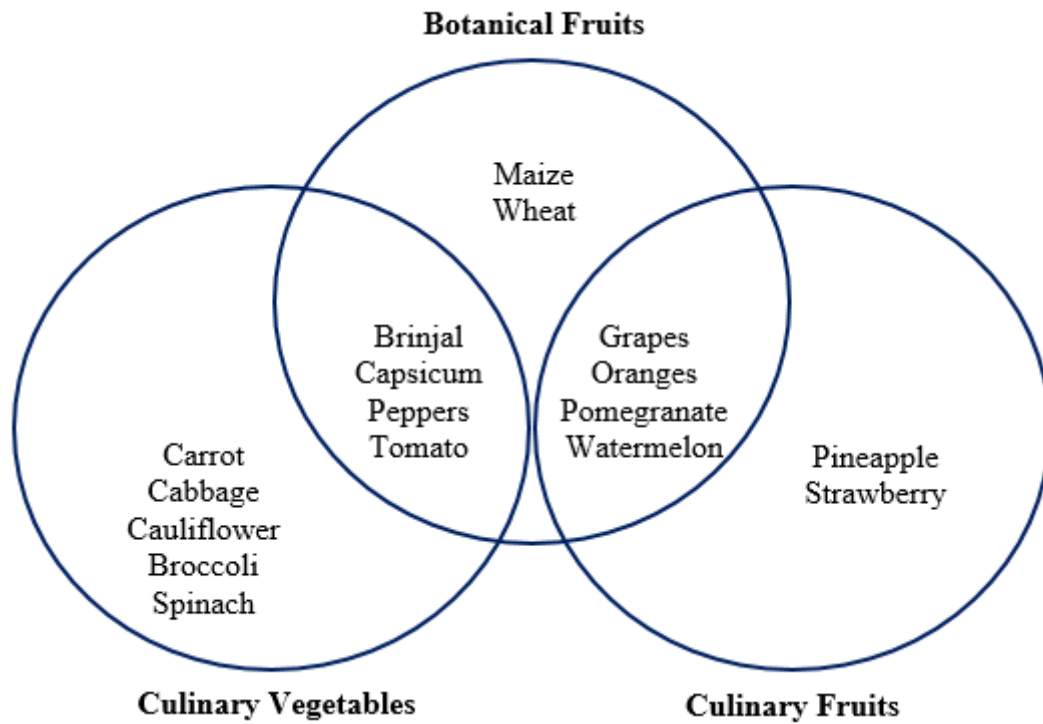


Fig. 1.1. Euler diagram for the overlaps of fruits and vegetables

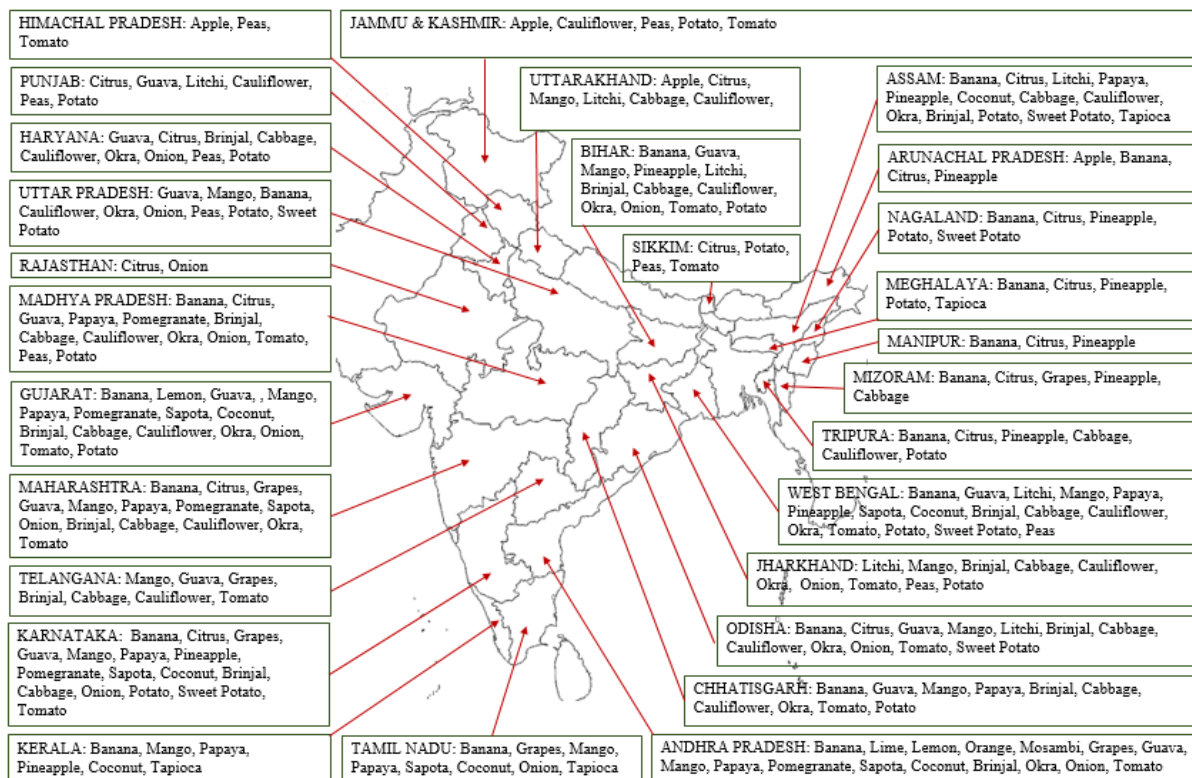


Fig. 1.2. Types of fruits and vegetables produced in different states of India

Fig. 1.3 depicts the state-wise cultivation area of fruits and vegetables in India. Fig. 1.4 depicts the state-wise production of fruits and vegetables in India. The data for Fig. 1.3 and 1.4 have been taken from the final report of “Area and production of horticulture crops for 2017 - 18 (Final)” from the National Horticulture Board. Fruits worth 668.75 USD Millions and vegetables worth 608.48 USD Millions have been exported from India in 2019-20 [APEDA, 2019]. The major importer of Indian fresh fruit and vegetables are UK, Netherland, UAE, Nepal, Sri Lanka, Bangladesh, Oman, Qatar, etc.

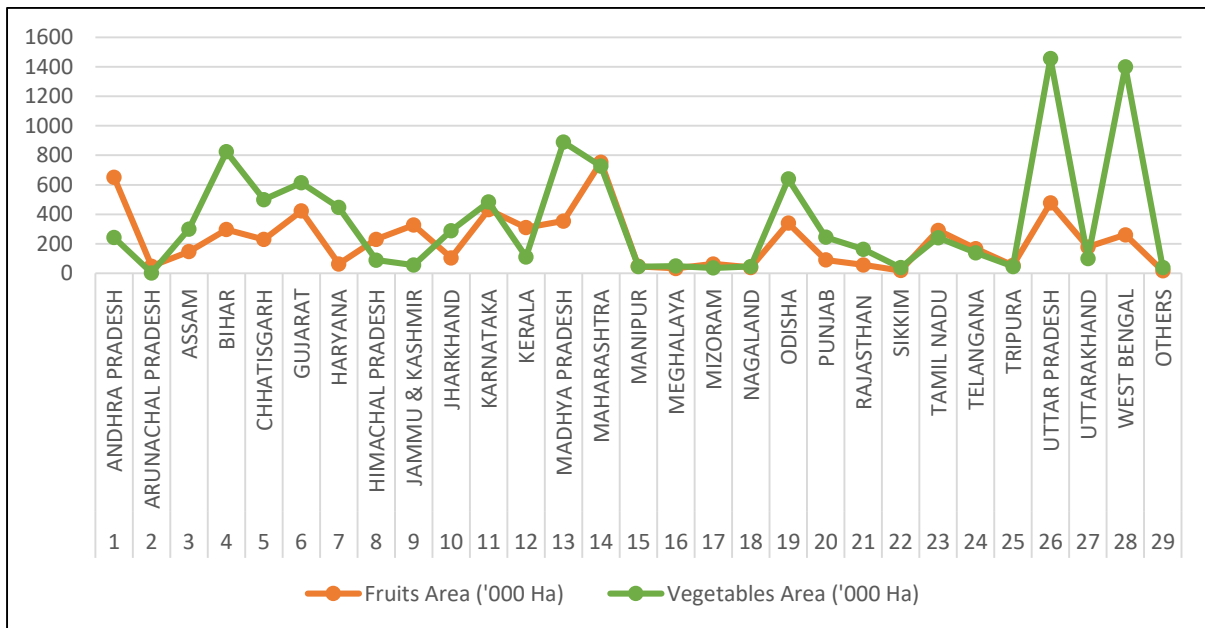


Fig. 1.3. State-wise fruits and vegetables cultivation area in India

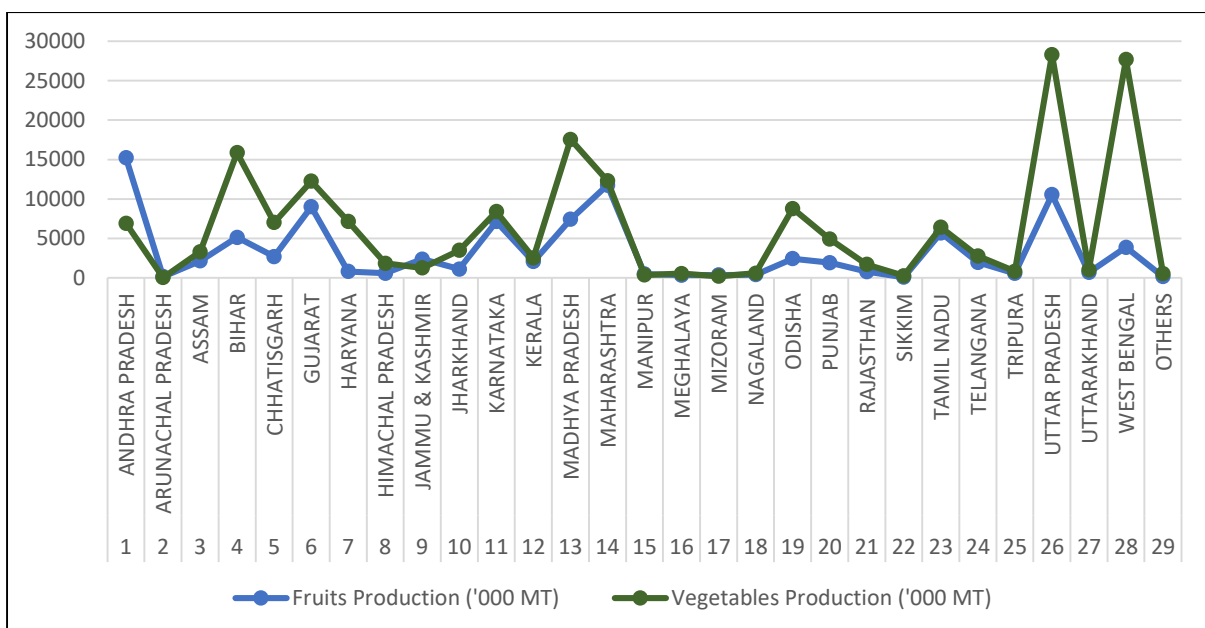


Fig. 1.4. State-wise fruits and vegetables production in India

People generally buy fresh fruits and vegetables from the open market. The scenario is changing now a day. The supermarkets are providing fresh fruits and vegetables like an open market. People are getting accustomed to this trend of buying fresh fruits and vegetables from supermarkets. The growing population creates an increment of demand for fruits and vegetables. There is a huge demand for fruits and vegetables in the food processing industry as well. This demand is the main reason for the huge growth of the agricultural business. Hence, cultivation is the major source of income for the majority in India as well as in other developing countries. The fruits and vegetables need to be harvested from the agricultural field. The fruits and vegetables pass many stages after harvesting in supermarkets or industries i.e. sorting, classification, grading, etc. Proper selection of suppliers reduces effort in all the processing. Supplier selection for fruits and vegetables in supermarkets or food processing industries is also coming under the same radar. There is a need for a large number of manual resources for doing those tasks on time and compete with the faster production chain in the world. The main problems are the lack of skilled resources and shortage of time. Hence, the harvesting, sorting, classification, and grading can be done by automation to solve those problems partially. Smart inventory management [Desai, 2019] can also be done with the help of automation and IoT.

Computer vision and machine learning have already kept his footsteps in the agricultural field as well as in industries, which are based on agricultural products. The automated processing of fruits and vegetables can be done with the help of Image Analysis and Machine Learning techniques. Hence, this research was initiated by us.

1.2 Aim

This work aims to propose an automated end-to-end framework for the processing of fruits and vegetables using image analysis and machine learning. The proposed framework will be highly efficient, and accurate. It will also provide an elegant and user-friendly solution by addressing the existing challenges of automated processing. Fig. 1.5 shows the end-to-end flow of the complete framework.

1.3 Objectives

There are multiple objectives to reach the goal. The proposed framework contains four stages. Those four stages are the four main objectives of this research. The motivation also includes the multi-criteria decision making for an appropriate selection of fruits and vegetable suppliers. The selection of the best supplier is a very crucial decision for any industry.

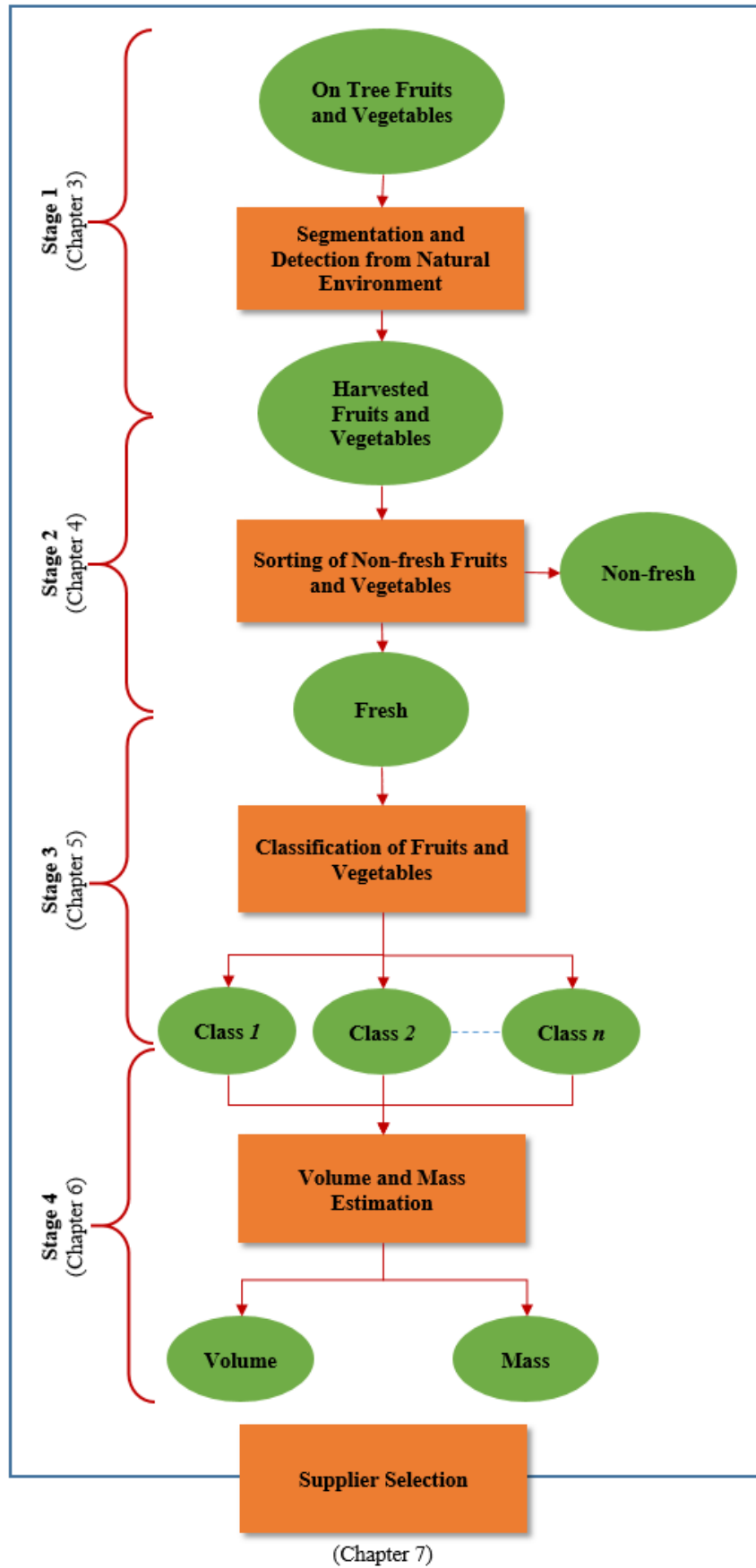


Fig. 1.5. The complete framework of fruits and vegetable processing

- The segmentation and detection of fruits and vegetables from the natural environment for auto harvesting robots.
- Sorting the rotten or defective fruits and vegetables from the lot. That means a classification of fresh and non-fresh fruits and vegetables from the visual appearance.
- Classification among different known types of fruits and vegetables with the help of its visual features. Identification of most discriminating features and suitable classification algorithms.
- The quality grading of fruits and vegetables. Estimating the volume and mass as grading parameters.
- Selection of an appropriate supplier for a fruits and vegetable store.

1.4 Challenges & Motivations

A literature survey was done to find the gap and challenges of existing works. The detailed survey has been presented in Chapter 2. The unaddressed challenges are the motivation of this work. Our findings in this regard are-

1.4.1 Accurate detection of the region of interest is very important for adjusting the size of the harvesting hand in an auto harvesting robot. Segmentation is necessary before marking the region of interest (ROI). The challenges of segmentation of fruits and vegetables from the natural background are-

- (a) The complex background, which may contain leaves, stems, sky, and other objects.
- (b) Dealing with the partial occlusion by stem or leaf.
- (c) The segmentation of fruits and vegetables which has similar color with its background e.g. unripe mango and leaf of mango tree both are green.
- (d) The appearance of more than one fruit or vegetable in the image creates confusion in ROI selection.



(a)



(b)



(c)



(d)

Fig. 1.6. Challenges of segmentation and detection of fruits and vegetables from natural background (a) Complex background (b) Occlusion by leaf (c) Similar object and background color (d) Multiple appearances of fruits

1.4.2 Rotten fruits are harmful to human health. It also damages other fruits and vegetables in the lot. The defective fruits and vegetables should also be removed from the lot to increase the trust of the customer. The challenges are-

- (a) There are a variety of fruits and vegetables and their visual appearance is also different.
- (b) The appearance after rotten, defect, or damage is also different for different types of fruits and vegetables.



Fig. 1.7. Challenges of detecting the non-fresh fruits and vegetables

1.4.3 Classification among different types of fruit and vegetables is required to place different fruits and vegetables automatically in a different bucket. The classification could be done with the help of visual features like texture, color, shape, and size. The challenges in discriminating different types of fruits and vegetables are-

- (a) An almost infinite variety of visual properties like color, shape, texture for identifying a potentially huge number of fruits and vegetables with sufficient accuracy and reliability.
- (b) Dealing with the fact that the visual properties are not constant but exhibit seasonal variations and also due to growth cycles.

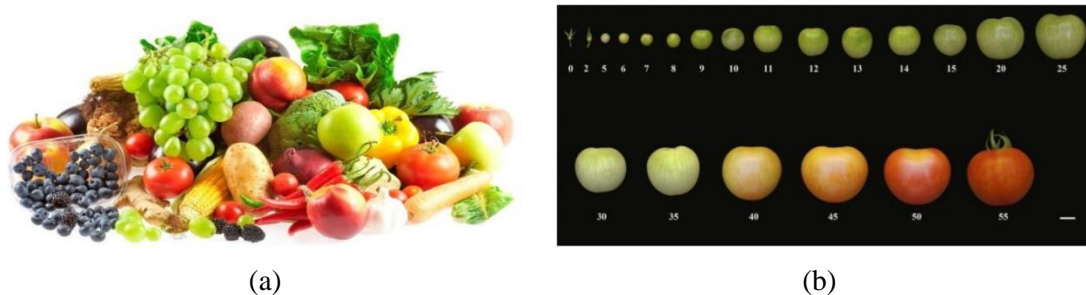


Fig. 1.8. (a) Various classes of fruits and vegetables (b) Different growth cycles of tomato fruit

(c) Dealing with inter-class similarities and intra-class differences.

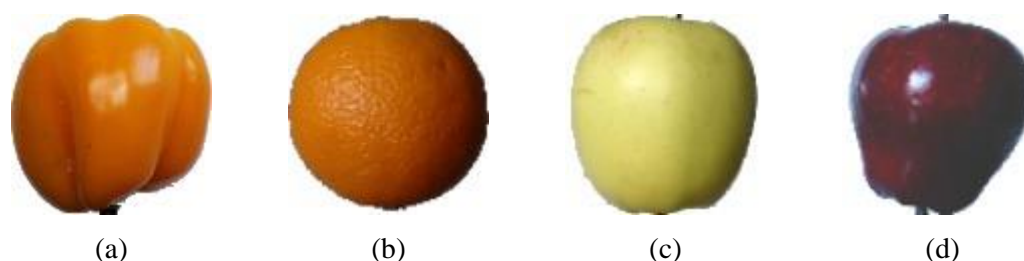


Fig. 1.9. (a)(b) The inter-class similarity between capsicum and orange (c)(d) Intra-class dissimilarities of apple

(d) Change of viewing angles



Fig. 1.10. The cauliflower from different viewing angles

(e) Possibilities of image transformations e.g. translation, rotation, scaling.

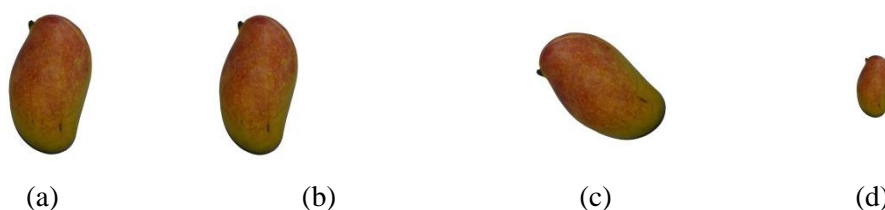


Fig. 1.11. Geometrical Transformations (a) Actual Image (b) Translation (c) Rotation (d) Scaling

1.4.4 There are different grading parameters for fruits and vegetables. Among them, some popular parameters are volume and mass. Volume and mass estimation from images is really difficult. The challenges are –

- (a) Estimation volume and mass from a single view image.
- (b) Dealing with irregular-shaped fruit and vegetable.

(c) Dealing with non-axisymmetric fruits and vegetables.



Fig. 1.12. Challenges of volume and mass estimation of fruits and vegetables (a) Single view (b) Irregular shape (c) Non-axisymmetric

1.4.5 The supplier selection problem has been addressed for different industries. This problem has not been explored well for fruits and vegetable suppliers. The supplier's performance needs to be evaluated for making a perfect choice. The challenges are-

- (a) Identification of different criteria for the supplier evaluation of a perishable item like fruits and vegetables.
- (b) Ranking of fruits and vegetable suppliers based on those multiple criteria.

1.5 Contributions

The problem and challenges are addressed in different published papers. This section provides the mapping between the objectives and published papers.

- Segmentation and ROI detection of fruits and vegetables from the natural environment. (Chapter 3)
 - S. Jana, S. Basak, and R. Parekh, “Automatic fruit recognition from natural images using color and texture features,” *Devices for Integrated Circuit (DevIC)*, pp. 620-624, IEEE, Mar. 2017. <https://doi.org/10.1109/devic.2017.8074025>
- Sorting of rotten or defective fruits and vegetables from the fresh one. (Chapter 4)
 - S. Jana, R. Parekh, and B. Sarkar, “Detection of Rotten Fruits and Vegetables using Deep Learning,” *Computer Vision and Machine Learning in Agriculture*, pp. 31–49, 2021. https://doi.org/10.1007/978-981-33-6424-0_3
 - S. Jana, R. Parekh, and B. Sarkar, “Automated Sorting of Rotten or Defective Fruits and Vegetables using Convolutional Neural Network”, *International Conference on Computational Intelligence, Data Science and Cloud Computing*, pp 43-55, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-4968-1_4
- Classification of different type of fruits and vegetables (Chapter 5)

- S. Jana, R. Parekh, and B. Sarkar, “Automatic Classification of Fruits and Vegetables: A Texture-Based Approach,” *Algorithms in Machine Learning Paradigms*, pp. 71–89, 2020. https://doi.org/10.1007/978-981-15-1041-0_5
- S. Jana, R. Parekh, and B. Sarkar, “An Approach Towards Classification of Fruits and Vegetables Using Fractal Analysis,” *International Conference on Computational Intelligence, Communications, and Business Analytics*, pp. 167–180, Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-8581-0_14
- S. Jana, and R. Parekh, “Shape-based Fruit Recognition and Classification,” *International Conference on Computational Intelligence, Communications, and Business Analytics*, pp. 184–196, Springer, Singapore, 2017. https://doi.org/10.1007/978-981-10-6430-2_15
- Estimation of volume and mass from image (Chapter 6)
 - S. Jana, R. Parekh, and B. Sarkar, “A De novo approach for automatic volume and mass estimation of fruits and vegetables,” *Optik*, vol. 200, p. 163443, Jan. 2020. <https://doi.org/10.1016/j.ijleo.2019.163443>
 - S. Jana, R. Parekh, and B. Sarkar, “Volume estimation of non-axisymmetric fruits and vegetables using image analysis,” *International Conference on Computing, Power and Communication Technologies*, pp. 628-633, IEEE, Sept. 2019. <https://ieeexplore.ieee.org/document/8940598>
- Supplier selection for a fruits and vegetable store (Chapter 7)
 - S. Jana, B. Sarkar, and R. Parekh, “A Holistic Framework for Quality Evaluation of Fruits and Vegetables Suppliers”, *International Conference on Computational Intelligence, Data Science and Cloud Computing*, pp 155-168, Springer, Singapore, 2021. https://doi.org/10.1007/978-981-33-4968-1_13

1.6 Summary

This chapter introduces the perspective of this work. Fruits and vegetables are very important for proper nutrition. There are many stages from harvesting fruits and vegetables to place them in the basket for sale in a supermarket. Those stages include harvesting, sorting, classification, grading. The appropriate selection of suppliers is required on top of all the stages. The manual execution requires skilled persons and a huge time. The agricultural field, as well as industry, needs automated processing of fruits and vegetables. The task of harvesting, sorting, classification, and grading can be automated with the help of image analysis and machine

learning techniques. This chapter also describes the objectives and the corresponding challenges to make a complete end-to-end framework. The challenges are the motivation of our work. This work addresses most of the challenges with a solution in different published papers.

Literature Survey

2.1 Introduction

The planned work involves multiple stages and contains multiple modules. There is no such research work that proposes a complete end-to-end framework incorporating all the modules. This is the main motivation for this work. Hence, the literature survey has been done separately for each of the modules. The state-of-the-art research works are elaborated here in different sections. The automatic harvesting robot must have a visionary system. The first task of making a visionary system is to segment and detect the fruit or vegetable from the natural environment. Section 2.2 illustrates prior works to address this problem. A lot of fruits and vegetables may contain some defective or rotten items. Those items should be detected and discarded as early as possible. Section 2.3 outlines the techniques used by the earlier researches to detect defective or rotten fruits and vegetables. The classification among different types of fruit and vegetable is required to place the fruits and vegetables in a different bucket. Section 2.4 describes the classification techniques used in previous approaches. Grading of fruits and vegetables is very important in the context of the supermarket. The volume and mass are very important grading criteria. Hence, previous works on volume and mass estimation of fruits and vegetables were illustrated in section 2.5. The selection of the best fruits and vegetable supplier is also very important from the management perspective of a fruits and vegetable store. Section 2.6 brings

the previous supplier selection techniques for the fruits and vegetable store. Section 2.7 contains the summary of this complete literature review.

2.2 Segmentation and Detection from Natural Environment

Jidong et al. proposed an apple recognition technique from the natural environment [Jidong et al., 2012; Jidong et al., 2016]. Fig. 2.1 shows the simplified process flow diagram of this technique. At first, the color features were extracted in different color spaces. The color feature was used to segment the apple from the background. Then edge detection and thinning operations were done after filling holes and removing noises from the segmented image. Randomized Hough transform (RHT) was applied to detect the apple regions from the edges. The experimentations were done with non-occluded, occluded (branches and leaves), and overlapped apple fruits. The detection accuracy is 100% for overlapped and non-occluded apples whereas the detection accuracy is 86% for the occluded apples. Hence, the detection performance should be improved for the apples occluded by the branches and leaves.

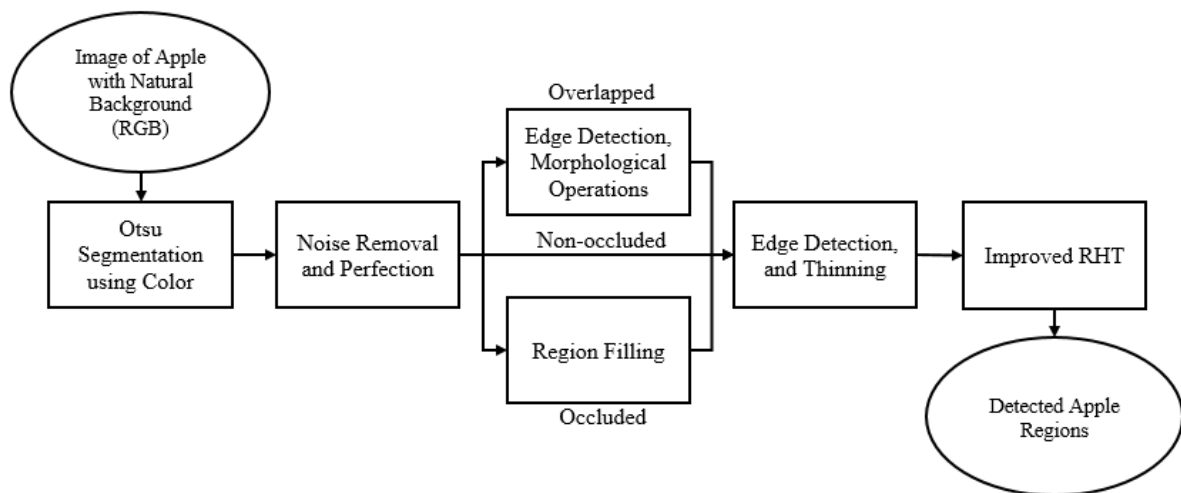


Fig. 2.1. Simplified process flow diagram of Jidong et al. (2016) approach

Xiang et al. proposed a technique for the detection and segmentation of occluded tomatoes from the natural environment [Xiang et al., 2013a]. Fig. 2.2 shows the simplified process flow diagram of this technique. Ripe tomatoes were easily segmented from the background using color differences. The edge points were extracted from the segmented image. The edges are sorted based on some pre-defined criteria. The edge points, which have abnormal curvature, were removed and applied a circle regression algorithm. The circle regression detects multiple circles but all the detected circles are not valid to be a tomato region. The detected circles will also be removed if the radius of the circles is not within the predefined range. If the distance

between the centers of two detected circles is less than the radius of those circles but their radius is within a predefined range, then the edges are assumed to be a part of a single tomato. The edges are merged and circle regression is applied again. The result shows that the detection rate is 90% where the occlusion is below 25%. The performance needs improvement for the moderate (occlusion is between 25% to 50%) or serious (occlusion is greater than 50%) occlusion.

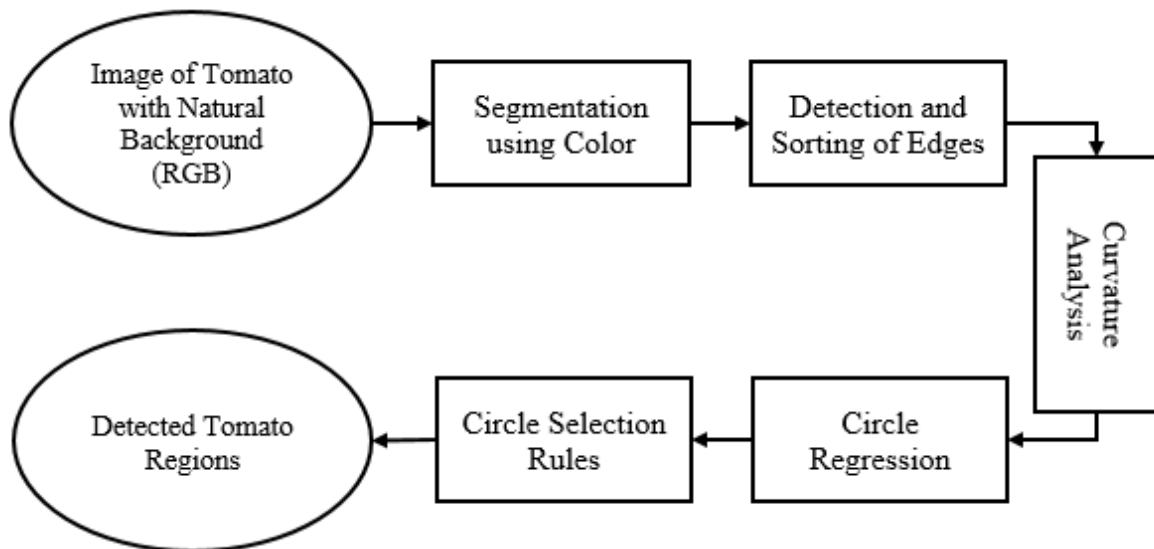


Fig. 2.2. Simplified process flow diagram of Xiang et al. (2013a) approach

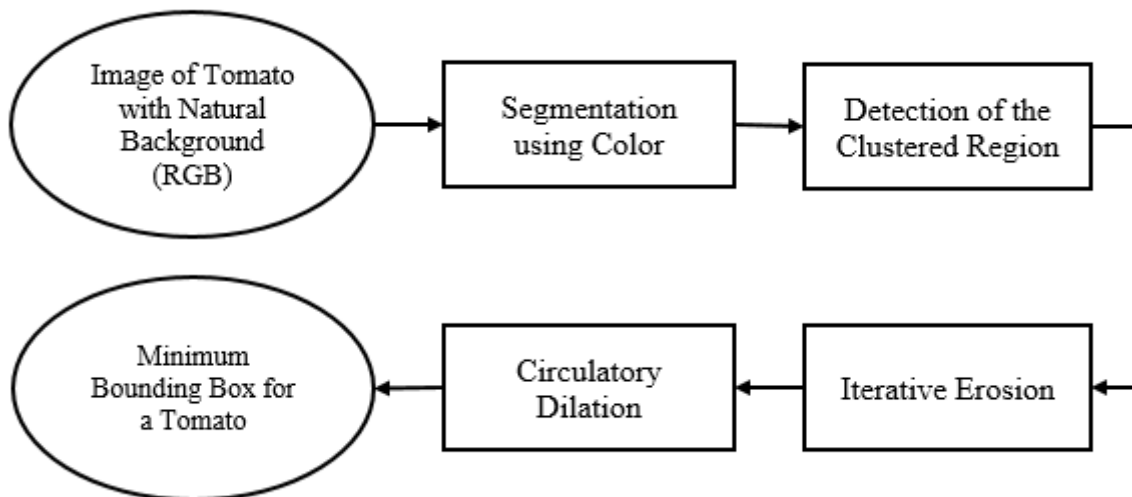


Fig. 2.3. Simplified process flow diagram of Xiang et al. (2013b) approach

Xiang et al. proposed another technique to detect clustered tomatoes [Xiang et al., 2013b] based on mathematical morphology. Fig. 2.3 shows the simplified process flow diagram of this approach. The tomato regions were segmented using normalized color differences. The longest side of the enclosing rectangle was selected as a clustered side. An iterative erosion was

performed on the segmented image to separate the clustered tomatoes. Finally, a circulatory dilation was done to recover from erosion for every seed region. The performance of this technique was good for adhered tomatoes or tomatoes with little occlusion. The performance of the overlapping tomatoes should be improved. The authors also observed that the shooting distance has very little impact on performance.

Gatica et al. proposed an olive fruit detection technique using a feed-forward neural network with a backpropagation learning algorithm [Gatica et al., 2013]. Their framework was based on two neural networks. The first network detects olive or not olive with features like eccentricity, area, size, and color features in RGB, HSV, Lab space. The second network was used for the selection of non-overlapping olives. The accuracy of the first network and second network are 97% and 88.8% respectively.

Nanaa et al. proposed a green mango fruit detection technique from the natural environment [Nanaa et al., 2014]. At first, the input RGB color image converted into a grayscale image to find the edges. The calculation of edge distance, morphology, and conversion to the binary image was done sequentially. They assumed that the shape of the mango is oval. The oval-shaped regions were detected as the potential places of mango by using RHT. A backpropagation neural network confirmed the final places of mango. The test was performed on 50 RGB images. The detection rate was 96.26% but it becomes low in case of an overlapping and occluded situation.

Xiang et al. proposed a detection technique of clustered tomatoes using binocular stereo vision technology [Xiang et al., 2014]. The depth map was acquired from the stereo vision camera. The depth map was denoised using an 8-neighbor mode. The type of clustering (adhering or overlapping) was detected from the depth difference of the front and back region using the iterative Otsu technique. The adhering tomatoes were detected by edge detection, curvature analysis, and circle regression on edge. The depth map segmentation, edge detection, and edge-sorting were done from iterative Otsu for the overlapping tomatoes. Then a similar process was applied for overlapping cases as applied for adhering type. The success rate was better for occluded tomatoes than adhering tomatoes. The success rate decreases with the increase of camera distance and the percentage of occlusion.

Lv et al. proposed a yellow apple segmentation [Lv et al., 2015] from the natural environment using image processing techniques. There is a significant color difference in apple foregrounds and background. The initial segmentation was done using k-means clustering on R+G-B value.

Then hole filling and noise removal operations were performed followed by a canny edge detection technique. Finally, a randomized Hough transform was applied on edges to detect the apple fruit region. The performance of this technique is limited to the fruit with only foliage. The segmentation from the complex background cannot be done with this technique.

Meng et al. proposed a detection technique for the overlapped apple from the natural environment [Meng et al., 2015]. At first, the image was converted to the color YUV model. Otsu thresholding was applied to the V component to segment the object from the background. The one-pixel boundary was extracted from the segmented image. The curvature of each boundary point was calculated. The boundary was broken into multiple parts by a curvature threshold value. The boundary parts are filtered based on the abrupt change in curvature. It was assumed that Apple is nearly circular. Hence, a circle fitting algorithm was applied to detect the individual apple from the overlapped situation. The overall accuracy is 90.6% but the performance is comparatively poor for the fruits located far from the camera or blocked by some obstacles than the fruits in close range or unblocked.

Sa et al. proposed an on-tree fruit detection technique using a deep convolutional neural network [Sa et al., 2016]. The vision system captures two modalities of image i.e. RGB color image and Near-Infrared image. VGG16 model had been fine-tuned for the detection of the fruit region. This model successfully does a pixel-level annotation on seven types of fruit. The f1 score of detection is 0.83.

Toon et al. proposed a deep learning model for a tomato harvesting robot [Toon et al., 2019]. The model contains two convolutional neural networks for making decisions of harvesting. The first network decides the tomato or non-tomato. The second network decides the ripe or unripe to harvest. The accuracy for the first and second networks is 76.37% and 98.79% respectively.

2.3 Sorting of Fresh and Non-fresh

Chandini et al. proposed a quality detection technique of fruits [Chandini et al., 2018]. Fig. 2.4. shows the simplified process flow diagram of this approach. At first, the input image has been resized and converted into HIS color space. The defective region was segmented using k-means clustering. The segmented image was converted into a grayscale version. Contrast, correlation, energy, and homogeneity were extracted from the GLCM. Multi-class SVM was used to classify fruit in the edible or inedible category. The model was trained with 75 apple fruits containing defective as well as healthy types. The system achieved 85.64% accuracy. The total

processing time was 0.68 sec which is acceptable for an automated system. The accuracy could be improved as well as the system needs to be tested with other types of fruits and vegetables.

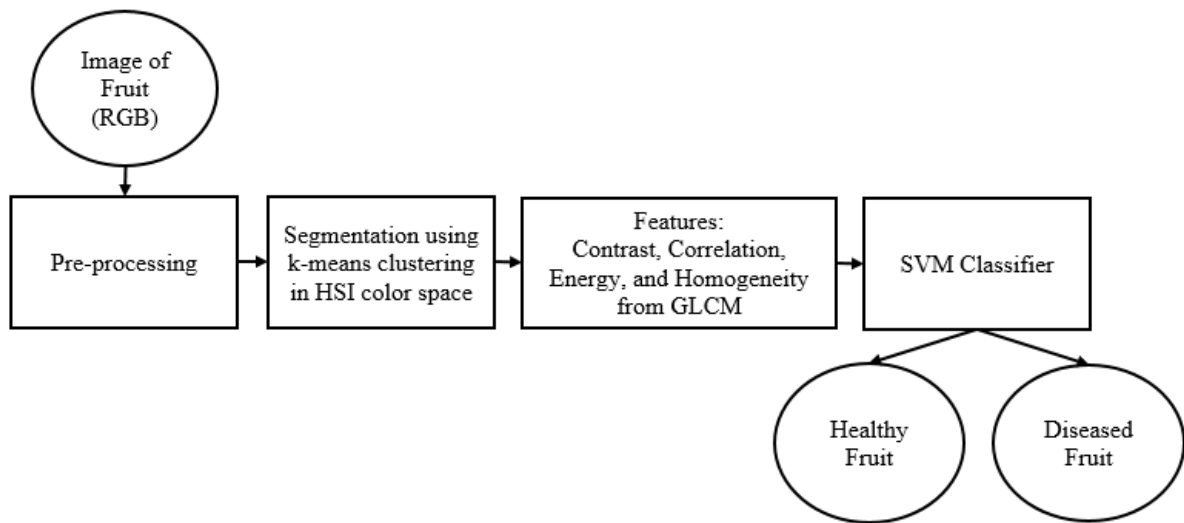


Fig. 2.4. Simplified process flow diagram of Chandini et al. (2018) approach

Karakaya et al. proposed a technique for the freshness classification of fruit [Karakaya et al., 2019]. Fig. 2.5. shows the simplified process flow diagram of this approach. Otsu thresholding was applied for segmenting the fruit object. The extracted features were grayscale histogram, energy, contrast, correlation and entropy from GLCM, and a bag of features. The SVM classifier was used for the classification of fresh and rotten fruit using those features.

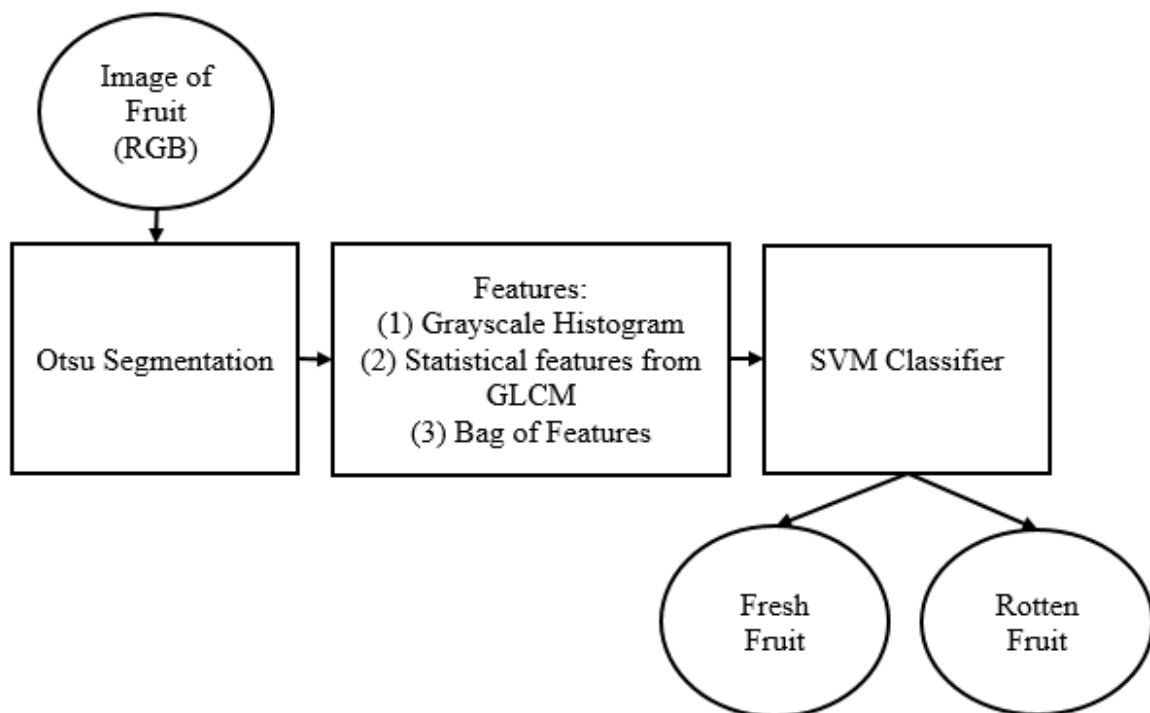


Fig. 2.5. Simplified process flow diagram of Karakaya et al. (2019) approach

Gómez-Sanchis et al. proposed an early detection of fungi in citrus fruits [Gómez-Sanchis et al., 2012]. The fungi, which belong to the *Penicillium* genus, cannot be detected with an open eye or normal RGB computer vision system. Hence, a hyperspectral imaging system was established to capture the image. The features were extracted from both spectral and spatial domains. The feature dimension had been reduced by the Minimum Redundancy Maximal Relevance method. Multilayer perceptron (MLP) neural network gives high classification accuracy.

Kamalakannan et al. proposed a technique for surface defect detection and classification of mandarin fruit [Kamalakannan et al., 2012]. Fuzzy thresholding was done on the grayscale version of the input image. Binary Wavelet Transform was applied to the segmented image. Moment invariant is extracted as a feature to classify with a linear rule-based model. At first, they detected the defective or not defective. Then classifies among three categories of defect i.e. splitting, pitting, and stem-end rot.

Capizzi et al. proposed a classification system among fresh, surface defect, morphological defect, color defect, and black mold of orange fruit using computer vision [Capizzi et al., 2015]. The input RGB image was converted to an HSV image. Five statistical features were extracted from GLCM for each channel in four directions. These statistical features were angular second moment, contrast, correlation, gradient module, and intensity symmetry. In total, 60 features were used in classification with a radial basis probabilistic neural network (RBPNN) classifier. The experimentation had been performed on a dataset of 400 images.

Ranjit et al. proposed a fruit disease classification technique [Ranjit et al., 2019]. The diseased region had been segmented using the k-means clustering technique. Shape (area, perimeter, major axis length, and minor-axis length), color (color mean in RGB, HSV, HIS color space), and texture (statistical features from GLCM) features were extracted from the diseased region. The classification performance using Neural Network classifier with those set of features was better than SVM and KNN classifier. Experimentation was done on a dataset of 2500 images, which was downloaded from the internet.

Dubey et al. proposed a two-stage defect detection model for fruits [Dubey et al., 2019]. Some pre-processing was done to remove noise and correcting illumination. The defective region was segmented and some region-oriented features were extracted. The features are no of objects, connectivity, area, perimeter, major axis, minor axis, convex area, diameter, eccentricity, filled area, solidity, and Euler number. In the first step, it detects the defective or not defective. In

the second step, it detects the stage of defect i.e. first stage, second stage, and the final stage. SVM classifier outperforms the other classifiers. A dataset of 2000 images of five types of fruit was used for experimentation.

Roy et al. analyzed some region segmentation techniques to segment the rotten region [Roy et al., 2019a]. The segmentation methods, which were explored, are (a) marker-based segmentation (b) color-based segmentation (c) edge detection. The edge detection technique was given the most accurate results in comparison with other techniques.

Roy et al. also proposed a segmentation technique of rotten fruit [Roy et al., 2019b]. The foreground was segmented in YCbCr color space. K-means clustering was done in Lab color space. A mask was generated for segmenting rotten regions based on clustering results.

da Costa et al. proposed a model for external defect detection of tomato fruit using deep learning [da Costa et al, 2020]. ResNet50 was fine-tuned and trained on a dataset of 43843 images. The superiority of this model is that the model does not confuse stem or calyx with external defects.

2.4 Classification of Fruits and Vegetables

Seng et al. proposed a fruit classification system using color and shape features [Seng et al., 2009]. Fig. 2.6. shows the simplified process flow diagram of this approach. The area, perimeter, roundness was used as shape features. The mean of RGB channels was used as a color feature. A dataset of 50 images was used for experimentation. kNN classifier gives a good classification accuracy of 90% on this dataset. The system can be tested with more images and more classes.

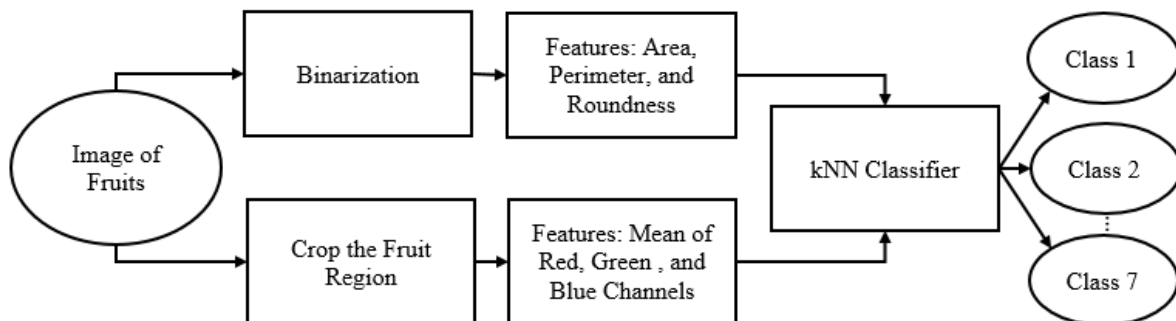


Fig. 2.6. Simplified process flow diagram of Seng et al. (2009) approach

Roomi et al. proposed an intra-class classification technique for mango fruit using shape-oriented features [Roomi et al., 2012]. Fig. 2.7. shows the simplified process flow diagram of

this approach. The ROI was segmented from using the Otsu technique in HSV color space. The features for classification were eccentricity, the division of major axis length and minor axis length, the ratio of area and square of the perimeter. The experimentation was done with three types of south Indian mango i.e. Totapuri, Alfonza, and Rumani. The classification accuracy is 90.91% using the Naïve Bayes classifier.

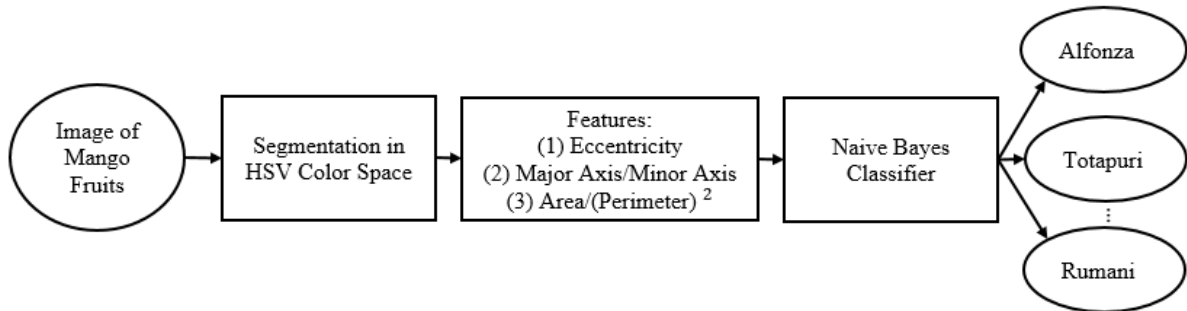


Fig. 2.7. Simplified process flow diagram of Roomi et al. (2012) approach

Cornejo et al. proposed a histogram-based fruit and vegetable classification system [Cornejo et al., 2016]. Fig. 2.8. shows the simplified process flow diagram of this approach. They had passed the input image through some pre-processing steps to segment the fruit or vegetable from the background. The extracted features were Census Transform Histogram (CENTRIST), and Hue-Saturation Histogram. Principal Component Analysis (PCA) reduced the feature dimension. The experimentation was performed on a dataset containing 2633 images of 15 fruit and vegetable categories. The SVM classifier gives maximum accuracy of 97.23% on this dataset with this set of features.

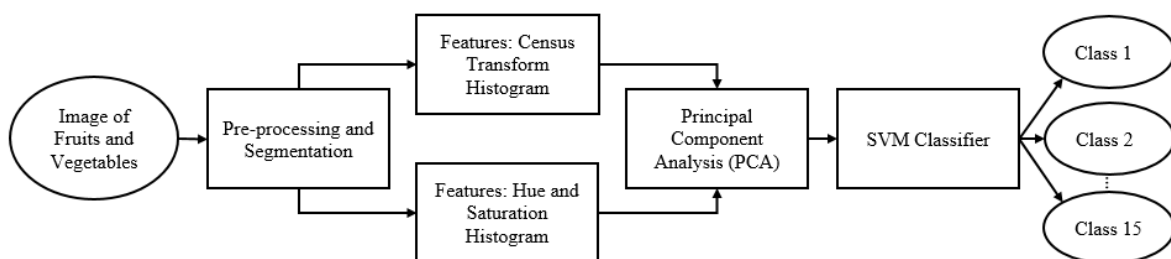


Fig. 2.8. Simplified process flow diagram of Cornejo et al. (2016) approach

Rocha et al. proposed a fruit classification system combining multiple features [Rocha et al., 2010]. The features were global color histogram, color coherence vector, Unser’s descriptor, appearance descriptor, and border/interior pixel classification. They have tested with multiple classifiers i.e. SVM, kNN, Linear Discriminant Analysis (LDA), etc.

Arivazhagan et al. proposed a fruit classification using a combination of color and texture features [Arivazhagan et al., 2010]. The color RGB image was converted into an HSV image. The co-occurrence matrix is formed from the *V* channel and five texture features were extracted from this matrix i.e. contrast, energy, homogeneity, cluster shade, and cluster prominence. Mean, standard deviation, skewness, and kurtosis were extracted from both the *H* and *S* channels as color features. The minimum distance classifier was used as a classification algorithm. The experimentation was done on a dataset of 2635 images of 15 types of fruits.

Zhang et al. proposed a feature fusion approach to classifying multiple fruits with kernel support vector machine (kSVM) [Zhang et al., 2012]. The split and merge technique was used to segment fruit from the background. They had extracted a 64 bin color histogram from the RGB image. 7 indexes were extracted as a texture feature from Unser's sum and difference histogram i.e. mean, contrast, energy, homogeneity, correlation, variance, and entropy. The extracted shape features were area, perimeter, Euler number, convex hull, solidity, major axis length, minor axis length, and eccentricity. PCA was applied to reduce the feature dimension from 79 to 14. The dataset contains 1653 images of 18 fruit classes.

Dubey et al. proposed a texture-based classification for 15 fruits and vegetable classes [Dubey et al., 2012]. The sum and difference of the intensity of the neighboring pixel were extracted from the segmented color image. A multi-class SVM gives the maximum classification accuracy on the dataset.

Haidar et al. proposed a system for the classification of 7 types of Date fruit [Haidar et al., 2012]. The classification features were color (mean and standard deviation of 3 color channels of RGB), texture (entropy of 3 channels of RGB, energy from GLCM), shape and size (major axis length, minor axis length, area, perimeter, eccentricity). Artificial Neural Network gives the best accuracy among the classifiers used by them.

Chowdhury et al. proposed a fruit and vegetable classification system for blind people [Chowdhury et al., 2013]. The visual features like shape, size, color, and texture were used with a classification algorithm for this task. The system will help blind people to buy fruits and vegetables from the store in a supermarket.

Ninawe et al. proposed an approach for fruit classification using a combination of texture, shape, and color features [Ninawe et al., 2014]. The region of interest was cropped before feature extraction. The entropy was computed as a texture feature. The area, perimeter, and

roundness were the shape features. The mean value of each color channel was computed as the color feature. kNN classifier was used to classify six types of fruit with this set of features.

Wang et al. proposed a feature fusion technique for fruit classification [Wang et al., 2014]. The fruit object was segmented from the background after some pre-processing steps. The color features were extracted from the HSV color space. The LBP was extracted as a texture feature. The extracted shape features were perimeter, area, and roundness. Finally, a fusion of decisions was made from individual backpropagation neural network classifiers with a specific type of features i.e. color, shape, and texture. The approach was tested on a dataset of 300 images of three classes. They stated that the classification of fruit images captured from a different angle and different illumination can be researched further.

Zawbaa et al. proposed a fruit classification system using two sets of features and a Random Forest Classifier [Zawbaa et al., 2014]. The first set contains color features like mean, variance, skewness, kurtosis, and shape features like eccentricity, Euler number, centroid. The second set contains the Scale Invariant Feature Transform (SIFT) feature. The system was developed in Matlab and tested on 178 images of 3 types of fruits i.e. apple, orange, and strawberry.

Vogl et al. proposed a classification technique for fruits in a mobile environment [Vogl et al., 2014; Kim et al., 2014]. The normal color, shape, and texture features were converted into code as a search key in the feature database. The evaluation strategies had been optimized using a Genetic Algorithm. In total, 1108 images of 36 different classes had been used for experimentation.

Naskar et al. proposed a fruit classification technique using multiple features and Artificial Neural Network (ANN) [Naskar et al., 2015]. They had extracted fractal dimension as a texture feature of the image after passing through a log Gabor filter, mean hue of ROI as color and area, perimeter as a shape feature. The system successfully classified six types of fruits.

Kuang et al. proposed a classification technique for fruits using a multi-feature fusion technique and LibSVM classifier [Kuang et al., 2015]. The features were extracted for multiple color channels i.e. global color histogram, Histogram Oriented Gradient (HOG), Local Binary Pattern (LBP), and LBP based on Gabor wavelet. They experimented on 1778 images of 5 classes with variety in background, location, size, and angles. The complex background and uneven illumination created misclassification in this approach.

Rachmawati et al. proposed a classification system for 32 classes of fruits [Rachmawati et al., 2015a]. The n bin color histogram of the RGB image was used as a classification feature in k NN classifier. They used the RGBD object dataset for testing.

Rachmawati et al. proposed another technique to develop a color palette for fruit classification [Rachmawati et al., 2015b]. The color samples were taken from an ROI of different fruit types. k -means clustering was applied to build the palette. The test sample with minimum distance from the centroid of a particular cluster was chosen as the class of fruit. 7 type of fruit was chosen from RGBD object dataset for testing. The uneven illumination and non-fruit pixels of ROI created an error in cluster formation. The fruits with multiple dominant colors also create a challenge. The palette was developed in RGB color space. The other color space can be explored in the future.

Al-falluji proposed a combined approach for classification using color, shape, and texture features [Al-falluji, 2016]. The mean, variance in RGB color space was used as color features. The extracted shape features were area, perimeter, eccentricity, Equiv diameter, major axis length, and minor axis length. The contrast, correlation, energy, and homogeneity from GLCM were used as a texture feature. SVM gives maximum accuracy with those features for the classification of five types of fruit.

Ronald et al. proposed an approach for the classification of three types of apple fruit [Ronald et al., 2016]. The RGB image was converted to a grayscale image. Otsu thresholding was applied to segment the object from the background. The noises were removed using an averaging filter. The color and size features were extracted and a Naïve Bayes classifier was trained with those features for classification. The dataset contains 150 images of three types of apple.

Hossain et al. proposed a fruit classification system as an industrial application using deep learning [Hossain et al., 2018]. They have modeled a convolutional neural network with six convolution layers. Another model has been proposed by fine-tuning the existing visual geometry group-16 (VGG-16) model. The experimentation was performed on two datasets. The first one was a publicly available dataset of 2633 images of 15 fruit types. The second one, which was collected by them from the internet, contains 5946 images of 10 classes.

A fruit classification system has been proposed for smart inventory management of refrigerators and avoiding last-minute rush [Desai, 2019]. The computer vision system will classify the type of fruit as well as count the number for a particular type of fruit. There will be

a threshold for restocking every item. When the count goes below the threshold value, the system will give an alert to the user through a smartphone or giving a notification or order to the vendor. The system will segment and crop the fruit by a bounding box. The shape features i.e. area, perimeter, eccentricity, and centroid, are used to identify a specific type of fruit. The experimentation was done with apples, bananas, and oranges. There are some disadvantages of this proposed technique. These are (a) the overlapping of fruit for reducing the storage (b) the fruits which are kept clustered e.g. grapefruit. The retrieval of count and proper shape features becomes difficult for those cases.

2.5 Volume and Mass Estimation

Venkatesh et al. proposed a novel volume and mass estimation technique for axisymmetric fruits [Venkatesh et al., 2015]. Fig. 2.9 shows the simplified process flow diagram of this approach. The RGB input image is converted to a binary image using Otsu thresholding. The image is rotated to place stem-calyx along the horizontal or vertical axis. The boundary contour was extracted from the binary image. Some shape-oriented parameters were measured from the fruit boundary points. They had set three criteria with those parameters to determine the shape of the fruit. The combination of values for those criteria categorizes a fruit into three types of shape i.e. circular, parabolic, and elliptical. The volume was estimated by the formula of the corresponding shape type. Mass was estimated from the mass-volume relationship. The experimentation was done on sweet lime, orange, lemon, and apple. The estimated volume was compared with the measured volume using the water displacement method.

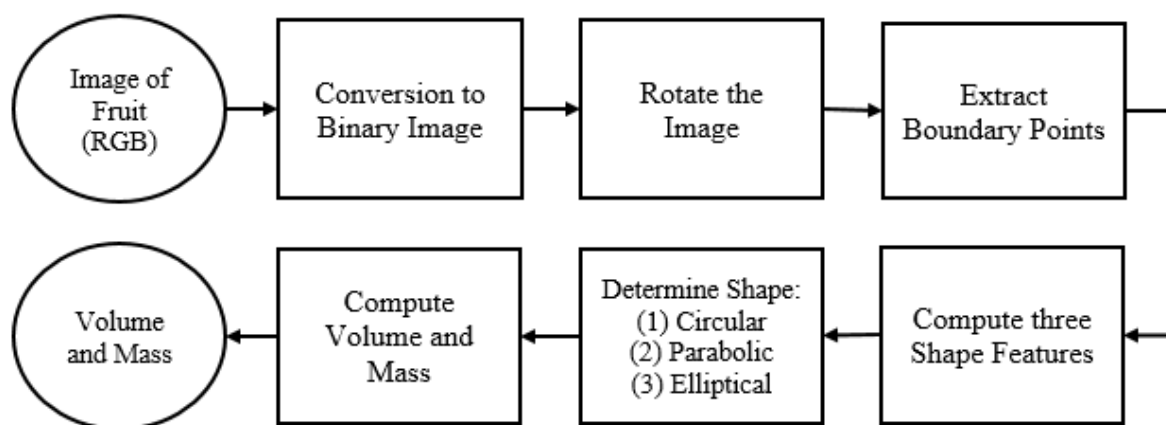


Fig. 2.9. Simplified process flow diagram of Venkatesh et al. (2015) approach

Uluişik et al. proposed a volume estimation approach for tomatoes using computer vision [Uluişik et al., 2018]. Fig. 2.10 shows the simplified process flow diagram of this approach.

An imaging setup was created with five high-resolution cameras to capture RGB images of the tomato. The image was converted to grayscale followed by segmentation using thresholding. Morphological dilation and filling of holes were done to improve segmentation. Edge detection was done using the Canny technique. The fruit was divided into multiple conical frustums along the major axis. The final volume is the sum of volumes of all frustums. The outcome of the proposed technique was compared with the volume, which is computed theoretically assuming tomato as an elliptical shape. They concluded that the shadow under the fruit is a challenge for accurate estimation of volume.

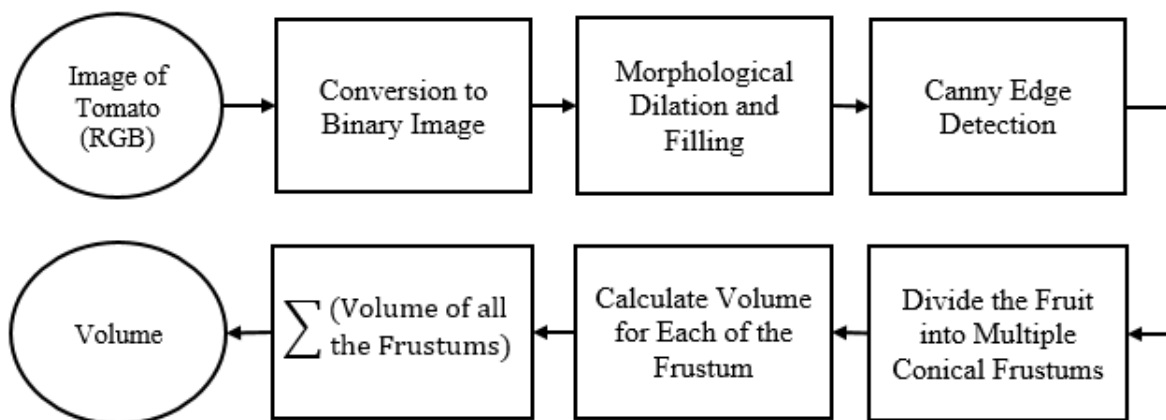


Fig. 2.10. Simplified process flow diagram of Uluişik et al. (2018) approach

Sabliov et al. proposed volume and surface area estimation techniques for axisymmetric ellipsoidal agricultural products [Sabliov et al., 2002]. The products were assumed as a sum of frustums of right circular cones. The sum of volume and surface area of its frustums is the final volume and surface area of the product respectively. They had tested their approach on products like egg, lemon, lime, peaches. The estimated volume was compared with the measured volume using the water displacement method.

Omid et al. proposed a volume and mass estimation technique for citrus fruits [Omid et al., 2010]. Two perpendicular camera setup was established to get radius from two views. The image was segmented from the background. The segmented object was divided into multiple elliptical frustums. The volume of a frustum was calculated by the formula. The volume of all the frustum was summed to get the volume of the entire object. The performance was compared with volume measured using the water displacement method. The weight was estimated from the correlation of estimated volume and actual weight. Fruit size does not have any effect on the performance of this technique.

Khojastehnazhand et al. proposed a volume estimation technique for tangerine fruit [Khojastehnazhand et al., 2010]. A double camera imaging setup was established in a perpendicular position. Three dimensions were measured from those two cameras. The volume was measured by approximating tangerine as an ellipsoid. They also proposed another method by dividing tangerine into multiple frustums. The volume of the frustums was summed to get the final volume.

Iqbal et al. proposed a volume estimation technique for axisymmetric apple fruit using a single view image [Iqbal et al., 2011]. The known factors were – (a) conversion factor in pixels/mm (b) fruit is shadow-free and glare-free (c) fruit is aligned horizontally along the stem-calyx axis (d) fruit is positioned at the center of the image. The apple fruit was categorized into circular, elliptical, and parabolic based on the combination of three shapes oriented criteria. The volume was obtained using the corresponding mathematical formula of volume estimation i.e. spherical, ellipsoid, paraboloid.

Siswanto et al. proposed a framework for volume measurement of irregularly shaped fruits [Siswanto et al., 2013]. The framework was built with the Monte Carlo method and computer vision. The bounding box was drawn to enclose the segmented object. The volume of the bounding box is V_0 . There is N number of random 3D points inside the bounding box in the real-world coordinate system. N_0 number of random points is contained by the fruit object. The volume of fruit is $(V) = V_0 * N_0 / N$.

Sa'ad et al. proposed a weight estimation technique for grading Harumanis mangoes [Sa'ad et al., 2015]. At first, the input image had been binarized using a threshold value. The morphological filling was performed to fill the holes inside the white object region. The centroid and boundary were extracted to estimate the volume of mango by cylindrical approximation. The weights were estimated from the scatter plot between actual weight and estimated volume.

Gokul et al. proposed a volume measurement technique for sweet lime [Gokul et al., 2015]. The input image was converted to grayscale followed by blurring. The edge of the object was extracted to measure the radius. The shape of the sweet lime was assumed as spherical. The volume was estimated using the formula of volume calculation of the sphere.

Dhameliya et al. created a dataset for the volume extraction of mango [Dhameliya et al., 2016]. They had extracted binarized shapes from every image but did not propose any solution for volume estimation.

Siswanto et al. proposed a volume measurement technique for axisymmetric food products [Siswanto et al., 2016]. The captured image was converted to a grayscale image and passed through the Gaussian filter to remove noises. Otsu thresholding was used to segment the object from the background followed by morphological opening and closing operation. The pixels are converted into the world coordinate system after cropping the ROI. The object boundary was extracted to construct a piecewise cubic polynomial. Volume was approximated by integrating this polynomial.

Concha-Meyer et al. proposed volume estimation techniques of strawberries, mushrooms, and tomatoes [Concha-Meyer et al., 2018]. Volume was determined by the surface fitting and wireframe model. The estimated volume using machine vision was compared with the water displacement method. A relationship was established between the estimated volume and actual weight.

2.6 Supplier Selection

In this literature survey, only Lin et al. [Lin et al., 2011] directly deals with the supplier selection problem of fruits and vegetable stores. Hence, some other paper has been surveyed which deals with the supplier selection problem of different industries.

Lin et al. proposed a fruits and vegetables supplier selection technique for the supermarket in Taiwan. [Lin et al., 2011]. The authors have identified multiple criteria for supplier selection. The initial screening has been done using a fuzzy Delphi method. The authors computed relative priority using the fuzzy Analytic Hierarchy Process.

Mohammed et al. proposed another two-stage supplier selection technique [Mohammed et al., 2017]. In the first stage, fuzzy TOPSIS is used to rank the suppliers. In the second stage, the optimum quantity of products is selected using multi-objective optimization by maximizing the social impacts as well as minimizing cost, travel time, and environmental impacts. The authors did a case study on livestock and meat packet suppliers.

Krishankumar et al. proposed a two-phase supplier selection technique for an automobile industry [Krishankumar et al., 2017]. In the first phase, a linguistic-based aggregation is applied for converting the linguistic terms of decision-makers directly into the aggregate linguistic rating. This direct aggregation technique preserves the information. In the next phase, an extended preference ranking organization method for enrichment evaluation (PROMETHEE) was applied for ranking the alternatives in a fuzzy environment.

Mananawigapol et al. proposed a hybrid supplier selection technique for the printing business [Mananawigapol et al, 2018]. In the first phase, the sustainable criteria are selected using SWOT analysis. Then weights of the criteria are extracted by using AHP. Finally, the rank of the suppliers is generated by using the TOPSIS.

Koganti et al. proposed another hybrid supplier selection approach for the small-scale industry of south India [Koganti et al., 2019]. They combined grey relational analysis (GRA), AHP, and TOPSIS. GRA shortlists the criteria, AHP calculates weights and TOPSIS ranks the suppliers.

2.7 Summary

This chapter describes the prior as well as state-of-the-art work for each of the problems using computer vision and machine learning techniques. The segmentation of fruits and vegetables from the natural environment is mostly done based on the color difference of the object and background. Then the binarized image passes through some morphological processing stages to make the foreground more accurate. Edge or boundary points are extracted from the segmented object region. Hough Transform or circle regression is applied on the edges to detect each fruit or vegetable present on the image. This type of algorithm suffers from irregularly shaped fruits and vegetables. The challenge increases when there is occlusion. The occlusion may occur due to the appearance of stem, leaves, or overlapping by other fruits. There are few attempts to address this challenge but there is a lot of scope for improvement.

The defective and rotten fruits and vegetables are harmful to health as well as it degrades the quality. The sorting of those defective or rotten fruits and vegetables has been done using image analysis and machine learning techniques. The problem has been solved as a binary classification of fresh and not fresh (defective, rotten, etc.). The defective or rotten regions are segmented using unsupervised clustering techniques. The extracted features from ROI are texture (statistical features from GLCM i.e. correlation, contrast, energy, and homogeneity), color (mean and histogram of different channels in different color space), shape (area, perimeter, major axis length, minor axis length, Euler number, convex area, etc.) and bag of features. The detection of defective and rotten are done by supervised machine learning algorithms i.e. Support Vector Machine, *k*NN, Neural Network, etc.

The classification of different types of fruit and vegetables is mostly addressed using supervised machine learning. The flow includes stages like pre-processing, feature extraction, and classification algorithm. The pre-processing includes segmentation, noise removal,

morphological operations, edge detection, etc. The researchers extracted visual features like texture, color, shape, and size from the fruit or vegetable image. The most used texture features are statistics (correlation, contrast, energy, homogeneity, entropy, etc.) from GLCM, LBP, Unser's sum and difference histogram, Gabor filter, HOG, etc. The color features, which was used in previous approaches, are mean, standard deviation, skewness, kurtosis, and histogram for different color channels in different color space. The most popular shape features for fruit and vegetable classification are area, perimeter, eccentricity, roundness, major axis length, minor axis length, etc. The machine learning models are trained with those features to predict the class label for unknown samples. The machine learning algorithms, which are mostly experimented for this problem, are k NN, Naïve Bayes, Support Vector Machine, Random Forest, Neural Network, etc. The experimentation was done with a very less number of classes in most of the works.

Volume and mass estimation by image analysis are explored here. Volume estimation of axisymmetric and regular shaped (spherical, ellipsoid, paraboloid, and cylindrical) fruits and vegetables are proposed by most of the researchers. The fruit and ROI are detected by segmentation, binarization, morphological processing, edge detection, etc. The volume calculation parameters are measured from the ROI. The volume is estimated using the mathematical formula for the corresponding shape type. In some approaches, the researchers divided the fruit and vegetable region into multiple frustums. The volumes of the frustums are calculated individually and finally summed to get the volume of the entire fruit or vegetable.

The appropriate choice of supplier is a very tough decision for the managerial personnel of any industry. The supplier selection in fruits and vegetable store is very important not only for high profit but also to reduce the wastage inside store and customer satisfaction. The supplier evaluation is done based on multiple criteria. Hence, the criteria are identified in the first stage then the ranking of supplier is done by using multi-criteria decision making techniques.

Segmentation and Detection from Natural Environment

3.1 Introduction

Harvesting of agricultural products i.e. fruits and vegetables is a very important task in cultivation. The manual harvesting of agricultural products is a very time-consuming task. The manual resources also need to be trained with the knowledge about the maturity of fruits and vegetables. Early harvesting may not give the proper taste or weight of fruits and vegetables. Sometimes delay in harvesting produces damaged or overripe fruits and vegetables, which is very critical to preserve and transport. Hence, the timing of harvesting is very crucial for the quality of fruits and vegetables. The shortage of manual resources in the harvesting period makes a big impact on business. Automated harvesting is very much needed to cope up with those situations for competing with a rapidly growing global market. Automated harvesting using computer vision can be very effective. The primary task for automated harvesting using computer vision is to segment the fruits and vegetables from natural backgrounds and adjusting the size of picking the hand of a harvesting robot [Feng et al., 2015] according to the fruit or vegetable size. The natural background may contain leaf, stem, sky, etc. as shown in Fig. 3.1. The prior approaches mostly used color-based segmentation for a particular type of fruit and vegetable. The task becomes very difficult when the fruit or vegetable object is nearly similar in color with backgrounds. The unripe mango is an example of this condition. A color-based segmentation algorithm cannot be applied to other fruits and vegetables with different colors.

Those obstacles cause inaccurate segmentation. This chapter addresses the challenges using a very efficient segmentation and region of interest (ROI) detection technique.



Fig. 3.1. Fruits under natural environment

3.2 Proposed Method

The segmentation technique, which is used here, is mainly based on graph theory. The input to the system is an RGB color image of any fruit or vegetable (I_{RGB}) with a background of the natural environment. This work aims to segment the fruit and vegetable from the natural background with ROI. Fig. 3.2 depicts the flow of the proposed technique. The proposed technique has two steps i.e. fruits and vegetable segmentation and ROI detection. In the first step, the input image (I_{RGB}) has been segmented using GrabCut. In the second step, the segmented image (I_S) has been passed through an ROI extraction module. The final output is a cropped ROI (I_{ROI}) and the parameters of minimum bounding box i.e. x and y coordinate top-left corner (x_0, y_0), width (w), height (h).

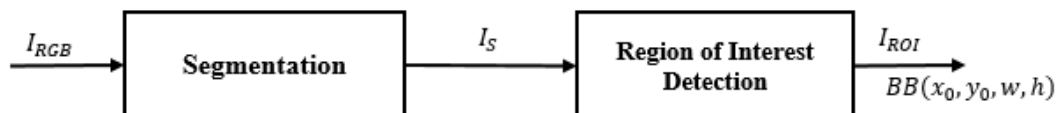


Fig. 3.2. Process flow diagram of the proposed technique

3.2.1 Fruits and Vegetable Segmentation

The first task of automated harvesting is to segment the fruit or vegetable object from the background. GrabCut [Rother et al., 2004] has been adopted here for doing the segmentation. This algorithm has been introduced by Rother et al. of Microsoft Research Cambridge, UK., It is a very efficient algorithm and requires very little interaction from the user. It contains two components. The first component is required for hard labeling of image pixels and the other component does a foreground and background classification based on the Gaussian mixture model (GMM). The other segmentation techniques use either region information or edge information. The GrabCut uses both the region and edge information. The information will

create an energy function. The best segmentation result will be generated if this energy function is minimized. Fig. 3.3 depicts the process flow of GrabCut. The human interaction would be a rectangle to locate the object. Outside the rectangular region, the pixels are sure short background. Inside the rectangle is unknown. The iterative process is applied to the rectangular region to segment the object properly. The steps performed in this algorithm are described here in detail.

Step 1: A rectangular region is selected for hard labeling. The pixels outside this region completely belong to the background. The pixels inside the rectangle are unknown. The hard labeling means this marking will not change further. The region inside the rectangle is the mixture of foreground and background.

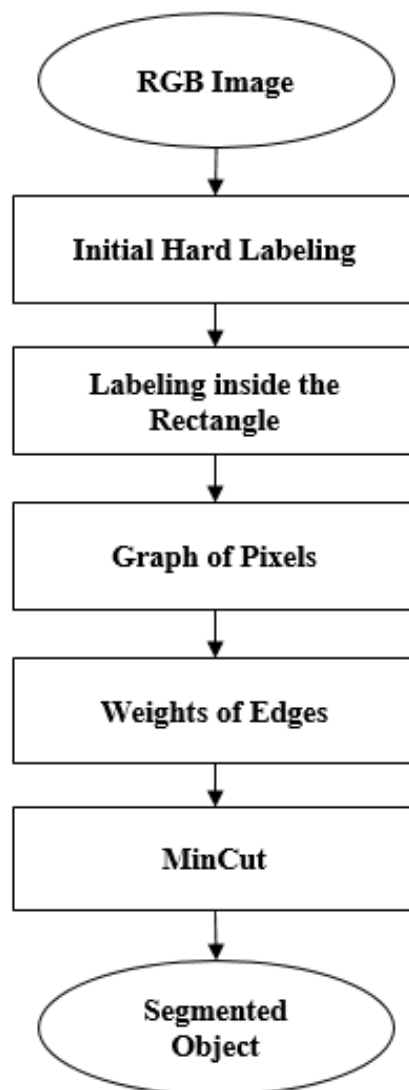


Fig. 3.3. Process flow of segmentation (GrabCut)

Step 2: Gaussian Mixture Model is applied to learn new pixel distribution in RGB color space based on hard labeling from the human. The pixels inside the rectangular region are labeled as probable object pixel or probable background pixel on the basis of the relation with hard labeled background pixel. The relation is established in terms of the closeness of color statistics. This labeling is one kind of color-based clustering.

Two GMMs are taken here, one for the foreground and another for the background. Each of the GMM contains five clusters ($K=5$, K is the number of the cluster) in total. The GMMs for foreground and background is represented by $\alpha_n = 1$ and $\alpha_n = 0$ respectively. This representation is completely based on the initial hard labeling. The probability distribution of a GMM is defined by multiple parameters. Every cluster of GMM is represented by three parameters. Refer to Eq. (3.1). Here, μ is the mean RGB value, π is the weighting coefficient, and Σ is a 3 x 3 covariance matrix. A trimap (T) is created for the segmentation. Refer to Eq. (3.2).

$$\theta = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \quad \text{where } \alpha = 0 \text{ or } 1 \text{ and } k = 1, 2, \dots, K\} \quad 3.1$$

$$T = \{T_B, T_U, T_F\} \quad 3.2$$

The background and foreground pixels are stored in T_B and T_F respectively. The unknown pixels are stored in T_U . The T_B is initialized by the pixels outside the rectangle, which is drawn in *step 1* for hard labeling. The unknown pixels are initialized by $T_U = \overline{T_B}$. The foreground pixels are initialized as $T_F = \emptyset$. The GMM for the background is formed by T_B with $\alpha_n=0$. The GMM for foreground is formed by T_U with $\alpha_n=1$. z_n are the pixels in RGB color space. The pixels in T_U will be assigned to clusters by using the minimum energy function. Refer to Eq. (3.3). The data term U can be defined as in Eq. (3.4) using color GMMs. $D(\alpha_n, k_n, \theta, z_n)$ is the Gaussian probability distribution function. Refer to Eq. (3.5). The smoothness term V remains unchanged for monochrome images unless the contrast is computed using Euclidean distance.

$$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z) \quad 3.3$$

$$U(\alpha, k, \theta, z) = \sum_n D(\alpha_n, k_n, \theta, z_n) \quad 3.4$$

$$\begin{aligned}
D(\alpha_n, k_n, \theta, z_n) &= -\log \pi(\alpha_n, k_n) \\
&+ \frac{1}{2} \log \det \Sigma(\alpha_n, k_n) + \frac{1}{2} [z_n - \mu(\alpha_n, k_n)] \Sigma(\alpha_n, k_n)^{-1} [z_n \\
&- \mu(\alpha_n, k_n)]
\end{aligned} \tag{3.5}$$

Now, the labeling of background and foreground pixels will be refined iteratively. The assignment of pixels to the closest cluster for foreground and background GMM refines the distribution of foreground and background pixels. The iteration occurs until the E is minimized.

Step 3: A graph (G), Eq. (3.6), is generated from this color distribution where V is the pixels and E is the edges connecting two neighboring pixels. Two extra nodes are added i.e. “Source” and “Sink”. The “Source” is marked “ S ” and “Sink” is marked as “ T ”. The modified set of the vertex is shown in Eq. (3.7). All the pixels are connected with the “Source” as it is the root node object pixels and the “Sink” as it is the root node of background pixels. The modified set of edges is E' . The “Source” and “Sink” should be separated properly from the modified graph (G'), Eq. (3.8), for the best segmentation result.

$$G = (V, E) \tag{3.6}$$

$$V' = (V) \cup \{S, T\} \tag{3.7}$$

$$G' = (V', E') \tag{3.8}$$

Step 4: The weights of the edges, which connect pixels with either “Source” or “Sink”, are determined by the probability to be foreground or background of those pixels. The edge information or pixel similarity defines the weights between the pixels. Weights are calculated from the refined GMMs. The weights between various edges are represented as-

- The weight of the edge between pixel i and j is w_{ij} .
- The weight of the edge between the S node and pixel i is a_i .
- The weight of the edge between pixel j and node T is b_j .

Fig. 3.4 represents the graph of pixels for segmentation. The weight of an edge will be low if the connecting pixels of that edge have a large intensity difference. That means there is a strong indication of edge. The weights will be high if there are similarities in pixel intensities. The cut will pass through the edges where the total weight of the cut is minimum.

Step 5: A mincut algorithm is applied to the graph. The mincut will segment the graph into two parts (i.e. “Source” node and “Sink” node) based on a cost function. The cost function is determined by the sum of the weights of all the edges that will be cut. Here, the capacity or cost of the S - T cut (A, B) is computed using Eq. (3.9), where A and B are the set of foreground pixels and background pixel respectively. A includes node S and B includes node T .

$$cut(A, B) = \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{(i,j) \in E'} w_{ij} \quad 3.9$$

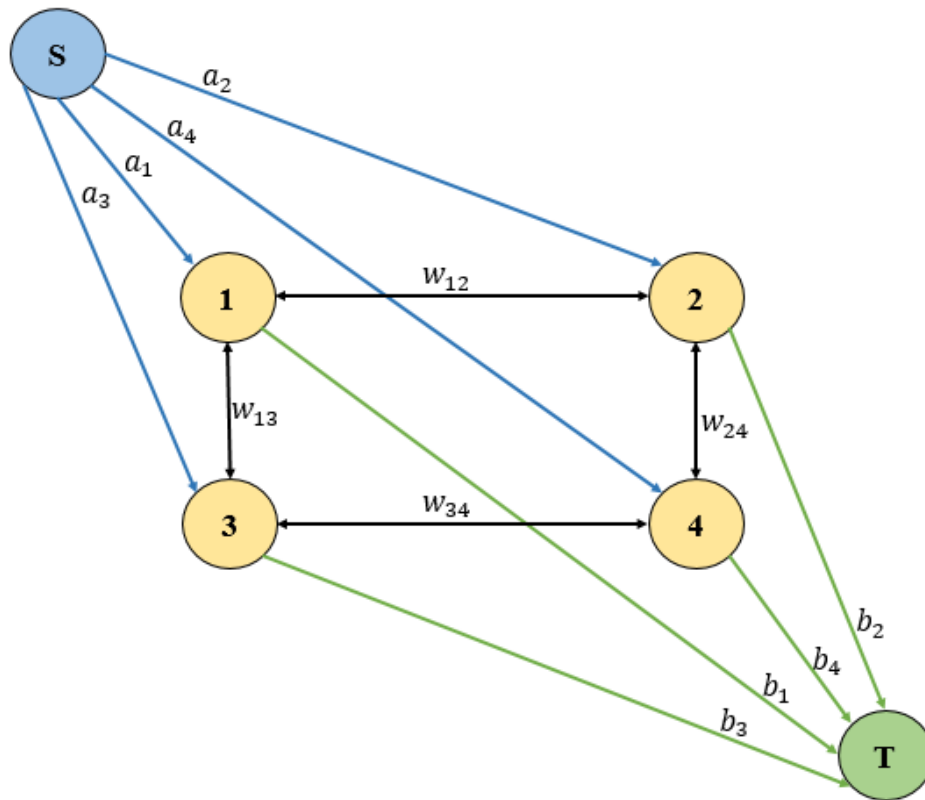


Fig. 3.4. Graph of pixels

The edges of this cut can be of three types. These are-

- The edge between i and T and $i \in A$. It contributes b_i to the cost function for making i as a foreground pixel.
- The edge between S and j and $j \in B$. It contributes a_j to the cost function for making j as a background pixel.
- The edge between i and j and $i \in A, j \in B$. It contributes w_{ij} to the cost function for separating the pixels i and j .

The motive of this segmentation is to minimize the cost function. The minimum value of $cut(A, B)$ makes the best segmentation result between foreground and background pixels.

Step 6: The iteration continues until it reached a convergence. The algorithm works very well for complex background segmentation because of this initial hard labeling. Fig. 3.5 visualizes the simplified steps of GrabCut.

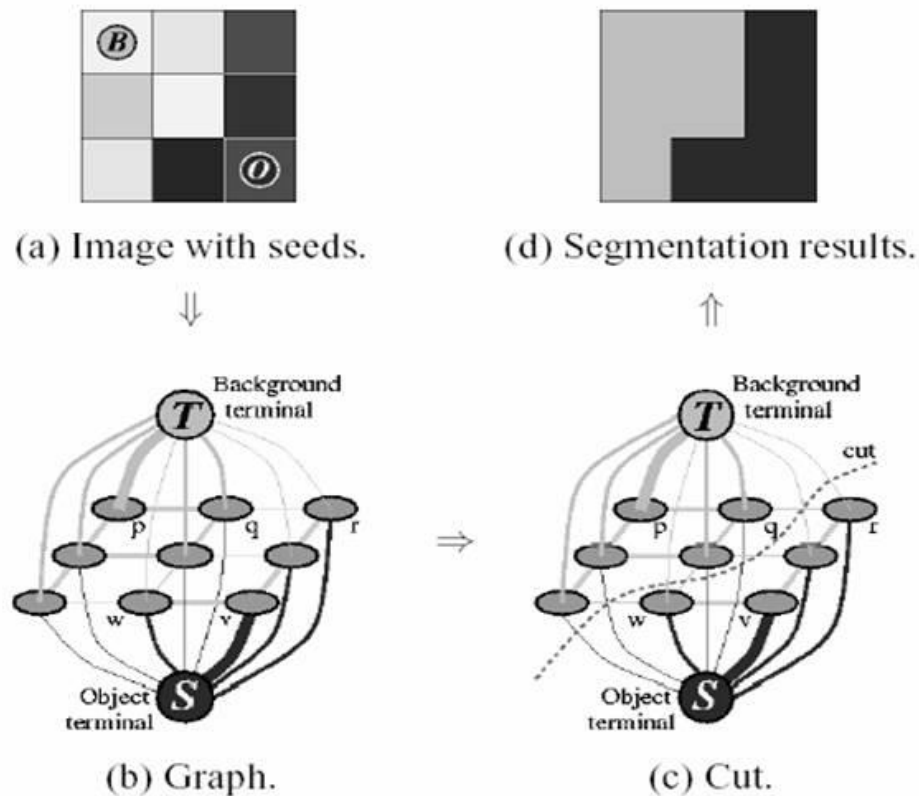


Fig. 3.5. Simplified steps of GrabCut [Marsh et al., 2020]

3.2.2 Region of Interest (ROI) Detection

Proper ROI detection is the main goal of automatic harvesting. The GrabCut may generate more than one region as the foreground. The foreground and background pixels are replaced by 1 and 0 respectively. A binary image is generated with the white object regions. The binary image contains a set of foreground regions (FR). Refer to Eq. (3.10). It has been observed after segmentation that there are some small regions present in the binary image along with the object region. The region area (RA) is measured in the number of pixels for each of the foreground regions. The region with maximum area is selected as the actual object region and the rest of the regions are discarded. Refer to Eq. (3.11). Finally, a bounding box is created to enclose the object region.

$$FR = \{fr_1, fr_2, \dots, fr_n\} \quad 3.10$$

$$ROI = \max\{ra_1, ra_2, \dots, ra_n\} \quad 3.11$$

3.3 Dataset

A dataset has been created for doing experimentation with the proposed methodology. The dataset is formed with the downloaded images from different sources over the internet. The dataset contains 8 types of fruits and vegetables. These are Apple, Asian pear, Cucumber, Mango, Orange, Pineapple, Pomegranate, and Strawberry. Each type of fruit or vegetable has 30 images. Hence, the dataset contains 240 images in total. All the images in the dataset contain a particular type of fruit or vegetable in the natural environment. Fig. 3.6 shows the sample images from each category in the dataset.





Fig. 3.6. Sample images from the dataset

3.4 Experimentations & Results

Some experiments are performed on the dataset for testing the effectiveness of the proposed technique. Fig. 3.7 shows the segmentation of two samples using GrabCut and then ROI detection. The input image may contain multiple fruit or vegetable. It is assumed that a single fruit or vegetable will be harvested at a time. Hence, the harvesting hand will be adjusted accordingly. The initial hard labeling gives the first indication of ROI. The region inside the marked rectangle contains the fruit or vegetable to be harvested. It may also contain another fruit or vegetable located at the back of the main fruit or vegetable to be harvested. The other regions except ROI are discarded by region area thresholding. Fig. 3.8 shows the final segmentation result using the proposed technique for each category of fruits and vegetables. The images in Fig. 3.8 are the output of the samples shown in Fig. 3.6. The performance can be seen by matching the samples between Fig. 3.6 and Fig. 3.8 sequentially. It is observed that

the average time taken for segmentation and ROI detection is below 1 second, which is acceptable for automated harvesting. It can be concluded that the segmentation and ROI detection performance is satisfactory.

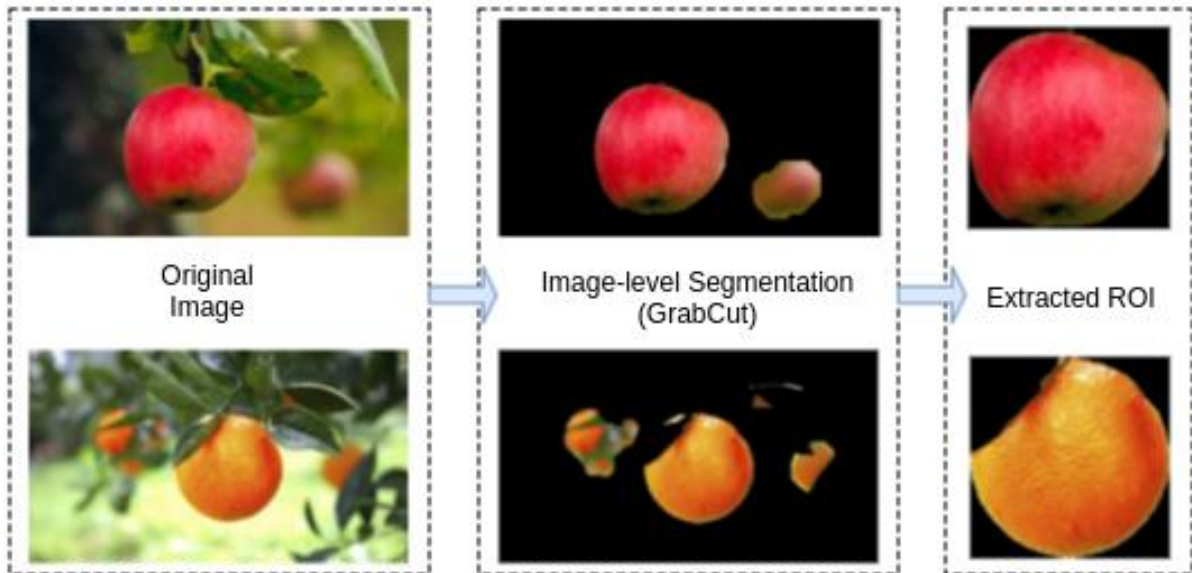


Fig. 3.7. The flow of segmentation and ROI detection



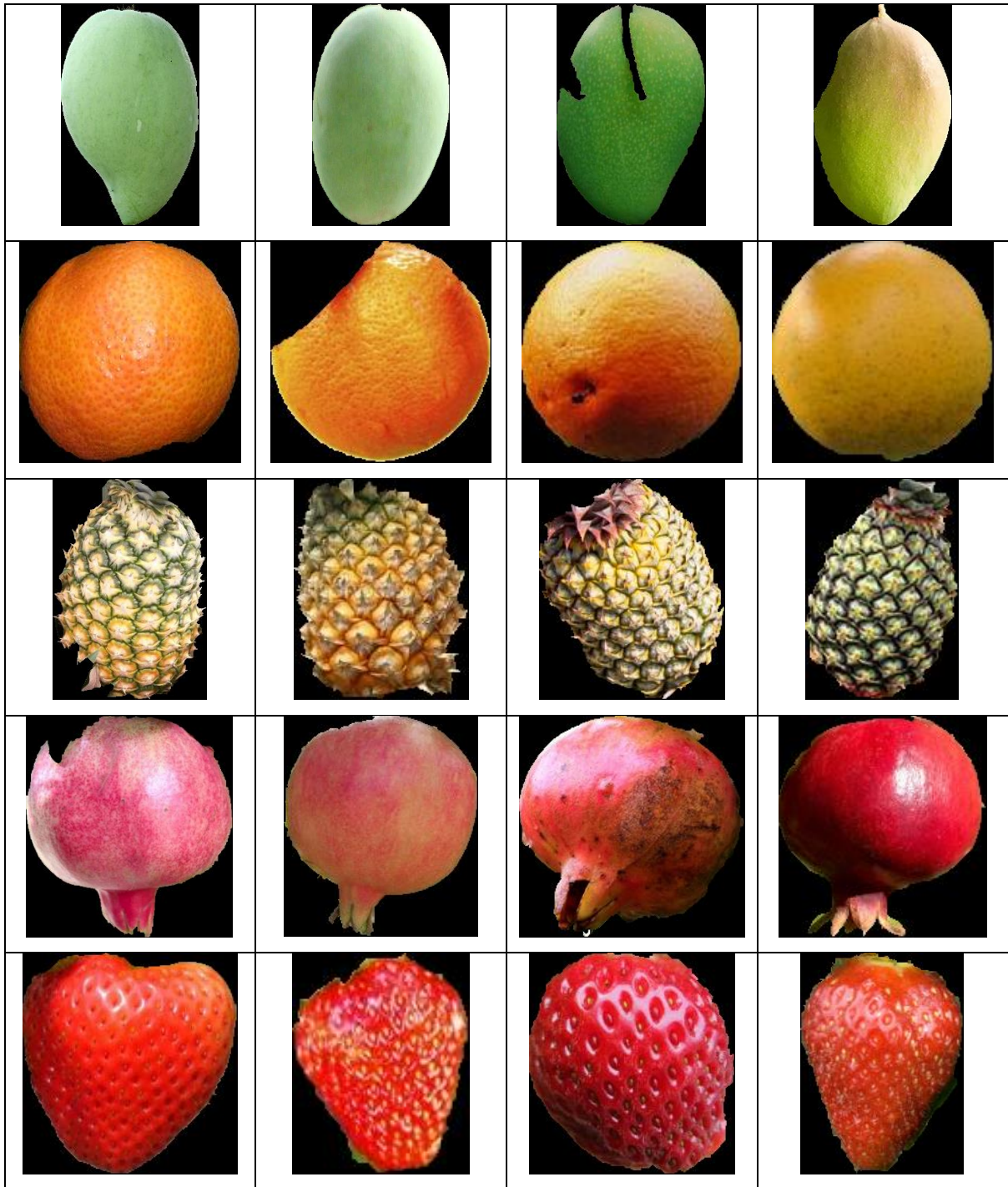


Fig. 3.8. The final output after segmentation and ROI detection using the proposed approach

The segmentation performance of the proposed approach is compared with the very popular Otsu [Otsu, 1979] segmentation algorithm in this context. Fig. 3.9 shows this comparison. It shows that where the background color is nearly similar to the foreground color, the proposed technique performs very well compared with Otsu thresholding. The reason is that Otsu thresholding performs on the image globally, where has proposed technique works locally after the initial hard labeling. The initial hard labeling adds the prior information about the

background pixels. This hard labeling helps to redistribute and label the pixels inside the initial ROI. The Otsu thresholding does not work with such prior information. Hence, the Otsu thresholding is not appropriate in this context.

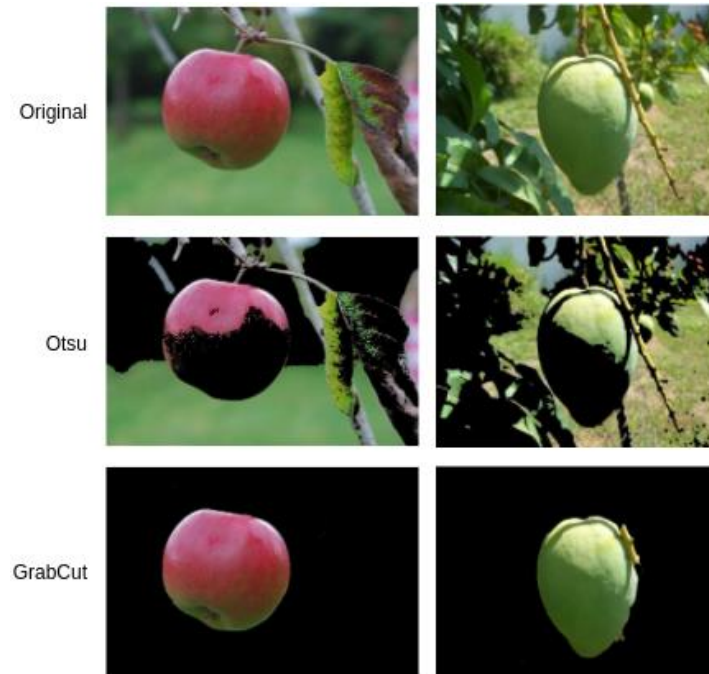


Fig. 3.9. Comparison of segmentation performance between the proposed approach and Otsu thresholding

3.5 Summary

Automated harvesting is very much needed in the agricultural field to deal with the problem of a labor shortage as well as to compete with the global market. The first step of automated harvesting is to segment the fruit and vegetable from the natural background and detect the ROI for adjusting the harvesting hand. Most of the prior works address this problem with color-based segmentation. The color-based segmentation performs well on a specific type of fruit or vegetable. The color-based segmentation suffers where the background is a natural environment and the background color is nearly similar to the fruit or vegetable color. GrabCut technique is used here to segment the object from the background. At first, hard labeling is done by a rectangular box. The region outside the box is the sure background and inside the box is unknown. A graph is formed with the pixels in the image. A mincut algorithm is applied to the graph to segment the foreground and the background. A region area thresholding technique is proposed here to detect the ROI from the segmented image. The ROI is cropped after the thresholding is completed successfully. The experimentation result shows that the performance is satisfactory for automatic harvesting.

The major contribution of this chapter-

- Preparing the dataset by collecting the images of fruits and vegetables with natural background
- Adopting the GrabCut technique for segmentation of fruits and vegetables from natural background
- An intelligent ROI marking technique after segmentation

The limitations of the proposed techniques-

- This approach works best with single fruit or vegetable with occlusion by branches and leaves.
- The approach is not suitable for overlapped fruits and vegetables.

Sorting of Fresh and Non-fresh

4.1 Introduction

Maslow's hierarchy of needs [Maslow, 1943] states that food is one of the physiological needs of human beings. Fruits and vegetables are consumed as a food item on daily basis either by cooking or directly. Modern people are very much concerned about the safety of food products because it directly affects our health. There is always a high demand for fresh and good quality fruits and vegetables in the open markets as well as in supermarkets. The fresh fruits and vegetables are delicious to eat as well as a very good source of important vitamins, minerals, etc. It also prevents us from various seasonal diseases. The food processing industry prepares various delicious food items from fruits and vegetables. The quality of food may degrade if it has been made from non-fresh fruits and vegetables. The rotten and defective comes under the non-fresh category of fruits and vegetables. The fruits and vegetables become rotten when it has been infected by harmful germs, bacteria, fungus, etc [Singh et al., 2018]. The defective or damaged items were found in the lot due to transportation [Cao et al., 2019] or by human errors. A single rotten fruit or vegetable may cause rot to multiple fruits and vegetables in the inventory. It spreads very quickly in the inventory because fruits and vegetables are highly perishable. The perishable product needs special care to store or process. Hence, non-fresh fruits and vegetables need to be detected and removed as early as possible. Most of the prior researchers used surface features and conventional machine learning algorithms for classifying

fresh and non-fresh for a particular species of fruit or vegetable. The motivation of this research is to propose a state-of-the-art technique that can classify the fresh and non-fresh (rotten or defective) fruits and vegetables irrespective of the species.

4.2 Sorting Parameters

The prime attributes of fruit and vegetable quality are color, texture, flavor, and nutritional value [Barrett et al., 2010]. At first the color from visual appearance is checked for acceptance or rejection of fruit and vegetables. The color says lots of things about the fruits and vegetables' maturity and ripeness. The glossiness, the ability to reflect light from the fruit and vegetable surface, is an important indication of surface moisture in the freshly harvested product. The appearance of defect and rot in the surface is a major indication of non-fresh fruits and vegetables.

The texture can be physically perceived through touch by hand or at the time of chewing inside the mouth. The surface texture can also be identified from the visual appearance. There are different textures for different types of fresh fruits and vegetables. The defective or rotten fruits and vegetables are rejected from the change in the normal texture of a particular fruit or vegetable.

The flavor comes next to color and visual appearances for picking fruit and vegetable as fresh to eat. The flavor can be defined by aroma and taste. The aroma can be perceived through the nose. The taste can only be perceived by tongue if it has been chewed in the mouth. Though the visual appearance and color have more impact than the aroma and taste to sort fruit in fresh and non-fresh. But, aroma and taste also make the desire of buying and consuming the fruit or vegetable again and again. The flavor is very difficult to classify. The primary tastes are sweet, salty, bitter, sour, etc. The aroma [Gould, 1977] can be flowery, fruity, burnt, etc. Some of the fruits and vegetables have a strong flavor at maturity and ripening. But the flavor degrades very quickly after harvesting mature and ripe fruits and vegetables. It makes the detection of fresh and non-fresh fruits and vegetables an easy task. Most of the fruits become sweeter at ripening. It can be said that a fruit of those categories is not fresh if it does not taste sweet.

Nutritional value is a hidden and most valued attribute of fruits and vegetables. The effects are seen on human health after a long time. Fresh fruits and vegetables contain different “macro” and “micro” nutrients. The nutrients include carbohydrates, fiber, vitamins, minerals, etc. There is a clear expectation of consumers that fresh fruits and vegetables contain a sufficient amount

of nutrients. This is very unfortunate that we can not get that information from the visual appearance. The variation of nutrients depends on the cultivation process and post-harvesting treatments.

The sorting of fresh and rotten or defective can be possible by those attributes. The visual appearance of the fruits and vegetables determines that they will be accepted or rejected. Visual appearance is not an extreme quality attribute [Kramer, 1965]. The aroma and taste come after the visual appearance. Those can be checked by smelling and eating. The nutrients value is taken a long time to affect the human body. Hence, computer vision-based sorting has to rely only on the visual appearance of fruits and vegetable surfaces. The change in color and texture can be used to classify fresh and non-fresh (rotten or defective) fruits and vegetables. The challenges of sorting the fresh and non-fresh are (a) the color in maturity and ripe is different for different types of fruits and vegetables (b) the pattern of rot or defect is different for different species of fruits and vegetables.

4.3 Proposed Method

This chapter proposes a convolutional neural network (CNN) architecture to address the challenges in the detection of rotten or defective fruits and vegetables. The proposed CNN model will be trained with the images of both fresh and non-fresh fruits and vegetables. The image of unknown fruit or vegetable sample will be sent to the trained CNN model for prediction. The predicted class label will be either fresh or non-fresh.

4.3.1 Convolutional Neural Network

The convolutional neural network is a state-of-the-art technique for image classification, object recognition. CNN contains multiple layers for different purposes of a classification problem. The benefit of using CNN is that it extracts features by preserving spatial information among pixels. A filter convolves through the image and extracts various image features i.e. edge, curve, color, etc. The selection of the filter size is very crucial with respect to the nature of the feature. The size of the filter should be large enough so that it can accommodate the feature containing many pixels as well as small enough for use in a repetitive manner. Fig. 4.1 shows a demonstration of the convolution process on a 6×6 binary image. Here, the convolution filter is a 3×3 matrix. The convolution starts from the top-left corner of the image without any padding and stride as 1 in both horizontal and vertical directions. The filter is multiplied with the corresponding pixels of the current position in each move. The sum of the multiplied pixel

is used here to generate a value in the feature map. The complete feature map is constructed with the values from all the moves during convolution.

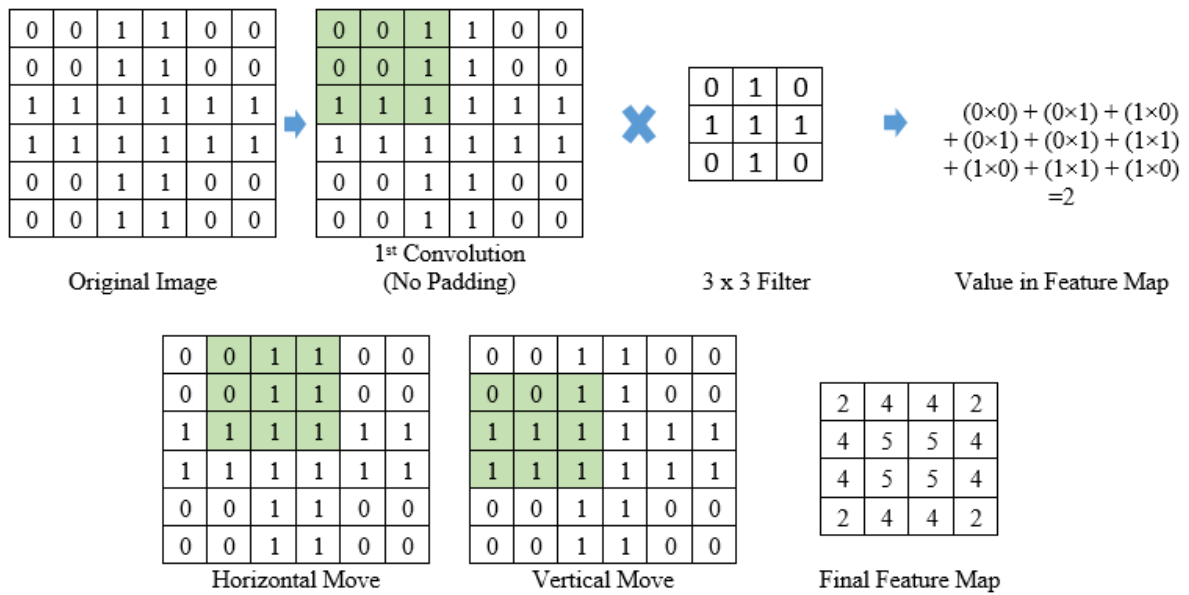


Fig. 4.1. A demonstration of the convolution process

The stride in convolution means the number of pixels to be escaped for a single move in a particular direction. The stride as [1 1] means the pixel will move just 1 pixel at the time in horizontal progress as well as vertical progress. The larger stride reduces the size of the feature map but increases the chances of missing small features. The padding is the process of adding dummy pixels to adjust the size of the feature map. The padding is generally done by adding pixels with a value of '0'. The filter generally moves through the image from the left side to the right side in the horizontal direction and top to bottom in the vertical direction. The filter will move separately in different channels for the image containing multiple channels. The mechanism of the filter movement is the same for all the channels. The purpose of applying multiple filters is to extract different types of features. The combination of multiple types of features improves the classification performance.

The Rectified Linear Unit (ReLU) is a very important activation function in the CNN model. The purpose of ReLU is to add non-linearity to the network. The larger dimension of the feature map becomes a headache to the network in terms of processing costs. Pooling is a popular technique to reduce the size of the feature map. The pooling has different varieties in terms of the process of calculating pooled values. The varieties of pooling are max pooling, sum pooling, average pooling, etc. The max pooling picks the maximum value from the pooling window. The sum pooling does the sum of the values in the pooling window and takes this

value of the sum. The average pooling computes the average of the values in the pooling window and takes this average value. The max pooling is very popular in image classification problems. Fig. 4.2 shows an example of max pooling. The different color represents different pooling windows. Here, the size of the pooling window is 2×2 . The red-colored value is the maximum value of the corresponding pooling window. The maximum value from each pooling window is selected for the reduced feature map.

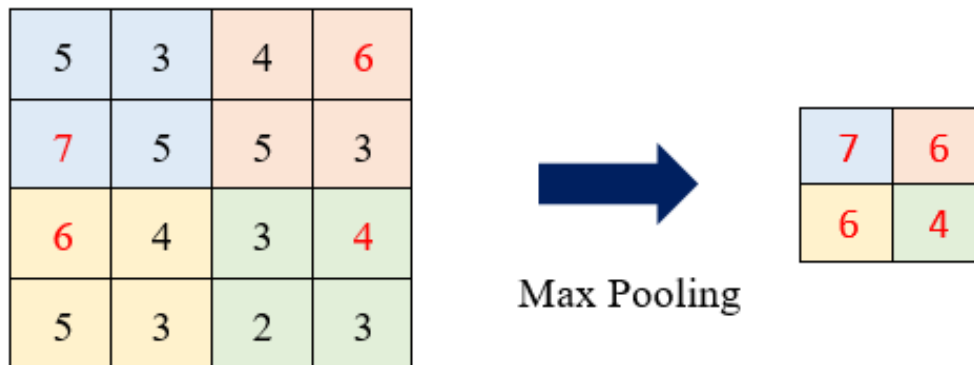


Fig. 4.2. An example of max pooling

4.3.2 Proposed Convolutional Neural Network Architecture

The architecture of the proposed CNN is very simple. Fig. 4.3 shows the architecture of the proposed CNN model. The proposed model contains 19 layers in total. The first layer is the input layer. The input layer receives a 64×64 RGB color image with zero center normalization. The next layer after the input layer is a convolution layer. This convolution layer is configured with 8 number of 3×3 filters, stride as $[1 \ 1]$, and zero paddings. The padding by zero has been adjusted in such a way so that the dimension of the feature map will have the same size as the input size. The convolution layer will extract more specific features with the progression of training. Then a batch normalization layer with 8 channels is added in the sequence of the network. This layer normalizes the features which have been extracted by the filters in the prior convolution layer. This layer makes the training faster as well as gives the network flexibility of independent learning. The batch normalization layer is followed by a ReLU layer and a max pooling layer. The ReLU is used here to add nonlinearity with the activation function as shown in Eq. (4.1). The ReLU keeps the positive input as it is and converts the negative input to zero. The max pooling layer is configured with a 2×2 pooling window, stride as $[2 \ 2]$, and padding $[0 \ 0 \ 0 \ 0]$. The first block of the net is formed with a convolution layer, a batch normalization

layer, a ReLU layer, and a max pooling. The sequence of the layers is very important here. The layers should be placed in sequence as shown in Fig. 4.3.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

4.1

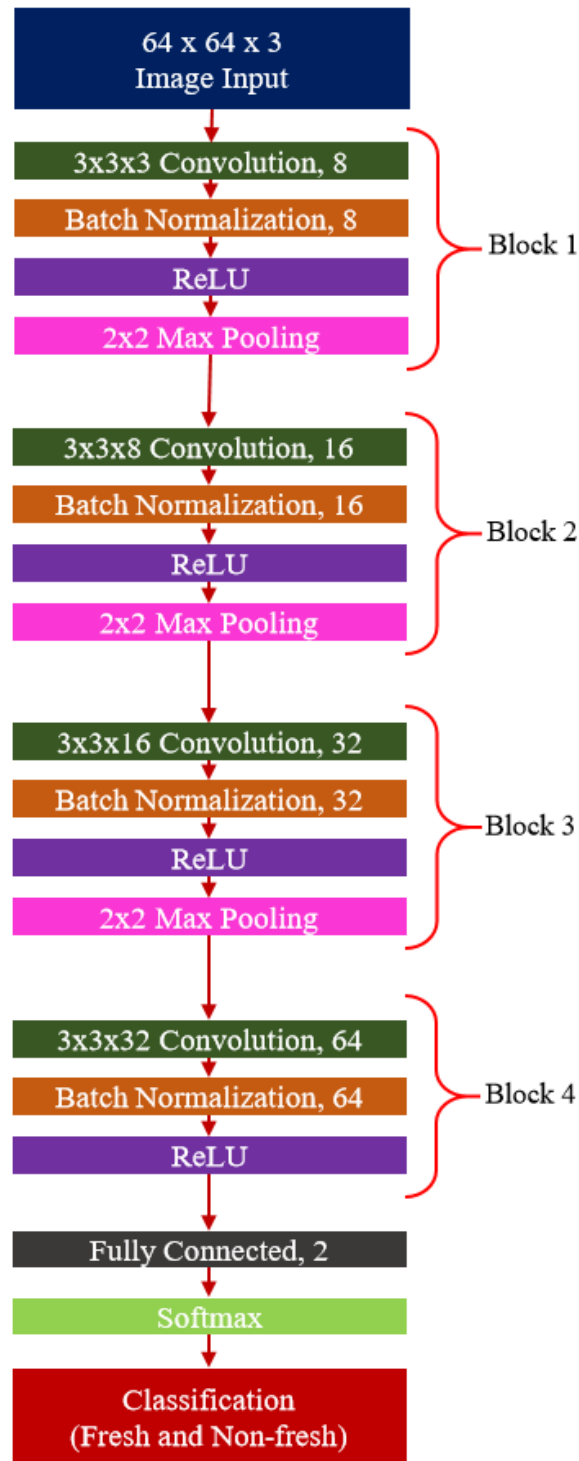


Fig. 4.3. The architecture of the proposed Convolutional Neural Network model

Three similar blocks have been added one after another. The number of convolution filters will be doubled in the consecutive block as the sequence progresses. The number of channels in the batch normalization layer will be adjusted similarly as the number of the filter in the same block. A special adjustment will be made on the 4th block of the sequence by removing the max pooling layer. The final block will be followed by three more layers. The sequence of layers is fully connected layer, softmax layer, and classification layer. This fully connected layer contains two nodes for two classes i.e. fresh and non-fresh. The softmax layer normalizes the output of the fully connected layer as well as generates prediction probabilities. Refer to Eq. (4.2). The classification layer predicts between the class fresh and non-fresh. The cross-entropy loss function, Eq. (4.3), is used in the classification layer. Here $i = 2$, because i stands for the number of classes. More specifically, the binary cross-entropy loss function is used for this binary classification problem. Refer to Eq. (4.4). Here, t_1 will be “1” when the class is positive and t_1 will be “0” for the negative class.

$$f(S)_i = \frac{e^{s_i}}{\sum_{k=1}^K e^{z_k}} \quad 4.2$$

$$CE = - \sum_{i=1}^{c=2} t_i \log(f(s_i)) = - t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1)) \quad 4.3$$

$$CE = \begin{cases} -\log(f(s_1)) & \text{if } t_1 = 1 \\ -\log(1 - f(s_1)) & \text{if } t_1 = 0 \end{cases} \quad 4.4$$

4.3.3 AlexNet Architecture

AlexNet [Krizhevsky et al., 2012] is a very popular convolutional neural network architecture. The architecture of AlexNet has been specially designed for object classification from high-resolution images. The AlexNet has been trained with 1000 classes of objects from the ImageNet dataset. This model secured the second position in the ILSVRC-2012 contest. The model receives an RGB image of size 227×227 as input. The model contains 25 layers in total. The model has been structured with 5 convolution layers, 2 cross channel normalization layers, 7 ReLU layers, 3 max pooling layers, 3 fully connected layers, and 2 dropout layers. The dropout reduces the overfitting of the fully connected layer. The final fully connected layer contains 1000 nodes and is followed by softmax layer and classification layer.

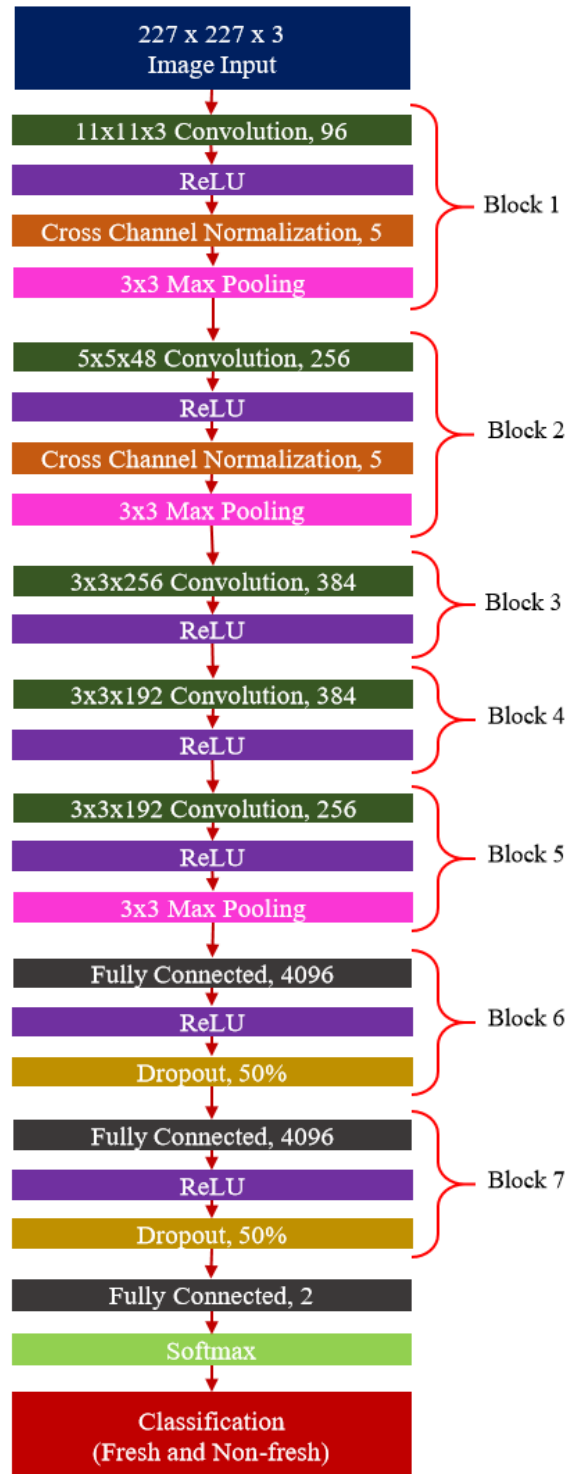


Fig. 4.4. The architecture of the fine-tuned AlexNet

Transfer learning is the way of reusing the knowledge of solving a similar type of prior problem at the time of solving a new problem. The transfer learning can be implemented by customizing a pre-trained network model for a particular classification problem. The AlexNet model has been trained with millions of images for a wider range of classes. The model has already learned a rich feature representation. Hence, it has been used for lots of classification problems

by fine-tuning with the help of the transfer learning technique. Transfer learning is also applied in the context of this problem. The final three layers have been customized. The final fully connected layer is replaced with a fully connected layer of two nodes as it is a binary classification problem between fresh and non-fresh. The softmax and classification layers at the end are also replaced. The new classification layer uses a binary cross-entropy loss function. The entire architecture of fine-tuned AlexNet is shown in Fig. 4.4.

4.4 Dataset

The dataset should have an ample amount of variety and large enough to train a deep learning model. The main dataset contains three types of fruits i.e. Apple, Banana, and Orange. The images are downloaded from a freely available online source [Kalluri, 2020]. The dataset contains 13599 images in total. Each of the fruit categories contains images of fresh samples as well as rotten or defective samples. The dataset has six categories of fruits i.e. fresh apple (232), rotten apple (327), fresh banana (218), rotten banana (306), fresh orange (206), and rotten orange (222). All the categories have an ample amount of varieties. Fig. 4.5 shows the samples from all six categories.

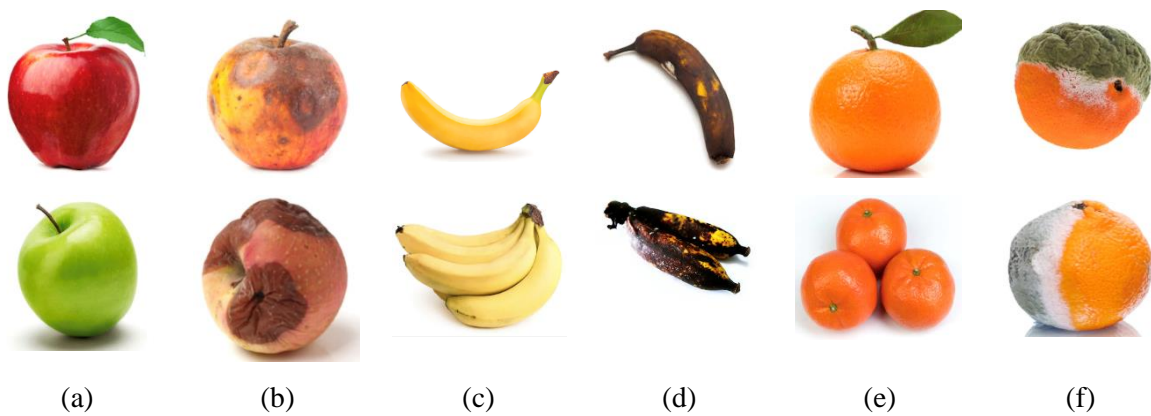


Fig. 4.5. Samples from all the categories (a) Fresh Apple (b) Rotten Apple (c) Fresh Banana (d) Rotten Banana (e) Fresh Orange (f) Rotten Orange

A large number of images are needed to train a deep learning model. Hence, eight different image augmentation techniques were applied to each of the samples to increase the number of images in the dataset. The augmentation techniques are rotation in five directions i.e. 15° , 30° , 45° , 60° , 75° , salt and pepper noise, translation, and vertical flip. Fig. 4.6 shows the actual sample and the images after applying different augmentation techniques.

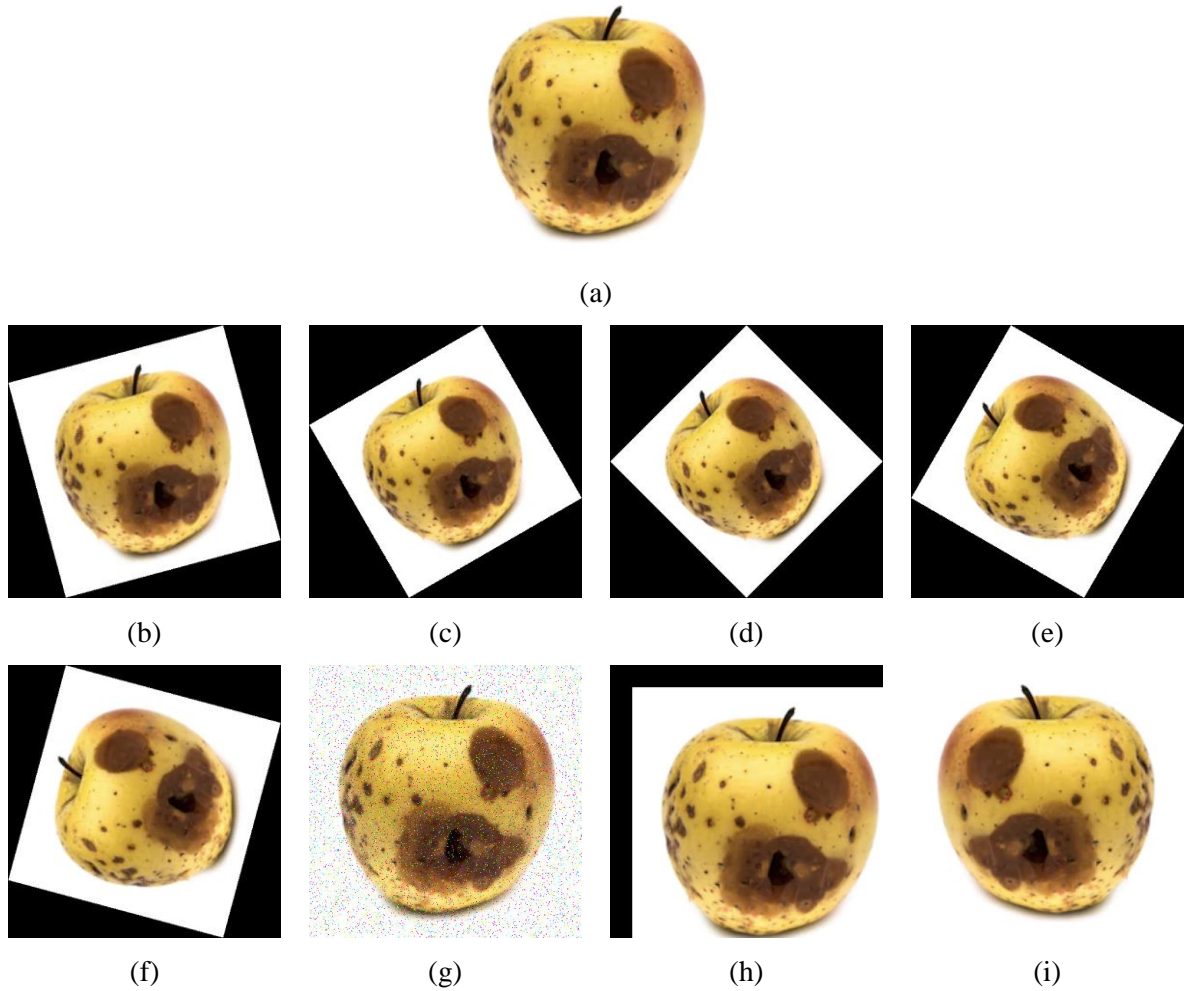


Fig. 4.6. (a) A sample image and its appearance after applying different augmentation techniques (b) 15° rotation, (c) 30° rotation, (d) 45° rotation, (e) 60° rotation, (f) 75° rotation, (g) Salt and pepper noise, (h) Translation, and (i) Vertical flip

Four datasets (i.e. Datasets 1, 2, 3, and 4) are created from those images for performing several experiments. Dataset 1 includes images of both fresh and non-fresh apples. Dataset 2 contains images of fresh bananas as well as non-fresh bananas. Dataset 3 contains images of fresh oranges and non-fresh oranges. Dataset 4 has two categories i.e. fresh and non-fresh. The fresh category contains fresh samples of all three types of fruits i.e. apple, banana, and orange. The non-fresh category contains non-fresh samples of those three types of fruits.

Another dataset of apple fruit has been created to test the performance of the proposed model i.e. Dataset 5. The source of images is different than dataset 1. Dataset 5 contains images of fresh apples (15) as well as rotten apples (244). In total, it contains 259 images. The images of fresh apples are taken from different online sources. The images of non-fresh apples are taken from a GitHub source [Siddiqi, 2020]. Fig. 4.7 shows some samples from dataset 5.



Fig. 4.7. Samples from dataset 5 (a), (b) Fresh apple (c), (d) Non-fresh apple

Dataset 6 is a mixture of different types of fruits and vegetables. The types of fruits and vegetables in dataset 6 exclude apple, banana, and orange. That means dataset 6 does not contain any type of fruits and vegetables that are in the previous five datasets i.e. 1, 2, 3, 4, and 5. Most of the images of dataset 6 are captured by us. A few images have been downloaded to incorporate more variety and balance in this dataset. All the images in dataset 6 are separated into two classes i.e fresh (68) and non-fresh (40). Fig. 4.8 shows a few fresh as well as non-fresh samples from dataset 6. The fresh class includes images of six different types of fruits and vegetables i.e. capsicum (red, green, yellow), cucumber, jackfruit, lime, strawberry, and tomato. The non-fresh class includes images of nine different types of fruits and vegetables i.e. carrot, jackfruit, lemon, mango, papaya, pear, strawberry, tomato, and zucchini.



Fig. 4.8. Samples from dataset 6 (a)(b) Fresh fruits and vegetables (c)(d) Non-fresh fruits and vegetables

Table 4.1. Distribution of training and testing images of different datasets

Dataset	Class	Total	Training	Testing
Dataset 1	Fresh Apple	2088	1600	488
	Non-fresh Apple	2943	1600	1343
Dataset 2	Fresh Banana	1962	1600	362
	Non-fresh Banana	2754	1600	1154
Dataset 3	Fresh Orange	1854	1600	254
	Non-fresh Orange	1998	1600	398
Dataset 4	Fresh Fruits and Vegetables	5904	4800	1104
	Non-fresh Fruits and Vegetables	7695	4800	2895
Dataset 5	Fresh Apple	15	-	15
	Non-fresh Apple	244	-	244
Dataset 6	Fresh Fruits & Vegetables	68	-	68
	Non-fresh Fruits and Vegetables	40	-	40

4.5 Experimentations and Results

The implementation and experimentations are done on Matlab 2018a. The input of the proposed model is an RGB 64×64 image. The reason behind the smaller size input is that the reduction of training time. All the images are resized to 64×64 to make the images compatible with the proposed CNN model. The training parameters are very important to build a robust deep learning model. The initial learning rate is given as 0.01. The maximum number of the epoch is 25. The proposed CNN model is trained separately with four datasets i.e. 1, 2, 3, and 4. The training and testing images are divided randomly for every dataset. The distribution of training and testing images is mentioned in Table 4.1. The images are shuffled in every epoch during the training.

In real life, the number of the non-fresh sample should be less compared with the number of fresh samples in a lot of fruits and vegetables. The detection of a non-fresh sample is more important than a fresh sample as we know that the non-fresh sample is very harmful to a lot of fruits and vegetables. Hence, the non-fresh class is considered a positive class and the fresh class is negative. The prediction performance on test images of those four datasets can be represented by various metrics. The sample, which comes from the positive class and is also predicted as the positive class, is marked as True Positive (TP). The sample, which comes from the positive class and is predicted as the negative class, is marked as False Negative (FN). The sample, which comes from the negative class and is also predicted as the negative class, is marked as True Negative (TN). The sample, which comes from the negative class and is predicted as the positive class, is marked as False Positive (FP). The classification performance is generally represented by accuracy. Refer to Eq. (4.5). Only the classification accuracy is not sufficient for this kind of class imbalance problem. Hence, some more specific performance measure is also considered in this experimentation. These are recall, precision, and f1 score. The recall, precision, f1 score can be computed using Eqs. (4.6), (4.7), and (4.8) respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad 4.5$$

$$Recall = \frac{TP}{TP + FN} \quad 4.6$$

$$Precision = \frac{TP}{TP + FP} \quad 4.7$$

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad 4.8$$

In total, eight performance metric is considered for this work. These are fresh and predicted as fresh, fresh but predicted as non-fresh, non-fresh but predicted as fresh, non-fresh and predicted as non-fresh, accuracy, recall, precision, and f1 score. The performance of the proposed model is measured on four datasets i.e. 1, 2, 3, and 4. The training and testing have been done four times on each dataset. The dataset has been shuffled on each run. The aggregate result of 4 runs is presented here for the proposed CNN in Table 4.2.

Table 4.2. The prediction performance using the proposed CNN model on different datasets

Performance Metrics (%)	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fresh and Predicted as Fresh	98.51	99.93	98.52	97.60
Fresh but Predicted as Non-fresh	1.49	0.07	1.48	2.40
Non-fresh but Predicted as Fresh	1.38	0.09	1.07	2.20
Non-fresh and Predicted as Non-fresh	98.62	99.91	98.93	97.80
Overall Accuracy	98.59	99.92	98.77	97.74
Recall	98.62	99.91	98.93	97.80
Precision	99.46	99.98	99.06	99.08
F1 Score	99.04	99.95	99.00	98.43

AlexNet, which is a very popular convolutional neural network model for object classification, is applied for this problem by transfer learning. The RGB images are resized to 227×227 for fitting into the input layer of AlexNet. The training and testing have been done with a similar distribution of images on four datasets (datasets 1, 2, 3, and 4) as done for the proposed CNN model. The experimentations and measurements of performance metrics have been done similarly as done for the proposed CNN model. Table 4.3 shows the performance of the fine-tuned AlexNet model in the context of this problem.

Table 4.3. The prediction performance using the fine-tuned AlexNet model on different datasets

Performance Metrics (%)	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fresh and Predicted as Fresh	97.95	98.55	100.00	99.43
Fresh but Predicted as Non-fresh	2.05	1.45	0.00	0.57
Non-fresh but Predicted as Fresh	0.34	0.00	0.31	3.11
Non-fresh and Predicted as Non-fresh	99.66	100.00	99.69	96.89
Overall Accuracy	99.21	99.65	99.81	97.59
Recall	99.66	100.00	99.69	96.89
Precision	99.26	99.55	100.00	99.78
F1 Score	99.46	99.77	99.84	98.31

Some more approaches are also implemented here for comparison on these datasets. The previous approaches are (i) histogram features by Capizzi et al., and Karakaya et al. et al. (ii)

GLCM features by Capizzi et al., Karakaya et al. and Chandini et al. (iii) bag-of-features by Karakaya et al. All those approaches used SVM classifier for prediction between fresh and non-fresh. Hence, those features are extracted and experimented with the SVM classification model separately on each dataset as done for the proposed CNN model and fine-tuned AlexNet model. Table 4.4 shows the performance using a grayscale histogram with an SVM classifier. The performance of GLCM based features with SVM classifier is shown in Table 4.5. Table 4.6 presents the performance of bag-of-features with the SVM classifier.

Table 4.4. The prediction performance using Grayscale Histogram with SVM on different datasets

Performance Metrics (%)	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fresh and Predicted as Fresh	48.31	95.93	53.94	53.89
Fresh but Predicted as Non-fresh	51.69	4.07	46.06	46.11
Non-fresh but Predicted as Fresh	41.25	2.99	37.69	48.59
Non-fresh and Predicted as Non-fresh	58.75	97.01	62.31	51.41
Overall Accuracy	55.97	96.75	59.05	52.09
Recall	58.75	97.01	62.31	51.41
Precision	75.79	98.70	68.20	74.62
F1 Score	66.19	97.85	65.12	60.88

Table 4.5. The prediction performance using GLCM Features with SVM on different datasets

Performance Metrics (%)	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fresh and Predicted as Fresh	76.38	89.23	84.15	79.91
Fresh but Predicted as Non-fresh	23.62	10.77	15.85	20.09
Non-fresh but Predicted as Fresh	43.17	13.26	36.18	28.99
Non-fresh and Predicted as Non-fresh	56.83	86.74	63.82	71.01
Overall Accuracy	62.04	87.34	71.74	73.47
Recall	56.83	86.74	63.82	71.01
Precision	87.83	96.28	86.39	90.35
F1 Score	69.01	91.26	73.41	79.52

Table 4.6. The prediction performance using Bag of Features with SVM on different datasets

Performance Metrics (%)	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Fresh and Predicted as Fresh	76.13	94.75	85.14	79.12
Fresh but Predicted as Non-fresh	23.88	5.25	14.86	20.88
Non-fresh but Predicted as Fresh	19.34	6.30	16.52	21.56
Non-fresh and Predicted as Non-fresh	80.66	93.70	83.48	78.44
Overall Accuracy	78.39	94.22	84.31	78.78
Recall	80.66	93.70	83.48	78.44
Precision	90.29	98.27	89.79	90.79
F1 Score	85.20	95.93	86.52	84.16

4.6 Analysis

Validation is very important for training a deep learning model. The images of the validation set are not a part of the training images. The Matlab provides a monitor to track the accuracy

and loss during the training. Fig. 4.9 and Fig. 4.10 show the snaps from the training monitor. Fig. 4.9 depicts that the training and validation accuracy is improving as long as the training progresses. Fig. 4.10 shows that the training and validation loss is decreasing with the progression of training. The network learns more complex features as long as the training progresses. The convolution layers at the beginning of the network learn low-level features. The convolution layers towards the end of this network learn high-level features. Fig. 4.11 shows the features extracted by the final convolution layer for both the proposed CNN and AlexNet. The AlexNet is already trained by 1000 classes. Hence, it has learned complex feature representations. The extracted features by the proposed CNN are less complex compared with the fine-tuned AlexNet but good enough to classify the fresh and non-fresh fruits and vegetables.

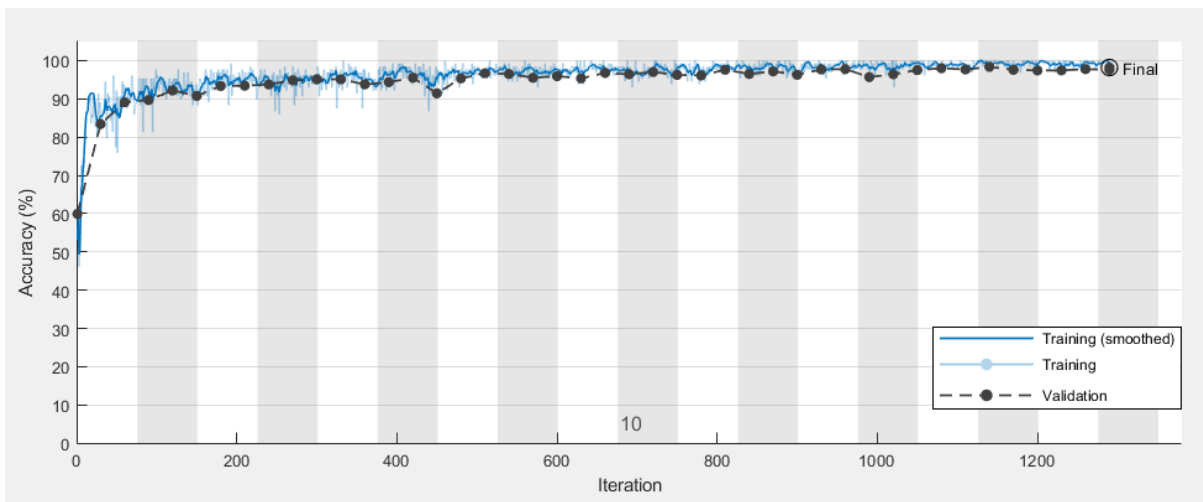


Fig. 4.9. Training vs validation accuracy for the proposed CNN model on dataset 4

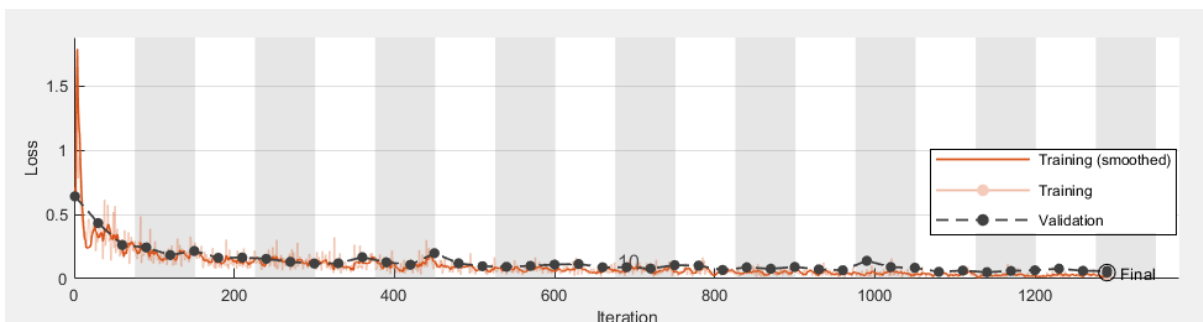


Fig. 4.10. Training vs validation loss for the proposed CNN model on dataset 4

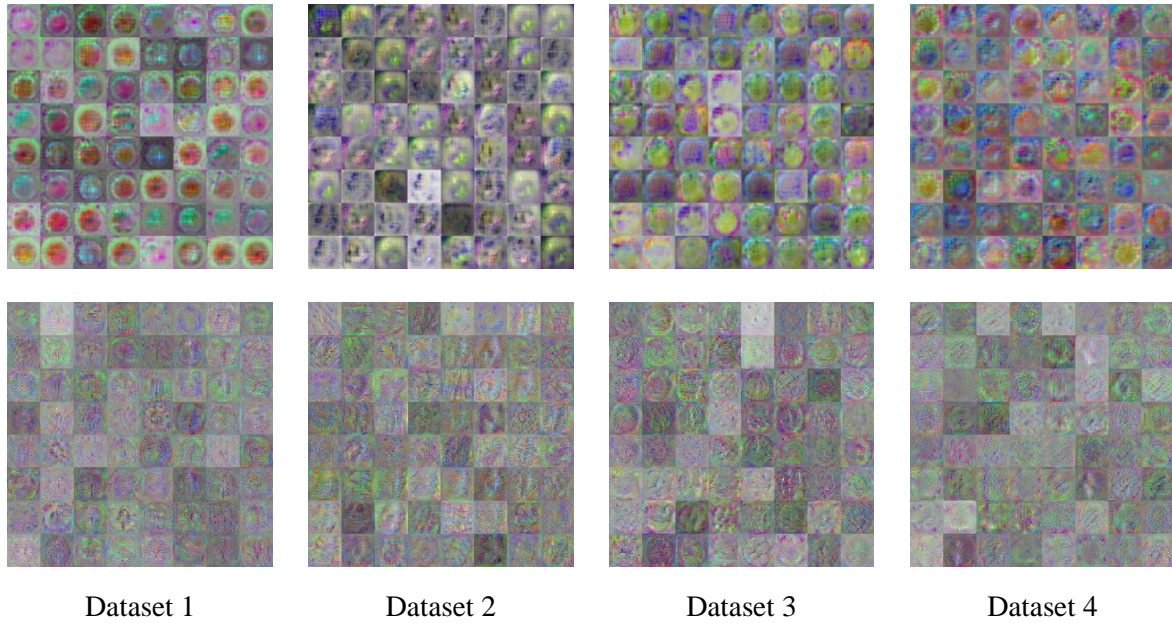


Fig. 4.11. The extracted features by final convolution layer in (row 1) Proposed CNN model (row 2) Fine-tuned AlexNet

The fine-tuned AlexNet and three prior machine learning approaches have been implemented and experimented in the context of this problem to compare the performance of the proposed approach. Fig. 4.12 shows the accuracy comparison of different prior approaches along with the proposed approach on four different datasets i.e. 1, 2, 3, and 4. Fig. 4.13, Fig. 4.14, and Fig. 4.15 show the comparison of recall, precision, and f1 score respectively for the same. The x -axis represents the four datasets and the y -axis represents the percentage value of corresponding performance metrics in those four plots. The different color represents different approaches as mentioned in the legend.

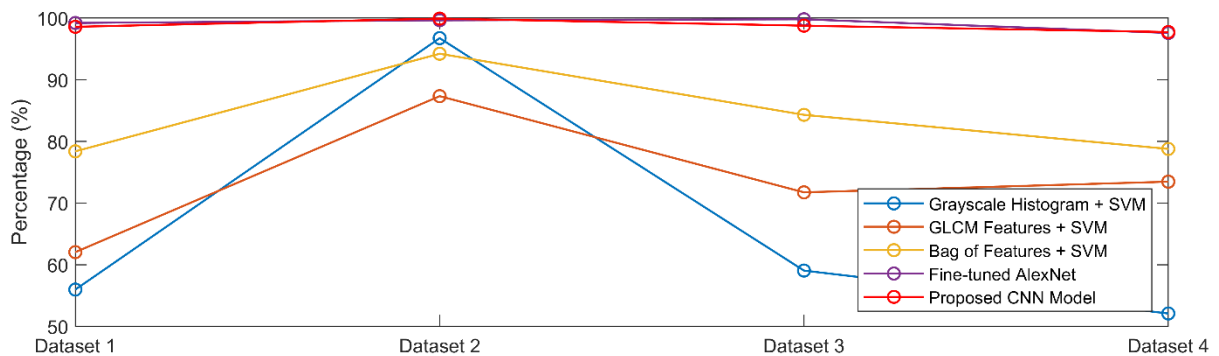


Fig. 4.12. Comparison of overall accuracy using different approaches on different datasets

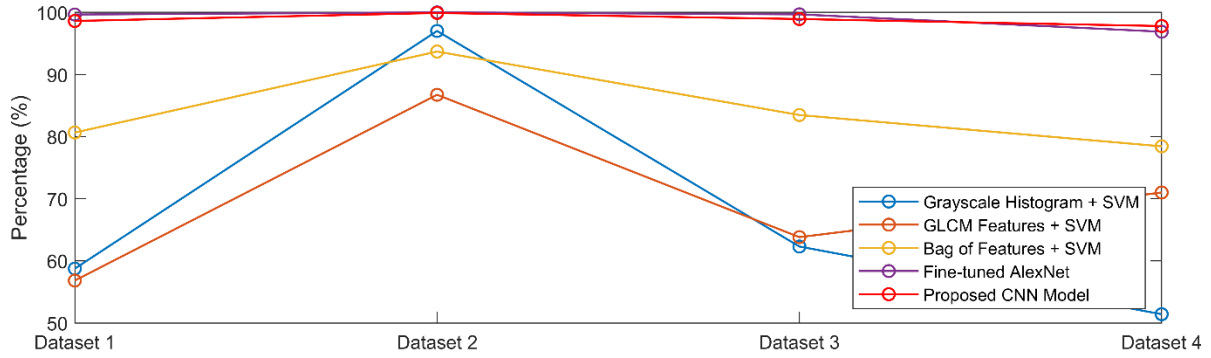


Fig. 4.13. Comparison of recall using different approaches on different datasets

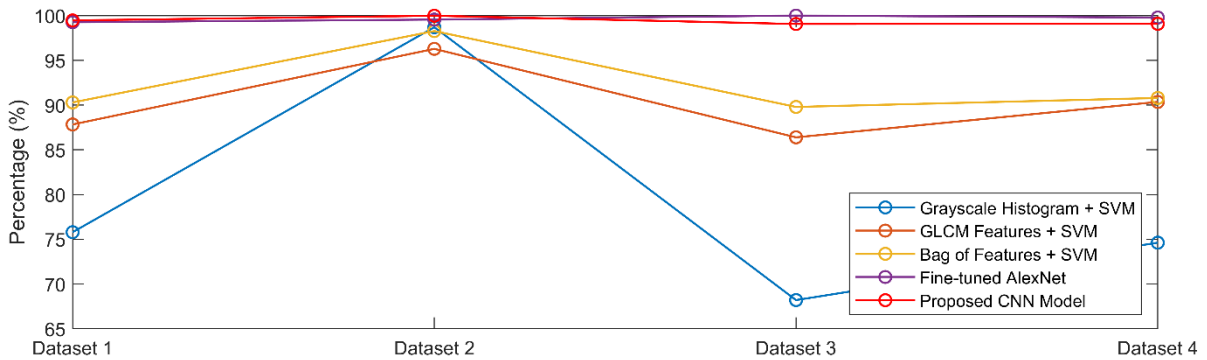


Fig. 4.14. Comparison of precision using different approaches on different datasets

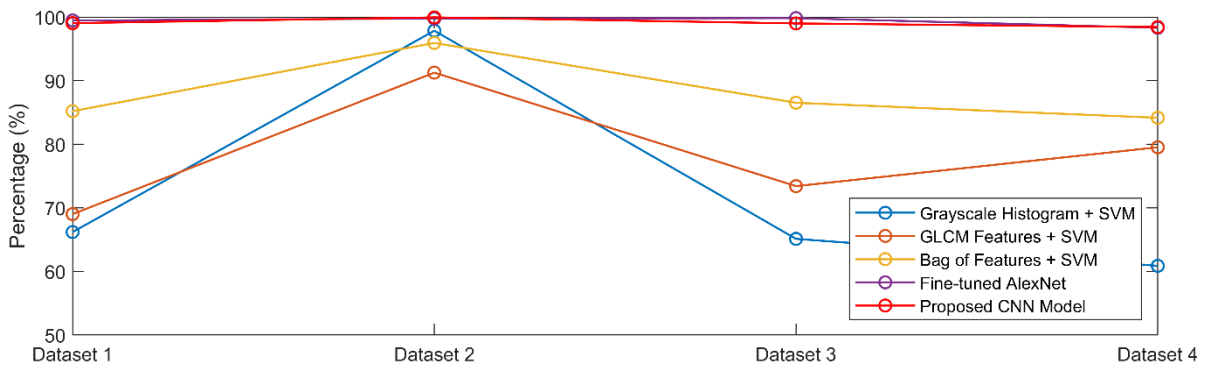


Fig. 4.15. Comparison of f1 score using different approaches on different datasets

The plot depicts that the proposed CNN performs nearly similar compared with the fine-tuned AlexNet model in the context of this problem. The difference of F1 score between those two network models is 0.42%, 0.18%, 0.84%, and 0.12% on dataset 1, 2, 3, and 4 respectively. The proposed CNN contains fewer layers than the AlexNet. The input size of the AlexNet is nearly four times the proposed CNN. Hence, the training cost of the proposed CNN is less than the fine-tuned AlexNet. The plots also show that the proposed CNN model outperforms in every aspect of performance than the prior machine learning approaches irrespective of datasets. The performance difference between the proposed approach and prior machine learning approaches

is 50% in many cases. The performance difference for the same on dataset 2, which contains fresh and rotten bananas, is comparatively less than other datasets. The bag-of-features with the SVM classifier perform well among the three prior approaches. The reason behind this performance difference may be the variety in our datasets.

Dataset 5 contains 259 images of only fresh and non-fresh apples. This dataset is only used here to test both the proposed CNN model and fine-tuned AlexNet, which was trained with the images of fresh and non-fresh apples from dataset 1. Table 4.7 shows the test results for both models on dataset 5. Fig. 4.16 shows the performance comparison between the test set of dataset 1 and dataset 5 when both the models are trained by images from dataset 1. The performance for the proposed CNN and fine-tuned AlexNet is the same on dataset 5. Hence, the redline is overlapped with the violet line.

Dataset 6 is a mixture of different types of fresh as well as non-fresh fruits and vegetables. The types of fruits and vegetables in this dataset exclude the types in dataset 4 i.e. apple, banana, and orange. This dataset is included here only to test the performance of the proposed CNN model and fine-tuned AlexNet model when it has been trained with different types of fresh and non-fresh fruits and vegetables but the test set does not include those types of fruits and vegetables. Table 4.7 shows the performance of the proposed CNN model and fine-tuned AlexNet on dataset 6 when the models are trained with the images from dataset 4. Fig. 4.17 shows the performance comparison between the test set of dataset 4 and dataset 6 when both the models are trained by images from dataset 4.

Table 4.7. The prediction performance on dataset 5 and dataset 6 using trained CNN & AlexNet

Performance Metrics (%)	Proposed CNN		Fine-tuned AlexNet	
	Dataset 5	Dataset 6	Dataset 5	Dataset 6
Fresh and Predicted as Fresh	93.33	95.59	93.33	95.59
Fresh but Predicted as Non-fresh	6.67	4.41	6.67	4.41
Non-fresh but Predicted as Fresh	1.23	2.50	1.23	10.00
Non-fresh and Predicted as Non-fresh	98.77	97.50	98.77	90.00
Overall Accuracy	98.46	96.30	98.46	93.52
Recall	98.77	97.50	98.77	90.00
Precision	99.59	92.86	99.59	92.31
F1 Score	99.18	95.12	99.18	91.14

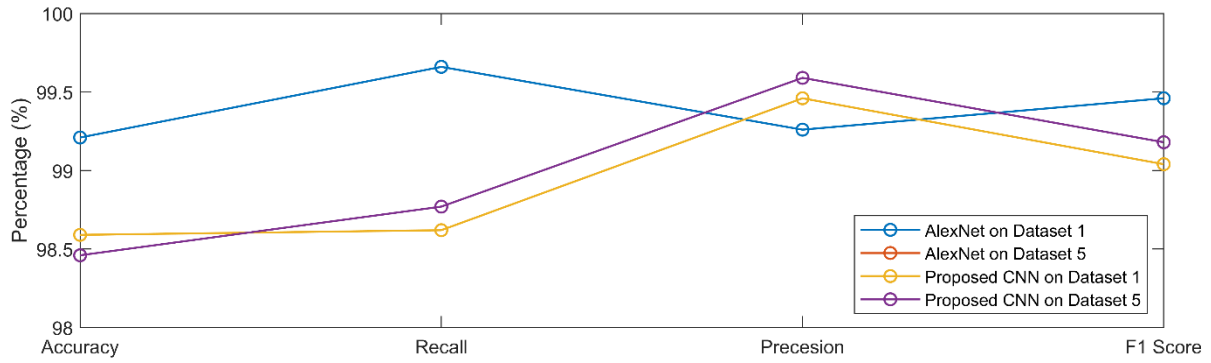


Fig. 4.16. Performance of CNN and fine-tuned AlexNet trained by Dataset 1

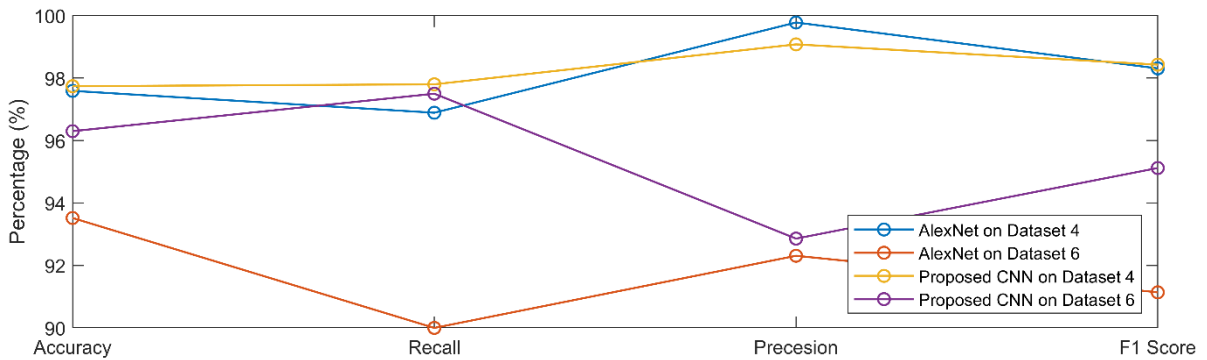


Fig. 4.17. Performance of CNN and fine-tuned AlexNet trained by Dataset 4

4.7 Summary

This chapter elucidates the need for the detection of non-fresh (rotten or defective) fruits and vegetables from a lot of fruits and vegetables. It also highlighted the challenges of solving the problem using computer vision and machine learning. Convolutional neural network architecture is proposed here to address the problem of fresh and non-fresh classification. A transfer learning on AlexNet has also experimented in the context of this problem. Some of the prior approaches are also implemented and tested here. The proposed CNN model outperforms the previous approaches over all the datasets.

There are different types of defects in different fruits and vegetables. The work does not specify the type of defect. Another limitation is that the sorting is done based on a single view. If there is a symptom of rot or defect from another view or another side, the system will not be able to detect accurately. Those limitations could be a very good scope of future research.

Classification

5.1 Introduction

There are lots of fruit and vegetable species in the world [Pennington et al., 2009]. Only edible fruits and vegetables are cultivated and brought for the customers to an open market or store located in a supermarket. The classification among different types of fruits and vegetables is needed for every aspect of processing in supermarkets and the food-producing industries. The fruits and vegetables should be separated according to the category before placing them in the store or processing them. There are some other aspects of classification as well. Fruits and vegetables classification in the mobile environment could be very effective for a visually impaired person to identify fruits and vegetable types. Manual classification is time-consuming and requires a large number of human resources. The barcode or catalog solves the problem partially. This is quite challenging to place the barcode in the huge number of fruits and vegetables in production. Sometimes searching from a large catalog of fruits and vegetables is not preferred by the customers. Hence, the classification should be done automatically and accurately in the conveyer belt using computer vision and machine learning.

This classification is not aimed at the taxonomy of the plant kingdom. This classification problem is targeted for business purposes and addressing the industry needs. The task of this classification system will be separating different types of edible fruits and vegetables. The

visual features should be available properly to classify fruit and vegetable type from an image. The visual features are shape, size, color, and texture. There are different challenges in the identification of visual features for automatic classification. The fruits and vegetables types have many inter-class similarities and intra-class dissimilarities. There are many fruits or vegetable types (*Genus*), which has different sub-types (*Species*) with different visual appearances e.g. Apple has many subtypes and sold for different prices. Hence, those sub-types also need to be separated in a different bucket for commercial purposes. Sub-types of a fruit or vegetable may have (a) different color but similar in shape and texture (b) different shape but similar in color and texture (c) different texture but similar in color and shape etc. Classification among different types of fruits and vegetables with their subtype is a very challenging task depending on a single type of feature.

It has been observed that the fruits and vegetables appear in a conveyer belt by placing the axis, which is connecting the stalk and calyx, along the horizontal plane. If the fruits and vegetables are seen from the top view, it may have appeared in any position as it is rotated vertically over the connector axis. The shape and color information is mostly unchanged from the top view due to this reason. If the fruits and vegetables are seen from the side view, it may have appeared in any position as it is rotated horizontally along with the connector axis. The shape and color of fruits or vegetable changes in different viewing positions, specially when it is seen from a side in a conveyer belt. The texture is the most reliable feature in this situation. Size also changes in the different growth stages of fruits and vegetables as well as with the change of the distance between the camera and object. The other challenges are noises, illumination variation, geometrical transformations, etc. Hence, considering all those difficulties we have designed a novel framework to classify fruits and vegetables.

5.2 Proposed Framework

This proposed framework aims to predict the class label for a fruit and vegetable sample. The proposed framework combines multiple visual features from two viewpoints to train a prediction or classification model. The trained classification model with an intelligent decision-making system predicts the label for a test sample of fruits or vegetables. The proposed framework has been split into two separate phases i.e. training phase and testing phase. Both the phrases have been integrated to build the complete framework as shown in Fig. 5.1.

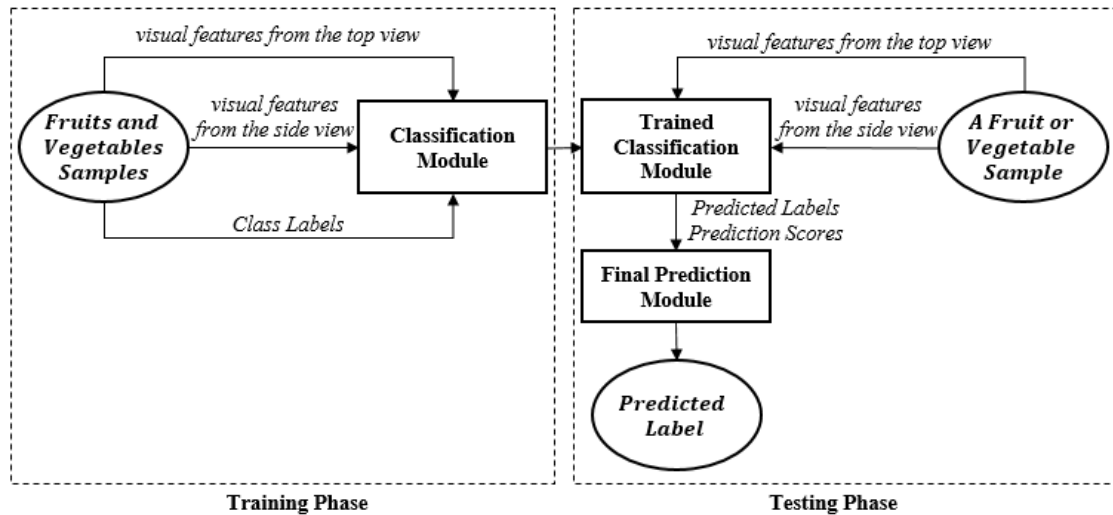


Fig. 5.1. A top-level view of the proposed framework for fruits and vegetable classification

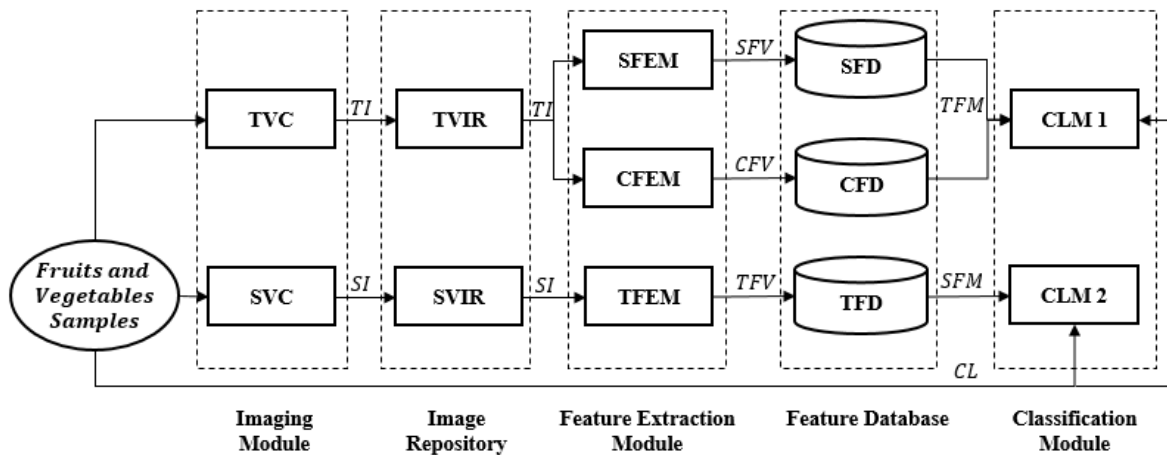


Fig. 5.2. The detailed architecture of the training phase

5.2.1 Components of Training Phase

The training phase consists of five different components. The components are divided based on the nature of the functionality. The components are Imaging Module (*IM*), Image Repository (*IR*), Feature Extraction Module (*FEM*), Feature Database (*FD*), and Classification Module (*CM*). Fig. 5.2 presents the detailed architecture of the training phase.

5.2.1.1 Imaging Module (*IM*)

This module is designed to capture the image of fruits or vegetable samples. The imaging setup will have two cameras in a perpendicular position and white background. The first camera (*TVC*) is fixed at the top of the object i.e. 90° and the second camera (*SVC*) is fixed at the left side of the object i.e. 0° . Both the cameras have fixed at 20 cm of distance from the marked position for keeping fruit or vegetable. The cameras are connected to a computer system to

capture the images from two different views. Fig. 5.3 shows the components and structure of the imaging setup.

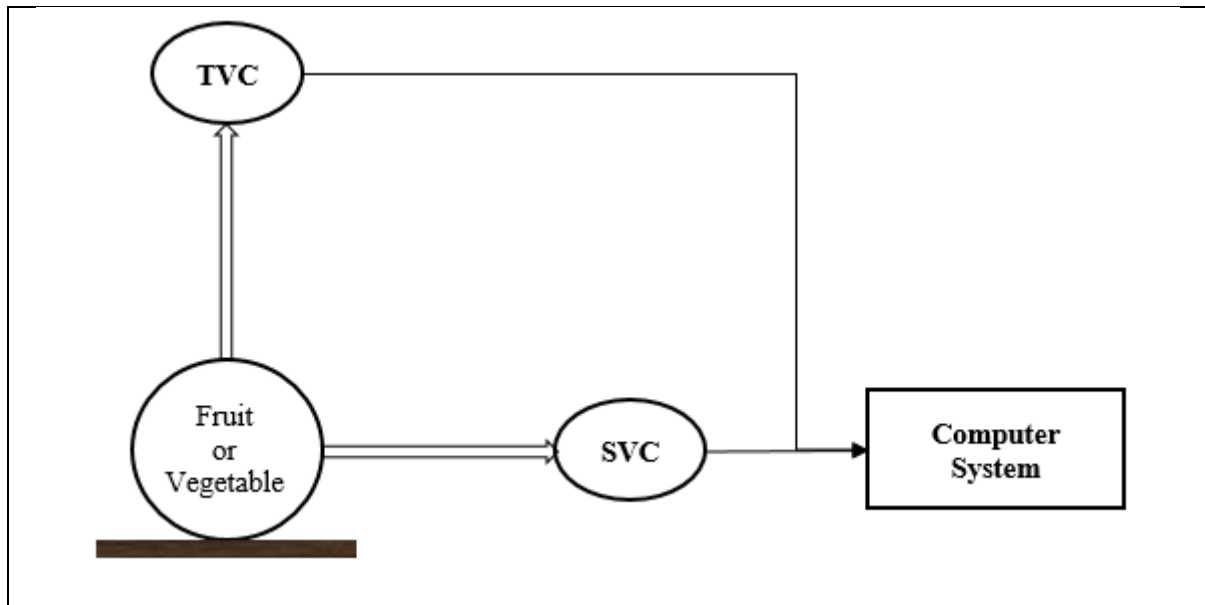


Fig. 5.3. Structure of the imaging setup

5.2.1.2 Image Repository (*IR*)

The training phase contains two separate image repositories. These are the top view image repository (*TVIR*) and side view image repository (*SVIR*). The *TVIR* stores the images captured through the *TVC*. The *SVIR* stores the images captured through the *SVC*. The *TVIR* and *SVIR* contain images of different fruits and vegetable classes under the corresponding subdirectory for a particular class. The RGB color images of fruits and vegetables are captured and stored in *jpg* format.

5.2.1.3 Feature Extraction Module (*FEM*)

This module is the most important module of this framework. The input of this module is two images i.e. top view image (*TI*) and side view image (*SI*). This module has three sub-modules i.e. shape feature extraction module (*SFEM*), color feature extraction module (*CFEM*), and texture feature extraction module (*TFEM*). All the submodules have multiple stages and steps for computing features from an image. The *SFEM* extracts shape features from the top view image to generate the shape feature vector (*SFV*). The *SFV* contains 8 features in total. The *CFEM* computes color features from the top view image to generate a color feature vector (*CFV*). The *CFV* contains 3 features. The *TFEM* extracts two types of texture features from the side view image to generate a texture feature vector (*TFV*). The *TFV* contains 23 features in total. The *FEM* processes two input images from two separate repositories and computes 34

features in total. This module grasps the maximum time of the total execution time for the training module.

5.2.1.4 Feature Database (FD)

The feature database is the collection of different types of features for different types of fruits and vegetables. This module is also constructed with three different sub-modules i.e. shape feature database (SFD), color feature database (CFD), texture feature database (TFD). Each of the feature databases contains a feature matrix. If any of the databases contains n number of features for m number of samples then the size of the corresponding feature matrix will be $m \times n$.

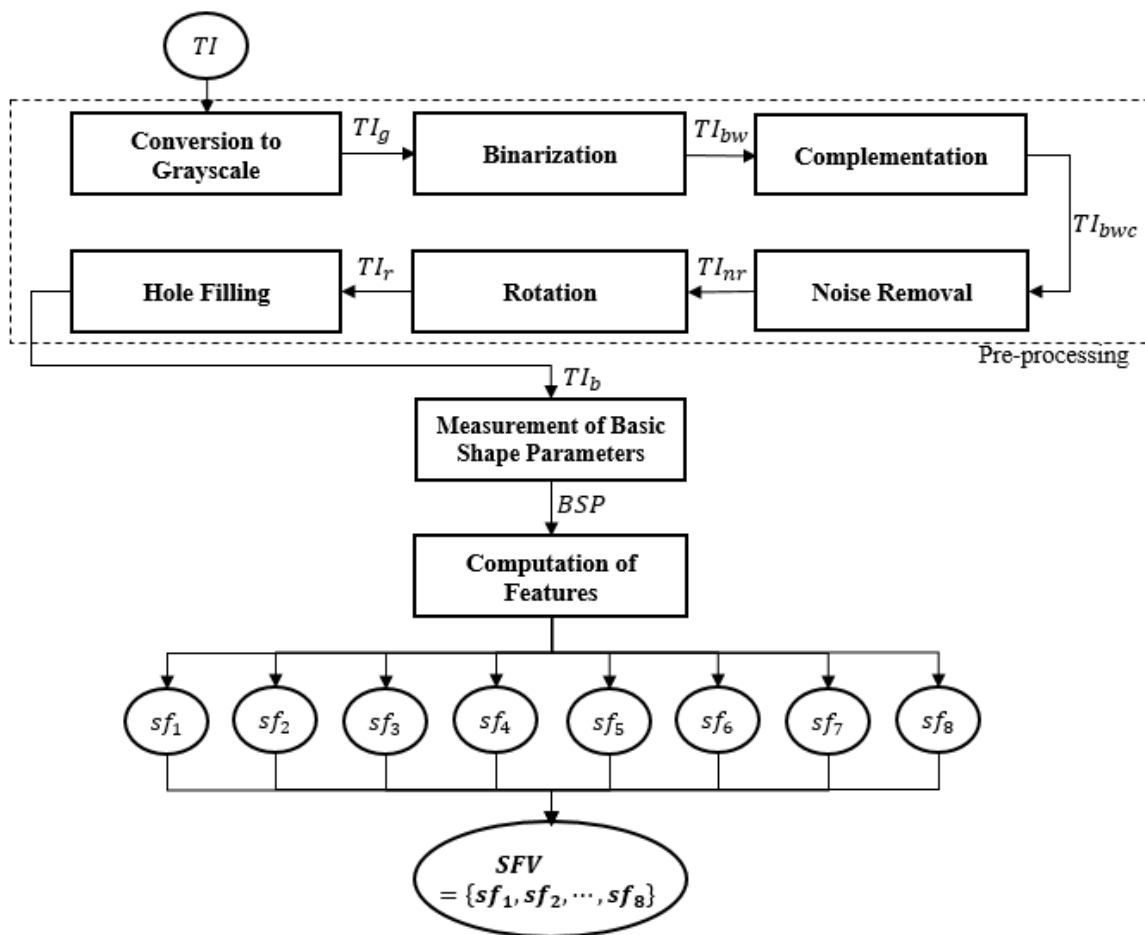


Fig. 5.4. The process flow in shape feature extraction module

5.2.1.5 Classification Module (CM)

This module plays a different role in the training phase and testing phase. In the training phase, there are two classification models i.e. classification model 1 (CLM 1) and classification model 2 (CLM 2). The models will be trained with the features of training samples. The shape feature

matrix and color feature matrix combined to generate the feature matrix for the top view (*TFM*). Assuming that there is m number of training samples. The size of the *TFM* is $m \times 11$. The *TFM* is feed into the *CLM 1* along with the class labels (*CL*) of m number of samples. The texture feature matrix is the feature matrix for the side view (*SFM*). The size of the *SFM* is $m \times 23$. The *SFM* is feed into the *CLM 2* along with the class labels (*CL*) of m number of samples. The *CLM 1* and *CLM 2* will be trained after the successful execution of this module.

5.2.2 Shape Feature Extraction Module (*SFEM*)

The shape feature extraction module takes an RGB color image (*TI*) from the top view camera. Fig. 5.4 shows the complete process flow in the *SFEM*. The image is passed through some pre-processing stages before feature extraction. Fig. 5.5 demonstrates the output of each pre-processing step with a sample of Red Delicious Apple. Some basic shape parameters (*BSP*) are measured from the pre-processed binary image. Finally, 8 shape features are computed from those basic parameters to generate a shape feature vector (*SFV*).

5.2.2.1 Pre-processing

The input RGB color image (*TI*), Fig. 5.5(a), is transformed into a grayscale image (*TI_g*) using Eq. (5.1). The grayscale image, Fig. 5.5(b) can be binarized using a fixed threshold value (T_s) as the background is white for the imaging setup. Refer to Eq. (5.2). The segmented binary image (*TI_{bw}*), Fig. 5.5(c), will have the object region as black and the background as white. The segmented image is complemented to make the white object region, Fig. 5.5(d). Refer to Eq. (5.3). There may have many noise regions, which are formed with just a few pixels. The region with maximum area is considered as the object region. The smaller noise regions are discarded by region area thresholding, Fig. 5.5(e). The fruit and vegetable objects may be oriented in any direction. The orientation angle of the major axis is measured with respect to the horizontal axis. The image is rotated with that orientation angle (θ) to normalize the images with the same orientation with respect to the major axis, Fig. 5.5(f). Refer to Eq. (5.4). There may have many holes inside the object region. A morphological filling [Soille, 1999; Dougherty et al., 2003] operation is performed to fill the inner holes of the object, Fig. 5.5(g). Refer to Eq. (5.5) and (5.6), where B is a 3×3 structuring element of ones and k is the counter of the iterative filling process. The stop condition is $X_k = X_{k-1}$.

$$TI_g = 0.2989 \times TIR + 0.5870 \times TIG + 0.1140 \times TIB \quad 5.1$$

$$TI_{bw}(x, y) = \begin{cases} 1 & \text{if } TI_g(x, y) > T_s \\ 0, & \text{otherwise} \end{cases} \quad 5.2$$

$$TI_{bwc}(x, y) = TI_{bw}(x, y)' \quad 5.3$$

$$TI_r[x, y] = TI_{bwc}[x, y] \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad 5.4$$

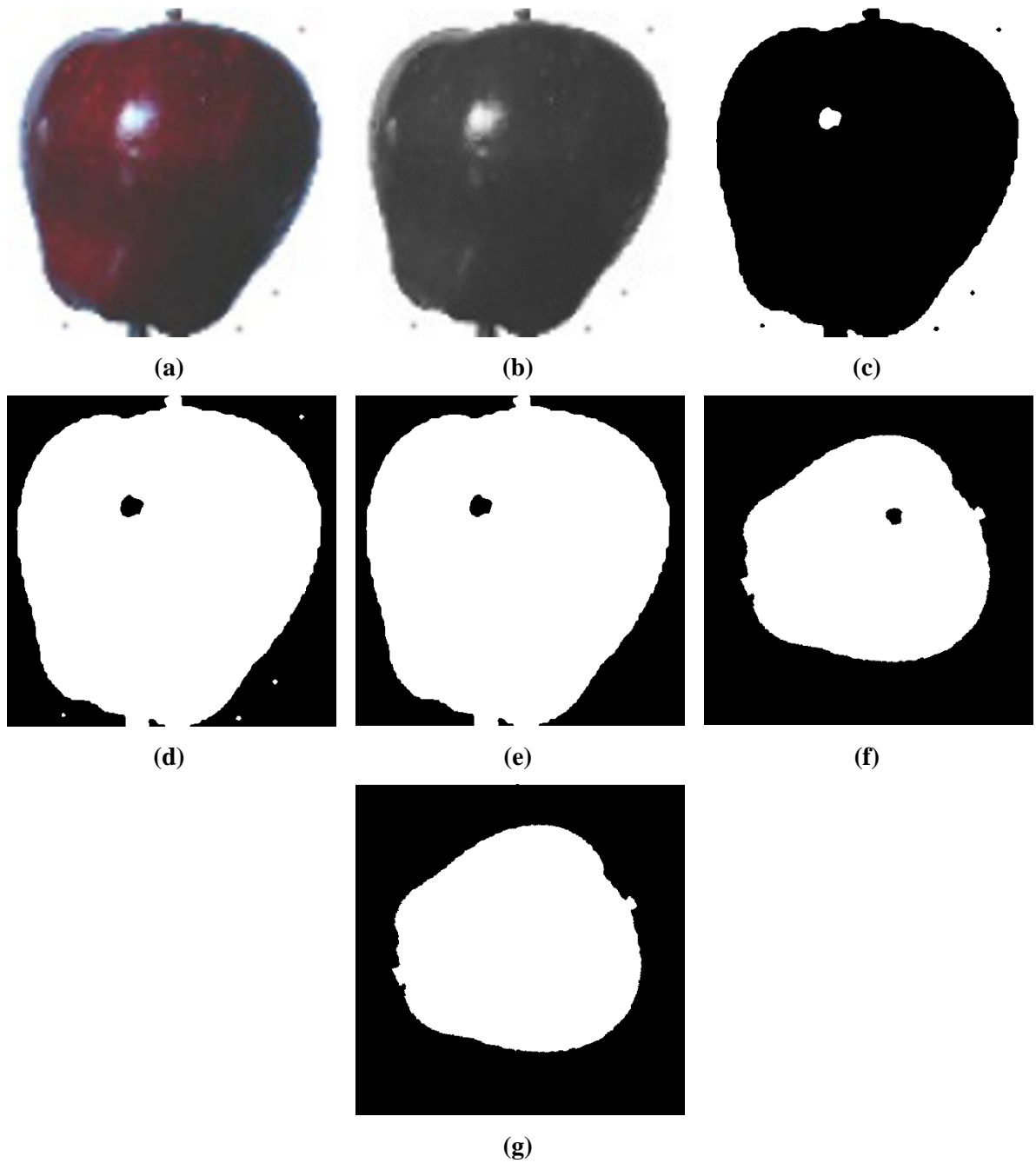


Fig. 5.5. (a) Input color (RGB) image (b) Grayscale image (c) Segmented binary image (d) Complemented binary image (e) After removing background noises (f) Rotated Image (g) After filling inner holes

$$X_k = (X_{k-1} \oplus B) \cap TI_r^c \quad 5.5$$

$$TI_b = TI_r \cup X_k \quad 5.6$$

5.2.2.2 Measurement of Basic Shape Parameters

The image (TI_b) will have a white object region after properly pre-processing the input image. Some basic shape parameters are measured from the ROI containing the region of fruits and vegetables. The area (A) and perimeter (P) are measured from the object region. The length of major axis (W), length of minor axis (H), and distance between two foci (F) are measured from the equivalent ellipse of the ROI. A minimum bounding box is drawn by enclosing the ROI.

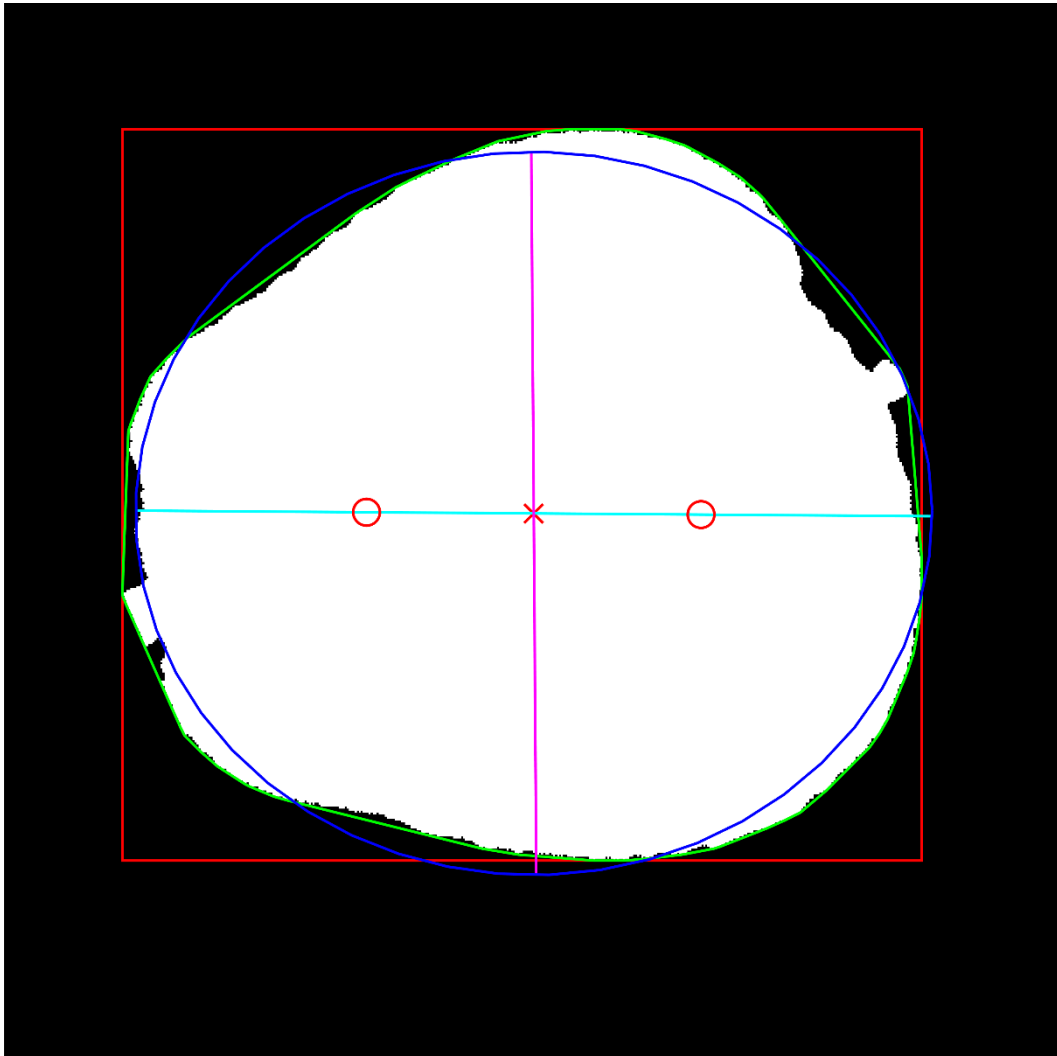


Fig. 5.6. Shape parameters for a sample

The width (W_b), height (H_b), and area (A_b) are computed for this bounding box. The smallest convex polygon is also drawn by enclosing the ROI. The area (A_c), and perimeter (P_c) are

calculated for the convex polygon. The unit of measurement is pixel for all the parameters. BSP contains the parameters. Refer to Eq. (5.7). The width and height are the measurements in horizontal and vertical directions respectively. Fig. 5.6 visualizes the parameters for a sample. The white region is the ROI. The pixels contained in the white region is the area (A) of the fruit or vegetable. The perimeter (P) is the number of border pixels in the white region. The approximated ellipse is shown in blue color. The centroid of the ellipse is marked by a red cross. The major axis and minor axis of the approximated ellipse are represented by a cyan line in the horizontal direction and a magenta line in the vertical direction respectively. The foci of the approximated ellipse are shown by two red-colored circular markers on the major axis. The red-colored rectangle represents the minimum bounding box. The smallest convex polygon is represented by green color.

$$BSP = \{A, P, W, H, F, W_b, H_b, A_b, A_c, P_c\} \quad 5.7$$

5.2.2.3 Shape Feature Computation

The parameters, which have been measured in the previous step, are dependent on many factors. The size of the fruit and vegetables changes a lot but the shape mostly remains the same. The geometric transformations are also responsible for varying those parameters. The scaling mostly causes variation in shape and size. Hence, some features have been proposed here by extending those basic shape parameters. Refer to Eqs. (5.8) to (5.15) for the eight extended features $sf_1, sf_2, sf_3, sf_4, sf_5, sf_6, sf_7,$ and sf_8 . The extended features are the ratio of similar kinds of parameters, hence those will not vary due to transformations or different growth stages of fruits and vegetables. The shape feature vector (SFV) is formed with those eight features. Refer to Eq. (5.16)

$$sf_1 = \frac{F}{W} \quad 5.8$$

$$sf_2 = \frac{H}{W} \quad 5.9$$

$$sf_3 = 1 - \frac{H_b}{W_b} \quad 5.10$$

$$sf_4 = \frac{H + W}{P} \quad 5.11$$

$$sf_5 = \frac{A}{P^2} \quad 5.12$$

$$sf_6 = \frac{A}{A_b} \quad 5.13$$

$$sf_7 = \frac{A}{A_c} \quad 5.14$$

$$sf_8 = \frac{P_c}{P} \quad 5.15$$

$$SFV = \{sf_1, sf_2, sf_3, sf_4, sf_5, sf_6, sf_7, sf_8\} \quad 5.16$$

5.2.3 Color Feature Extraction Module (CFEM)

The surface color of fruits and vegetables is a very important feature descriptor for the identification of the fruits and vegetable class. This module also takes an RGB color image (TI) of fruits and vegetables from the top view camera as an input.

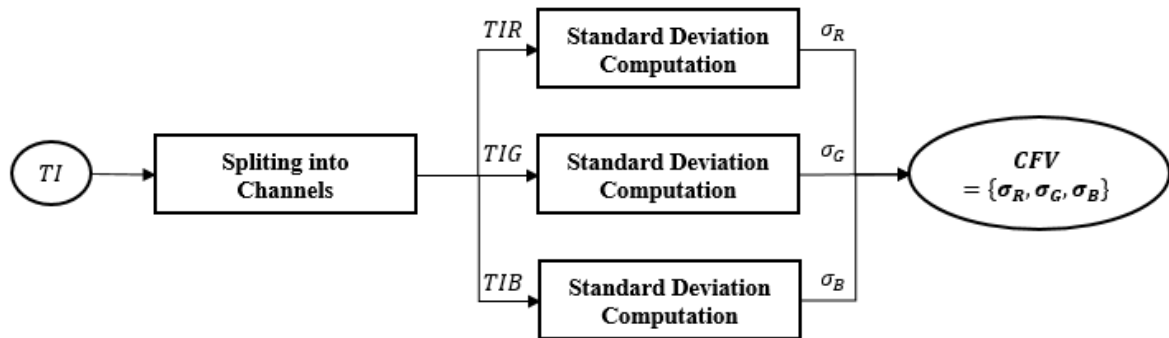


Fig. 5.7. Process flow in color feature extraction module

Hence, the color information in RGB color space is used as a classification feature. Fig. 5.7 depicts the process flow for color feature extraction and color feature vector generation. The input image has been split into three color channels i.e. TIR , TIG , and TIB . The standard deviation of the color channels is computed using Eqs. (5.17), (5.18) and (5.19) for red, green, and blue channels respectively, where μ_R , μ_G , and μ_B are the mean for the corresponding channels. σ_R , σ_G , and σ_B are the standard deviation of TIR , TIG , and TIB respectively. The color feature vector (CFV) is formed with σ_R , σ_G , and σ_B . Refer to Eq. (5.20).

$$\sigma_R = \sqrt{\frac{\sum_{k=0}^{N-1} (TIR_k - \mu_R)^2}{N}} \quad 5.17$$

$$\sigma_G = \sqrt{\frac{\sum_{k=0}^{N-1} (TIG_k - \mu_G)^2}{N}} \quad 5.18$$

$$\sigma_B = \sqrt{\frac{\sum_{k=0}^{N-1} (TIB_k - \mu_B)^2}{N}} \quad 5.19$$

$$CFV = \{\sigma_R, \sigma_G, \sigma_B\} \quad 5.20$$

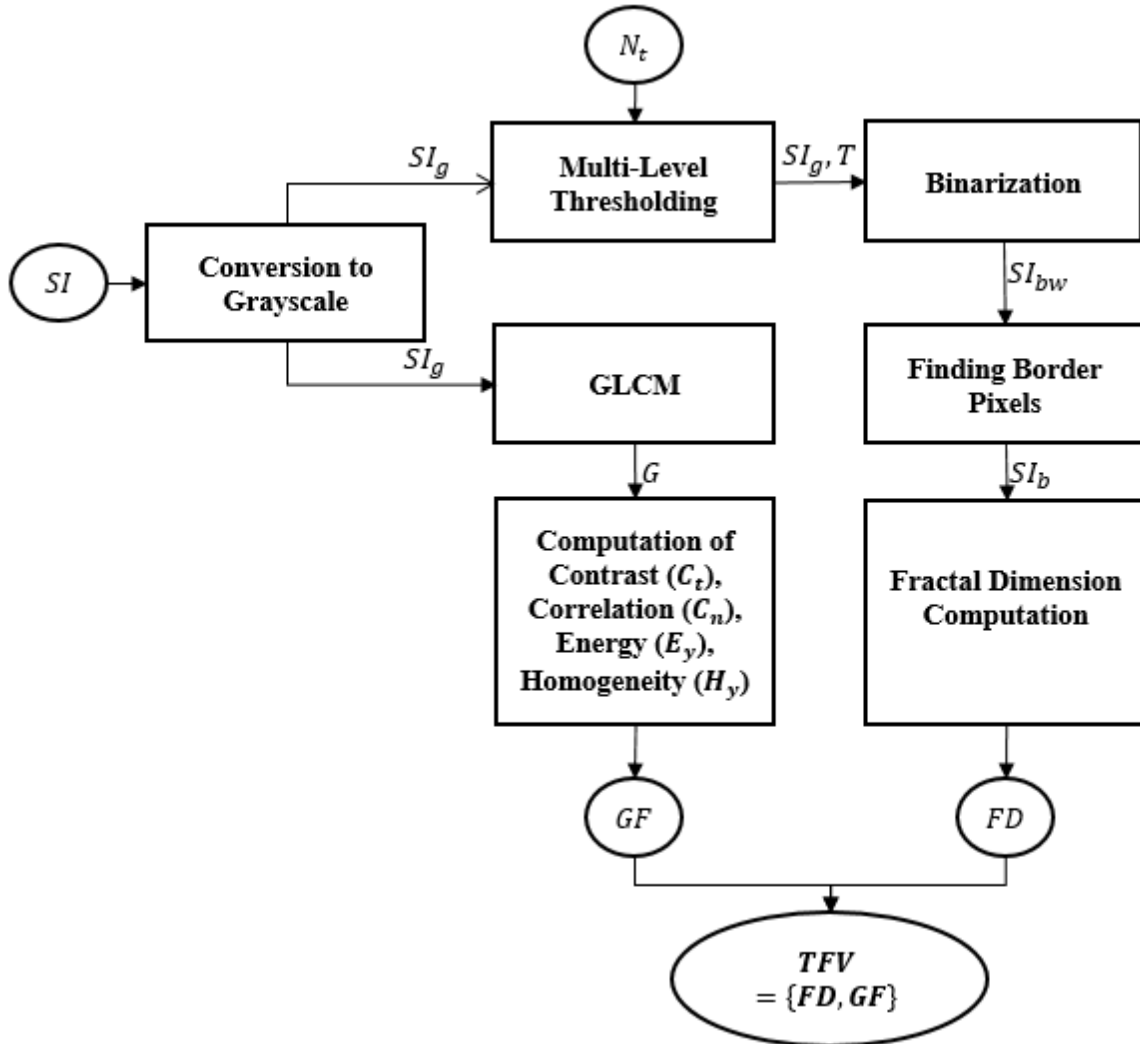


Fig. 5.8. Process flow of texture feature extraction module

5.2.4 Texture Feature Extraction Module (TFEM)

The texture is a very powerful feature descriptor for image classification. Two types of texture features are extracted here for the classification of fruits and vegetables. The first one is the fractal dimension of fruit or vegetable surface and the second one is statistical measures from GLCM (Gray level co-occurrence matrix). Fig. 5.8 depicts the process flow of the entire texture feature extraction module. The texture is the least changing feature when the fruit or vegetable is rotated in the horizontal plane keeping the major axis along the same plane. Hence, only the texture features have been extracted from the side view image (*SI*).

5.2.4.1 Fractal Dimension Computation

The fractal dimension has been discovered by Mandelbrot [Mandelbrot, 1967]. It is a never-ending pattern of breaking a shape into smaller pieces with a fixed scale factor. It is mathematically computed using Eq. (5.21). Here, N is the number of smaller pieces from a larger one after breaking and F is the scale factor of comparing smaller pieces with the larger one.

$$D = \frac{\log N}{\log (1/F)} \quad 5.21$$

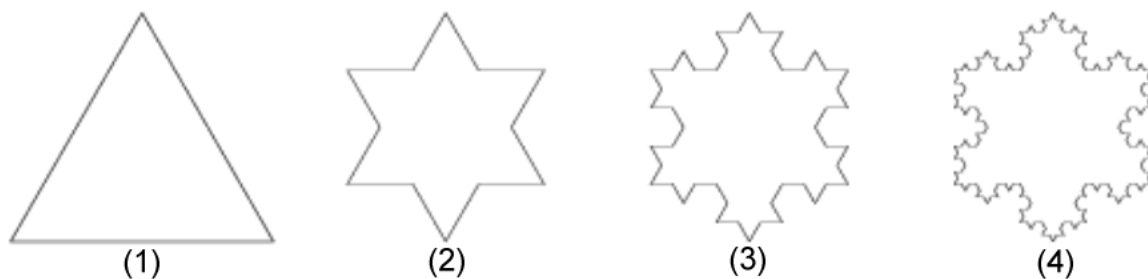


Fig. 5.9. First four iterations of Koch Snowflake [Francis, 2012]

Koch Snowflake [Koch, 1904] is an example of a fractal pattern. It is considered here to demonstrate the calculation process. Fig. 5.9 shows the first four iterations of Koch Snowflake. In each iteration, every side of the triangle is broken into 4 smaller pieces and the size is exactly $1/3$ of the original length. The dimension of Koch Snowflake using this formula is 1.231. This dimension is a fraction, not an integer. Hence this is called the fractal dimension.

Hausdorff's box-counting method [Costa et al., 2012; Feng et al., 1996] is used here to calculate the fractal dimension of fruit and vegetable surface. The detailed process flow of fractal

dimension computation has been shown in Fig. 5.8 in a parallel flow with GLCM features extraction.

- *Conversion to Grayscale:* RGB input image (SI), Fig. 5.10(a), is split into three components i.e. SIR , SIG , SIB . The input RGB image is converted into a grayscale image (SI_g), Fig. 5.10(b), using Eq. (5.1).



Fig. 5.10. (a) Input RGB color image (b) Grayscale image

- *Multi-level Thresholding:* Otsu thresholding is a very popular global thresholding technique. Hence, multilevel Otsu thresholding [Liao et al., 2001] is applied to the grayscale image. This algorithm runs in a very low storage space with high processing speed. It is assumed that the grayscale image (SI_g) contains M number of pixels, which has a gray level from 1 to L . If f_i represents the total number of the pixels in i^{th} gray level then the probability of the i^{th} gray level (p_i) in the image is computed using Eq. (5.22). The number of threshold level (N_t) should be provided as an input to generate a set T , which contains N_t number of threshold values i.e. $\{T_1, T_2, T_3, \dots, T_{N_t}\}$, based on the histogram distribution in gray level. The image will be divided into k number of classes where $k = (N_t + 1)$. The cumulative probability of k^{th} class (ω_k) is computed using Eq. (5.23). The mean intensity of k^{th} class (μ_k) is computed using Eq. (5.24). Therefore, the mean intensity of the whole image (μ_T) and between-class variance (σ_B^2) is computed using the Eqs. (5.25) and (5.26) respectively. The between-class variance (σ_B^2) is maximized to select the set of optimum threshold values $\{T_1^*, T_2^*, T_3^*, \dots, T_{N_t}^*\}$. Refer to Eq. (5.27). The optimum threshold values are stored in ascending order.

$$p_i = \frac{f_i}{M} \tag{5.22}$$

$$\omega_k = \sum_{i \in C_k} p_i \quad 5.23$$

$$\mu_k = \sum_{i \in C_k} \frac{i \cdot p_i}{\omega_k} \quad 5.24$$

$$\mu_T = \sum_{i=1}^L i \cdot p_i = \sum_{k=1}^K \mu_k \omega_k \quad 5.25$$

$$\sigma_B^2 = \sum_{k=1}^K \omega_k (\mu_k - \mu_T)^2 \quad 5.26$$

$$\{T_1^*, T_2^*, T_3^*, \dots, T_{N_t}^*\} = \text{Arg Max}\{\sigma_B^2(T_1, T_2, T_3, \dots, T_{N_t})\},$$

$$1 \leq T_1 < \dots < T_{N_t} < L \quad 5.27$$

- *Binarization*: The grayscale image is binarized by thresholding with all the threshold values of set T . Refer to Eq. (5.28). It will generate N_t number of binary images. Fig. 5.11 shows the sample binary images generated from Fig. 5.10(b) by thresholding with those fixed threshold values where $N_t = 4$.

$$SI_{bw}(x, y) = \begin{cases} 1 & \text{if } SI_g(x, y) > T \\ 0 & \text{otherwise} \end{cases} \quad 5.28$$

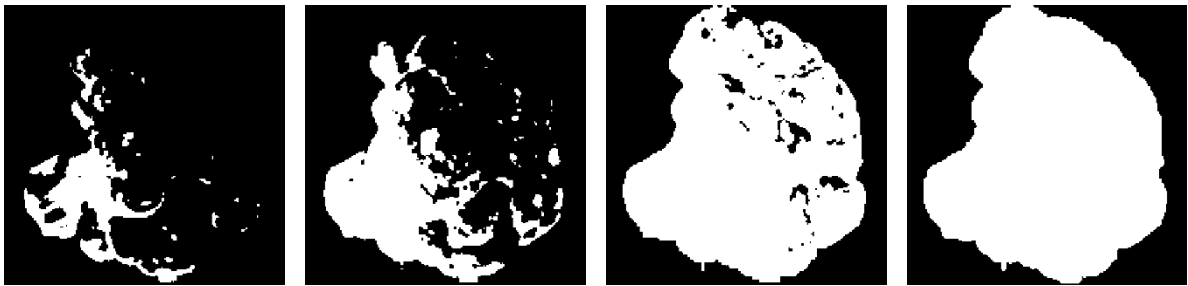


Fig. 5.11. Binary images generated from Fig. 5.10(b) using Eq. (5.28) with $N_t = 4$

The set T is already sorted. Hence, a range of threshold is taken by selecting two consecutive threshold values i.e. lower threshold (T_l) and higher threshold (T_h). There will be $(N_t - 1)$ number of threshold ranges. The grayscale image is binarized with a range of threshold values. Refer to Eq. (5.29). It will generate the same number of binary images as the number of

threshold ranges. Fig. 5.12 shows the sample binary images generated from Fig. 5.10(b) by using a ranged threshold where $N_t = 4$.

$$SI_{bw}(x, y) = \begin{cases} 1 & \text{if } SI_g(x, y) > T_l \text{ and } SI_g(x, y) < T_h \\ 0, & \text{otherwise} \end{cases} \quad 5.29$$

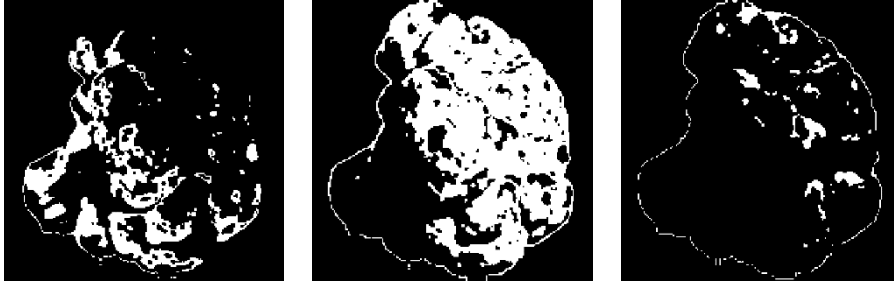


Fig. 5.12. Binary images generated from Fig. 5.10(b) using Eq. (5.29) with $N_t = 4$

In total, there will be NB number of binary images after completion of the binarization process. The value of NB can be computed using Eq. (5.30). Each of the binary images contains detailing of the fruit or vegetable surface for a particular threshold value or range.

$$NB = 2(N_t) - 1 \quad 5.30$$

- *Finding Border Pixels:* The binary images have been passed through a border extraction process. A pixel $SI_{bw}(x, y)$ will be marked as a border pixel if it has value 1 and any of its 8 neighbor pixels has value 0. The pixel, which does not satisfy this condition, is the non-border pixel. The value of the border pixel will remain as 1. The value for non-border pixels is set as 0. Border image (SI_b) is generated from each of the binary image (SI_{bw}) using Eq. (5.31). Fig. 5.13 and 5.14 show the sample border images generated from the images in Fig. 5.11 and Fig. 5.12 respectively. The border images of a fruit or vegetable surface contain a specific fractal pattern in different threshold levels.

$$SI_b(x, y) = \begin{cases} 1 & \text{if } SI_{bw}(x, y) = 1 \text{ and } SI_{bw}(x', y') = 0 \\ & \text{where } (x', y') \in N_8[(x, y)] \\ 0, & \text{otherwise} \end{cases} \quad 5.31$$

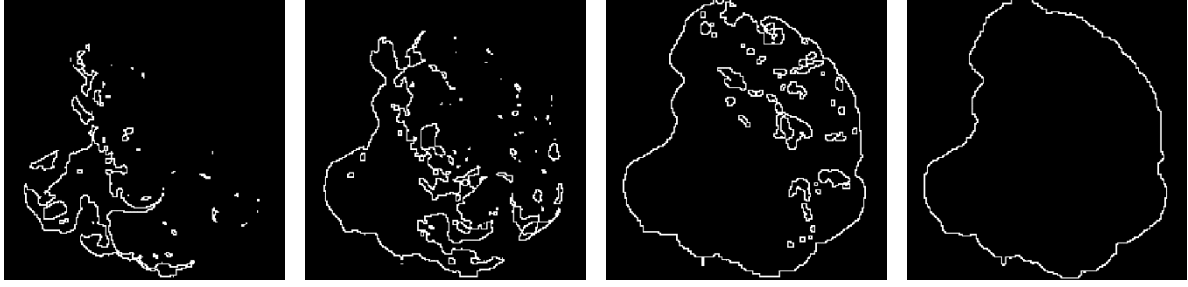


Fig. 5.13. Border images generated from the images in Fig. 5.11



Fig. 5.14. Border images generated from the images in Fig. 5.12

- *Dimension Computation:* It has been assumed that the size of the border image (SI_b) is $(H \times W)$. Hausdorff's box-counting method needs an initial box size to start the algorithm. Initial box size (S) can be computed using Eq. (5.32). The border image is padded $(S - H)$ number of rows and $(S - W)$ number of columns with $SI_b[H \times W]$. Now, the size of the border image becomes $S \times S$. The resized border image will be divided into multiple square grids of size $\epsilon \times \epsilon$. It will count the number of box ($\bar{N}(\epsilon)$) which contains at least one pixel of the object. In the first iteration, the number of the box will be 1 as the initial value of ϵ is S . In the second iteration, the size of the box has been scaled down to half of the first iteration i.e. $\epsilon/2$ in both directions. Again, the counting will be performed for the number of the box containing at least one pixel of the object. A similar process will be continued iteratively. The iteration will stop when $\epsilon \leq 1$. In every iteration, we will have a coordinate point of (x, y) . The value of x will be computed as $\log(1/\epsilon)$ and the value of y as $\log N(\epsilon)$. A least square method is applied to the coordinate points to fit those points into a straight line. The slope of the line is the fractal dimension of the border image (SI_b). Refer to Eq. (5.33) to compute the slope. The fractal dimension is computed for all the border image generated from a side view image (SI). The dimensions are stored in a set FD as shown in Eq. (5.34).

$$S = 2^{\lceil \log_2(\max(H, W)) \rceil} \quad 5.32$$

$$D = \frac{\Delta \{\log N(\epsilon)\}}{\Delta \{\log(1/\epsilon)\}} \quad 5.33$$

$$FD = \{D_1, D_2, D_3, \dots, D_{NB}\} \quad 5.34$$

Algorithm for Fractal Dimension Computation:

Input: Grayscale Image (SI_g) and Number of threshold level (N_t)

Output: Fractal Dimensions (D)

- 1 Calculate (N_t) number of optimum threshold values using multilevel Otsu thresholding technique and store them in set T in ascending order
- 2a *for* $i = 1$ to N_t
Generate a binary image (SI_{bwi}) for the threshold value T_i using Eq. (5.28)
end for
- 2b *for* $j = 1$ to $(N_t - 1)$
Generate a binary image ($SI_{bw(N_t+j)}$) for the threshold range T_j to T_{j+1} using Eq. (5.29)
end for
- 3 *for* $k = 1$ to $(2N_t - 1)$
Generate border image (SI_{bk}) from each of the binary image (SI_{bwk}) using Eq. (5.31)
end for
- 4 *for* $k = 1$ to $(2N_t - 1)$
Compute Fractal Dimension (D_k) for the border image (SI_{bk}) using Eq. (5.33)
end for

5.2.4.2 GLCM features

Spatial texture features are very important for image classification. A Gray-level Co-occurrence Matrix (GLCM) [Haralick et al., 1973; Capizzi et al., 2015] is formed to demonstrate the spatial relationships among the pixels. If the total number of intensity levels in grayscale is L then GLCM is the probability of occurrence of intensity i to its neighbor j , which is positioned in a distance d and angle θ . Refer to Eq. (5.35). The dimension of the G depends on the number of gray levels (i.e. 256 x 256).

$$G = Pr(i, j | d, \theta, L) \quad 5.35$$

A GLCM cannot be directly used as a classification feature. The extracted statistical texture properties work as a good image classification feature. Hence, the transpose of GLCM is added to the original GLCM to make it symmetrical. The GLCM is normalized by dividing all elements by the sum of the GLCM. Contrast (C_t), Correlation (C_n), Energy (E_y) and

Homogeneity (H_y) is computed from normalized symmetrical GLCM as shown in Eqs. (5.36) to (5.39). $S(i, j)$ is the value of $(i, j)^{th}$ position in the symmetrical normalized GLCM, and the range of i, j is 1 to L . The features are extracted with a distance of 1 pixel in four directions i.e. $0^\circ, 45^\circ, 90^\circ$ and 135° . The statistical features are the scalar value contains a specific pattern for a class of fruit and vegetable.

$$C_t = \sum_{i,j=1}^L |i - j|^2 S(i, j) \quad 5.36$$

$$C_n = \sum_{i,j=1}^L \frac{(i - \mu_i)(j - \mu_j)S(i, j)}{\sigma_i \sigma_j} \quad 5.37$$

$$E_y = \sum_{i,j=1}^L S(i, j)^2 \quad 5.38$$

$$H_y = \sum_{i,j=1}^L \frac{S(i, j)}{1 + |i - j|} \quad 5.39$$

The feature vector for GLCM (GF) is formed with Contrast (C_t), Correlation (C_n), Energy (E_y) and Homogeneity (H_y) in four directions. Refer to Eq. (5.40). The features vector from GLCM (GF) is combined with the feature vector from the fractal dimension (FD). The final texture feature vector (TFV) is the combination of GF and FD . Refer to Eq. (5.41).

$$GF = \{Ct_{0^\circ}, Ct_{45^\circ}, Ct_{90^\circ}, Ct_{135^\circ}, Cn_{0^\circ}, Cn_{45^\circ}, Cn_{90^\circ}, Cn_{135^\circ}, Ey_{0^\circ}, Ey_{45^\circ}, Ey_{90^\circ}, Ey_{135^\circ}, Hy_{0^\circ}, Hy_{45^\circ}, Hy_{90^\circ}, Hy_{135^\circ}\} \quad 5.40$$

$$TFV = \{FD, GF\} \quad 5.41$$

5.2.5 Classification Module (CM)

The classification module is responsible for doing the most important task of prediction. The classification module contains two classification models i.e. *CLM 1* and *CLM 2*. These models are based on a supervised machine learning algorithm. Supervised learning works on the priory labeled dataset. The proposed framework will classify the known fruits and vegetable types, hence the framework has been structured with the supervised classification models. A very popular classification algorithm has been used here as a classification model i.e. k-Nearest Neighbor (kNN). It is one of the simplest but efficient classification algorithm. It is suitable for

a large dimension of data. Here, the dimension of the feature vector is 34. Hence the kNN is an appropriate choice for the classification. It does prediction based on entire training data. The steps of kNN are mentioned here in detail.

- *Steps of kNN :*

1. Assume that there is m number of samples in the training dataset and n number of features.
2. The unknown test sample has also n number of features.
3. Calculate the distance between the test sample and each of the training samples in terms of features. There are many distance measures i.e. Euclidean, city block, correlation, cosine, etc.
4. Sort the training samples in the ascending order of distance from the test sample.
5. Check the class label of the first k number of training samples from the sorted list.
6. Select the class label, which has a maximum vote among first k training samples, as the predicted label for the test sample.

5.2.6 Testing Phase

The main task of the testing phase is to take an unknown test sample as input and label the sample as the output. The training phase gives the trained classification models. The testing utilizes the trained models to predict the class of unknown test samples. The testing phase has four components i.e. Imaging Module (IM), Feature Extraction Module (FEM), Classification Module (CM), and Final Prediction Module (FPM). Fig. 5.15 shows the detailed architecture of the testing phase.

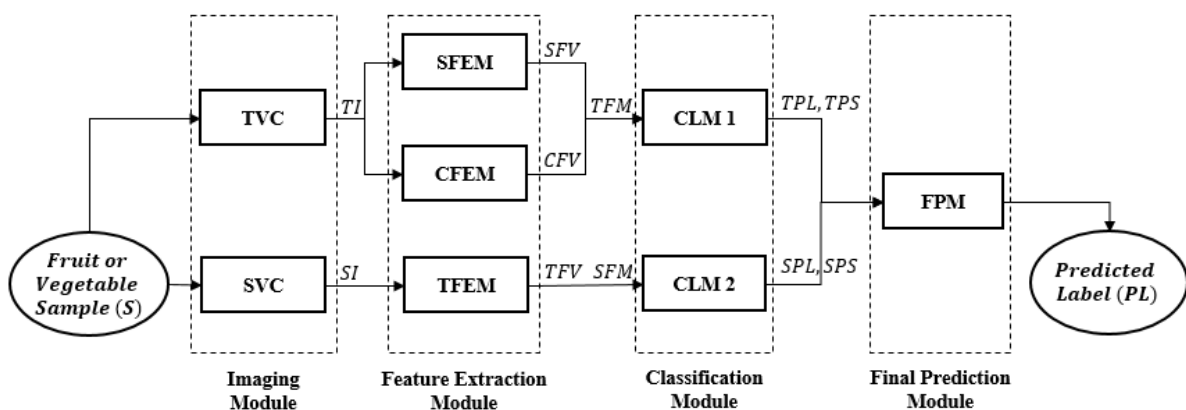


Fig. 5.15. The detailed architecture of the testing phase

5.2.6.1 Imaging Module (*IM*)

The imaging module in the testing phase contains two cameras in a perpendicular position. The test samples are placed in the same imaging setup as the training phase. The top view camera (*TVC*) captures the top view image (*TI*) of the test sample. The side-view camera (*SVC*) captures the side view image (*SI*) of the test sample. The *TI* and *SI* are sent to the feature extraction module (*FEM*).

5.2.6.2 Feature Extraction Module (*FEM*)

This module is incorporated in the testing phase to extract the same features from the test sample as in the training phase. This module contains submodules like shape feature extraction module (*SFEM*), color feature extraction module (*CFEM*), and texture feature extraction module (*TFEM*). The *SFEM* and *CFEM* receive a copy of *TI*. The *SFEM* generates the shape feature vector (*SFV*) and *CFEM* generates the color feature vector (*CFV*). The *SI* is sent to *TFEM* to generate the texture feature vector (*TFV*). The feature, sequence, and lengths of *SFV*, *CFV*, and *TFV* are similar as in the training phase. The *SFV* with 8 features and *CFV* with 3 features are combined in a feature matrix, which is the feature matrix top view image (*TFM*). Here, the dimension of *TFM* is 1×11 . The *TFV* with 23 features is converted to a features matrix for side view (*SFM*). Here, the dimension of *SFM* is 1×23 .

5.2.6.3 Classification Module (*CM*)

The classification module in the testing phase contains two trained classification models i.e. *CLM 1* and *CLM 2*. The top view feature matrix (*TFM*) is given to the trained *CLM 1* and the side view feature matrix (*SFM*) is given to trained *CLM 2*. The trained classifiers produce the predicted levels and prediction scores. The prediction score of a class means the probability of belonging to that class. The prediction score is generated for all the classes. The class with the maximum probability of prediction is selected as the predicted level. The predicted level is the predicted class for the test sample. The trained *CLM 1* produces top view prediction level (*TPL*) and prediction score (*TPS*). The trained *CLM 2* produces side view prediction level (*SPL*) and prediction score (*SPS*).

5.2.6.4 Final Prediction Module (*FPM*)

The final prediction module receives the predicted level and prediction score from trained *CLM 1* and trained *CLM 2* for top view and side view respectively. This module has two steps specially designed to make the final decision about the predicted class label. Fig. 5.16 shows

the flow of this decision-making process. In the first step, it checks whether the *TPL* and *SPL* are equal or not. If *TPL* and *SPL* are equal, then the final predicted level (*FPL*) will be either *TPL* or *SPL*. If *TPL* not equals to *SPL*, then it comes to the second condition checking. In the second step, it checks whether the prediction score of *TPL* is greater than or equals to the prediction score of *SPL*. If the prediction score for *TPL* (*TPS*) is greater than the prediction score for *SPL* (*SPS*) then *FPL* will same as *TPL* otherwise the *FPL* will be equals to *SPL*.

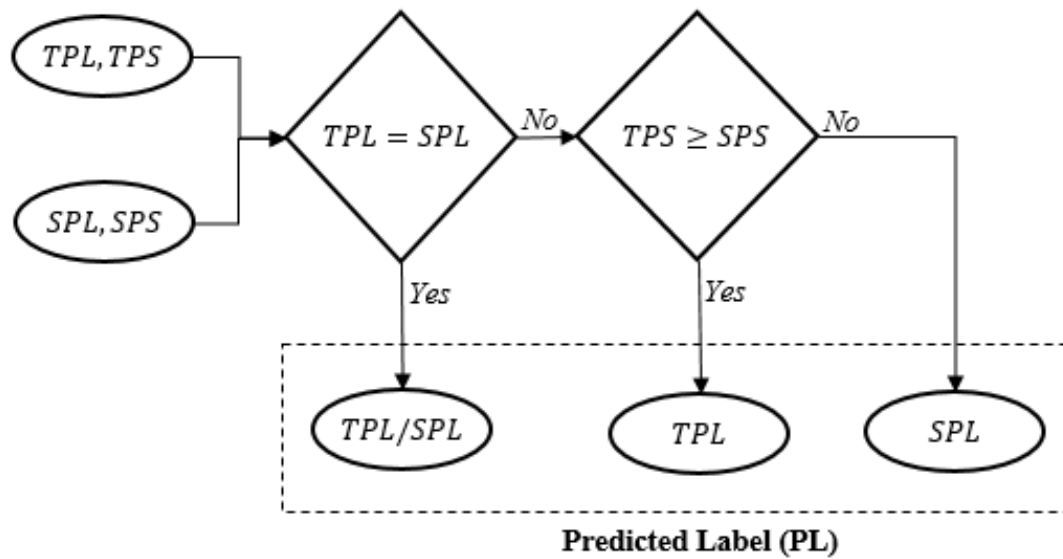


Fig. 5.16. The detailed flow of the final prediction module

5.3 Dataset

The dataset [Mureşan et al., 2018] is collected from a source that is freely available on the internet. The fruits and vegetables are placed with a shaft of speed at 3 rpm. The white background has been arranged. The RGB color images are extracted from video frames, which were taken by rotating the shaft around the vertical axis. The dataset has been divided into two sub-datasets i.e. top view dataset and side view dataset. The top view images are placed in the top view dataset and side view images are placed in the side view dataset for all the classes. All the images are scaled to fit in size [100 100] and saved in jpg format. The scaling is done for all the images to maintain uniformity throughout the dataset. There is no restriction that the images should be resized to only [100 100]. It can be resized to any size but it should be uniform throughout the dataset. Both the dataset contains more than a hundred labels. Here, we have taken 35 types of fruits and vegetables for experimentation. Table 5.1 depicts the 35 types of

fruits and vegetables along with the number of samples for the corresponding fruits and vegetable class.

Table 5.1. Class labels and number of images for the corresponding class

Class Label	Top View Dataset			Side View Dataset		
	Total Number of Samples	Training Samples	Testing Samples	Total Number of Samples	Training Samples	Testing Samples
Apple Golden	328	150	178	328	150	178
Apple Red Delicious	328	150	178	328	150	178
Apricot	328	150	178	328	150	178
Avocado	242	150	92	242	150	92
Banana	200	150	50	200	150	50
Beetroot	200	150	50	200	150	50
Blueberry	308	150	158	308	150	158
Capsicum Green	296	150	146	296	150	146
Capsicum Red	296	150	146	296	150	146
Capsicum Yellow	296	150	146	296	150	146
Carambola	328	150	178	328	150	178
Cauliflower	312	150	162	312	150	162
Cherry	328	150	178	328	150	178
Cocos	328	150	178	328	150	178
Eggplant	312	150	162	312	150	162
Guava	328	150	178	328	150	178
Kiwi	294	150	144	294	150	144
Lemon	328	150	178	328	150	178
Limes	320	150	170	320	150	170
Lychee	328	150	178	328	150	178
Mango	328	150	178	328	150	178
Melon	328	150	178	328	150	178
Mulberry	328	150	178	328	150	178
Onion Red	200	150	50	200	150	50
Orange	311	150	161	311	150	161
Peach	328	150	178	328	150	178
Pear	328	150	178	328	150	178
Pineapple	328	150	178	328	150	178
Pomegranate	328	150	178	328	150	178
Potato Red	200	150	50	200	150	50
Potato White	200	150	50	200	150	50
Raspberry	328	150	178	328	150	178
Strawberry	328	150	178	328	150	178
Tomato Maroon	205	150	55	205	150	55
Tomato Red	311	150	161	311	150	161

There is some type of fruits and vegetables that has few intra-class varieties. The subtypes are also treated as separate classes. The count of 35 labels includes classes, as well as intra-class variety for few classes, e.g. Apple, has two labels i.e. Apple Golden and Apple Red Delicious. The complete dataset contains 20814 images in total. Fig. 5.17 shows one sample for each of

the classes from the top view dataset. Fig. 5.18 shows one sample for each of the classes from the side view dataset.

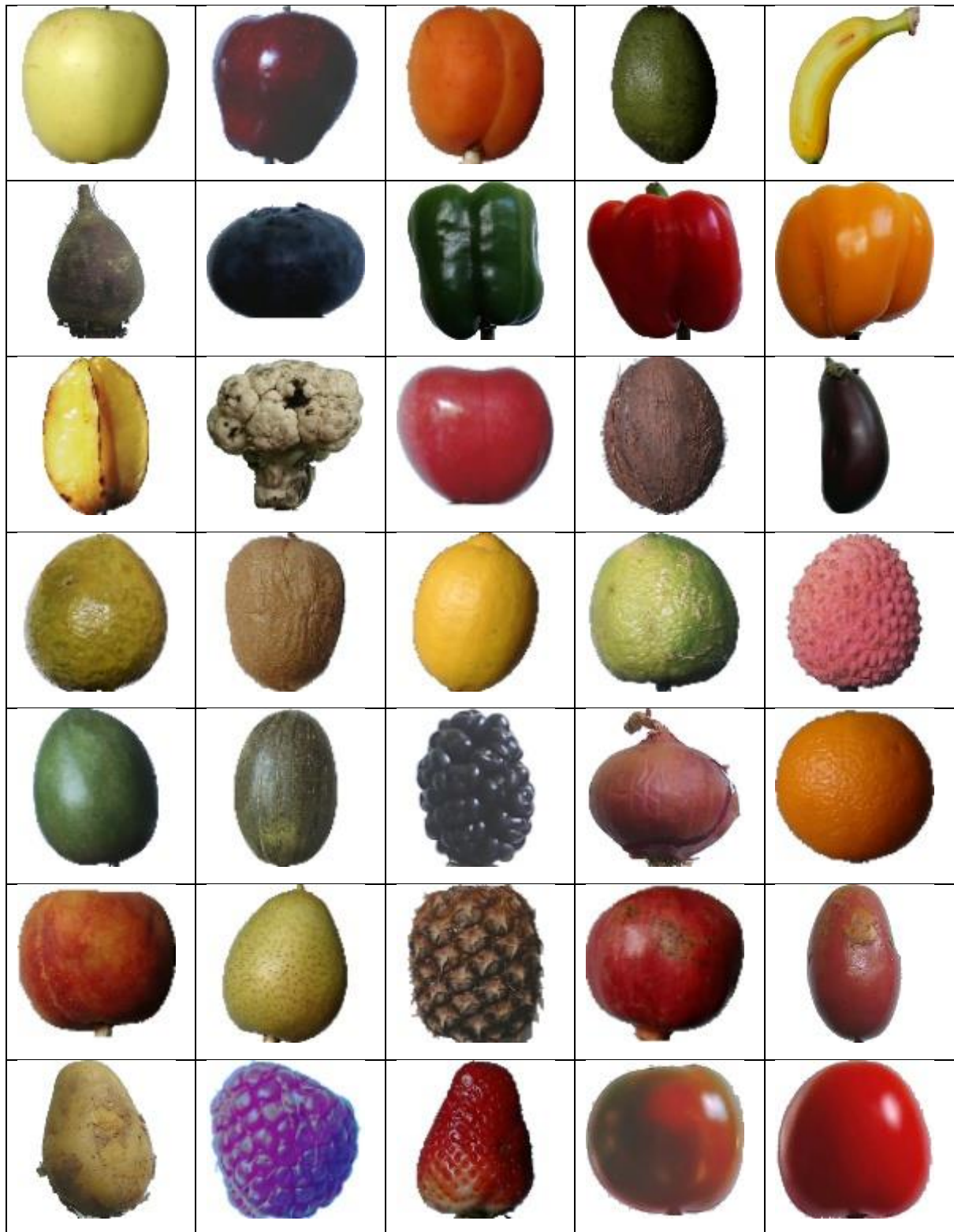


Fig. 5.17. Sample images for each class from the top view

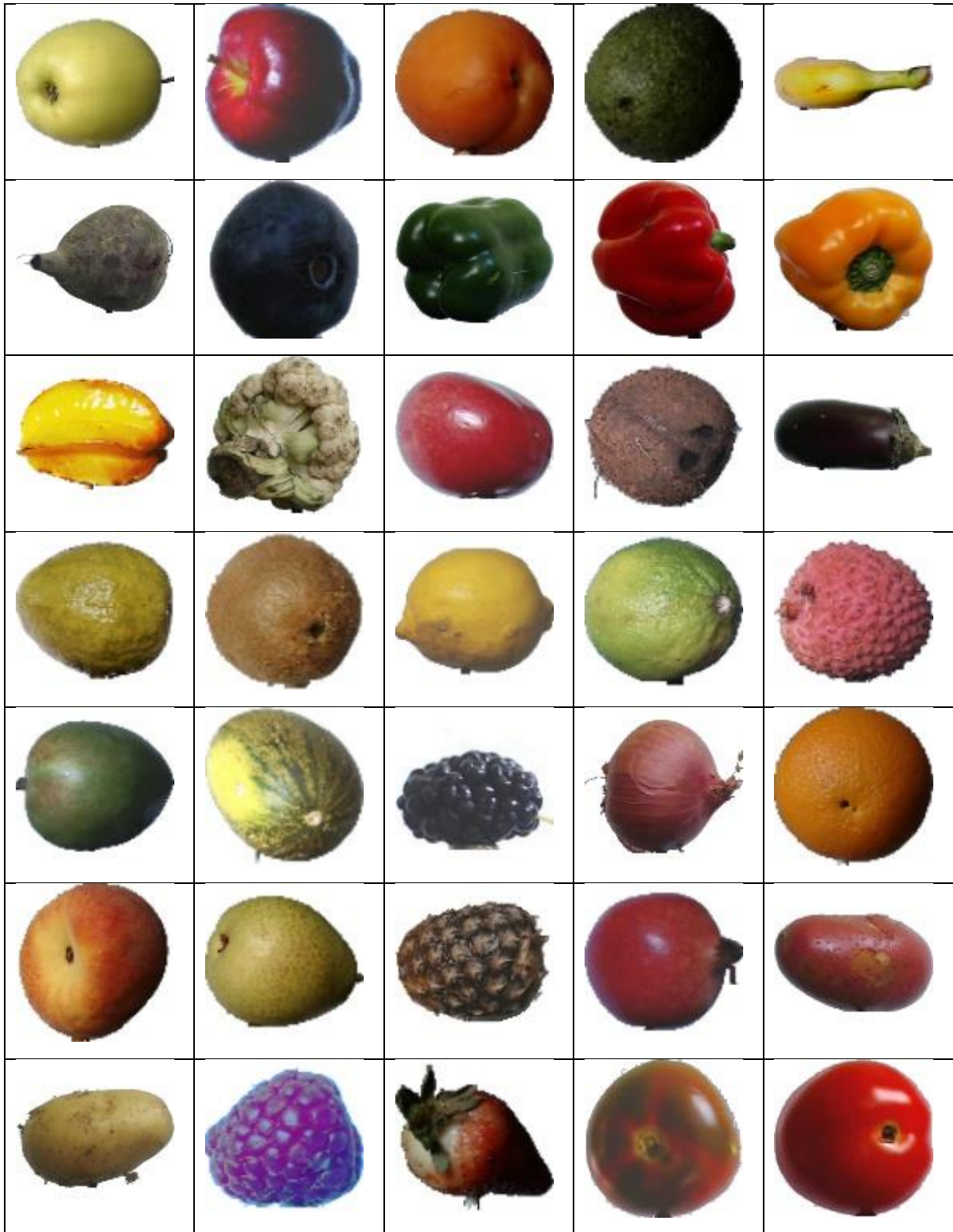


Fig. 5.18. Sample images for each class from the side view

5.4 Experimentations and Results

A lot of experimentation has been carried out to build and test the robustness of this framework. The proposed approach has been implemented in Matlab 2018a. The system has been

configured with an Intel Core i5 processor, 4GB RAM, and Windows 10 operating system. The dataset has two subdirectories i.e. top view directory, side view directory. Both the directory contains images of 35 classes. The images of all the class labels are randomly divided into two parts i.e. training and testing. Table 5.1 shows the division of the number of images in training and testing. 12 random classes have been chosen to see the discriminability of features among different classes. 50 training samples are chosen randomly from each of the 12 classes. Fig. 5.19 to Fig. 5.40 depicts the ability of discrimination among different classes using different individual features, which has been identified in this work. The X-axis represents the index of image files. The Y-axis represents the value of the corresponding features. The colored line represents a particular class as mentioned in the legend. Fig. 5.19 to Fig. 5.26 visualizes 8 shape features on top view images. As the shape features are the ratio of basic shape parameters, hence it is not varying much across the samples of the same class. Fig. 5.27 to Fig. 5.29 depicts the color features on top view images. The plots show that the discriminability is very good using standard deviation using the different color channels in RGB color space. The skewness and kurtosis on different color channels are also plotted in this context, but the discrimination power is very poor compared with the standard deviation. Fractal dimensions have been computed for a different threshold level of side view images. Fig. 5.30 to Fig. 5.36 presents the discriminability of fractal dimensions on different threshold levels or ranges. It is also observed that the discriminability of the fractal dimension from a lower threshold range is better than the same from a higher threshold range. GLCM features i.e. Contrast (C_t), Correlation (C_n), Energy (E_y) and Homogeneity (H_y) have been computed from side view images. Fig. 5.37 to Fig. 5.40 presents those four GLCM features with a distance of 1 pixel and an angel of 0° .

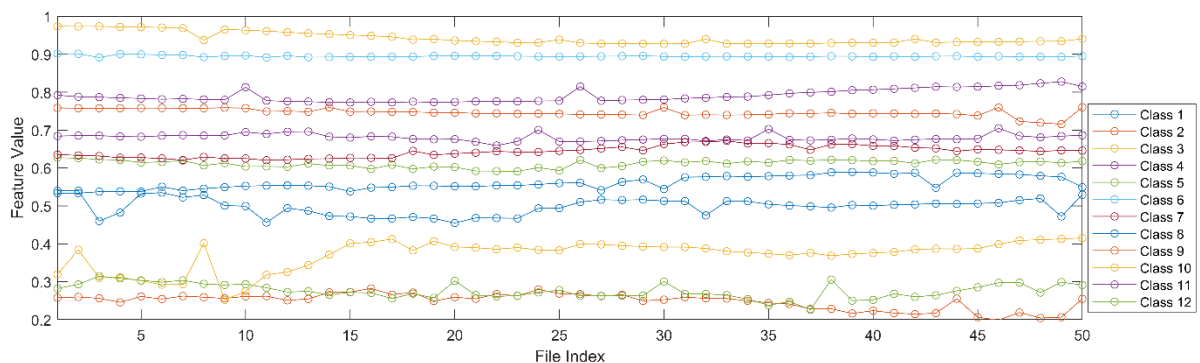


Fig. 5.19. Plot for sf_1

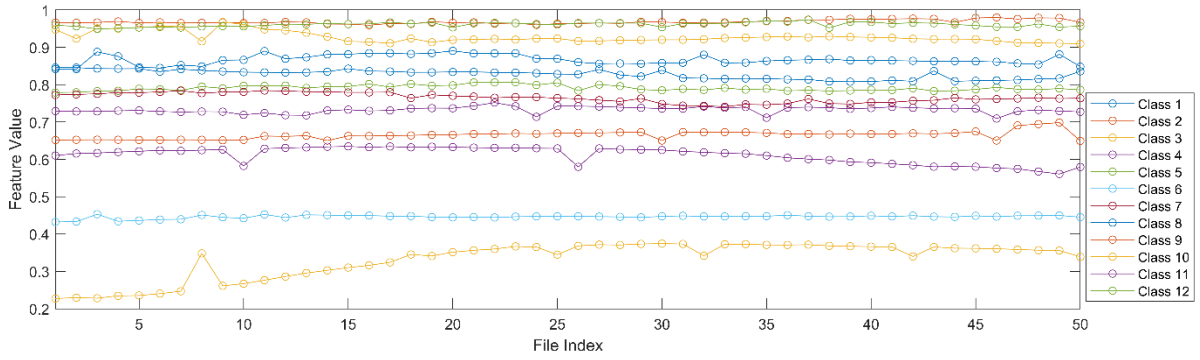


Fig. 5.20. Plot for sf_2

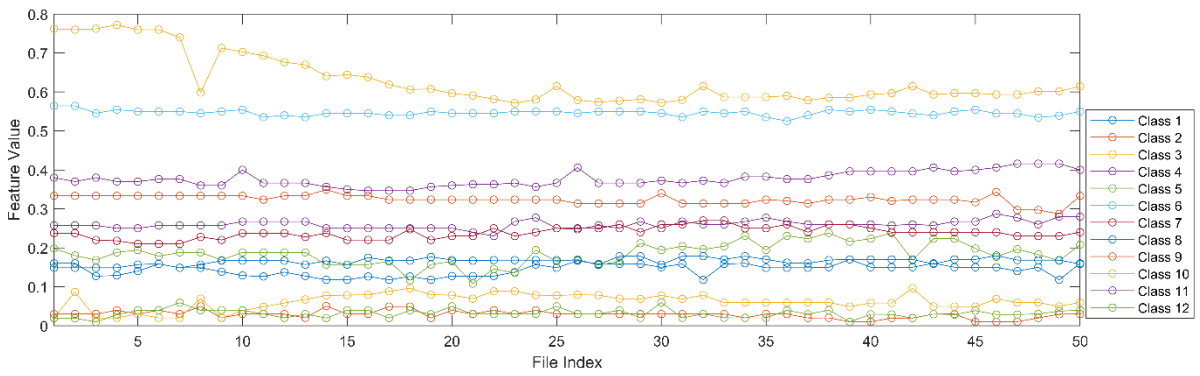


Fig. 5.21. Plot for sf_3

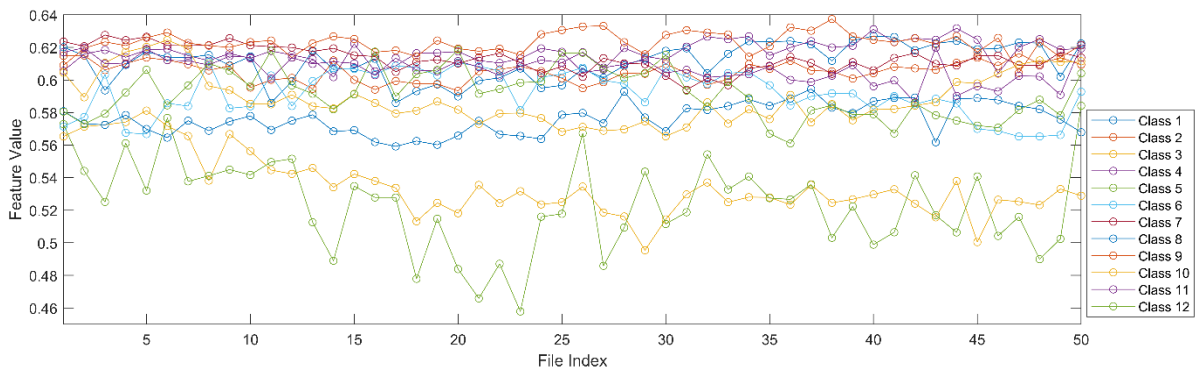


Fig. 5.22. Plot for sf_4

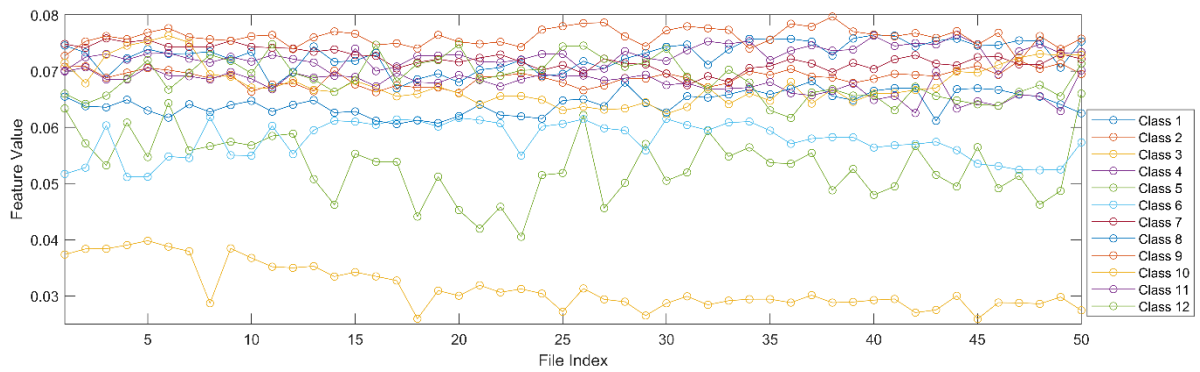


Fig. 5.23. Plot for sf_5

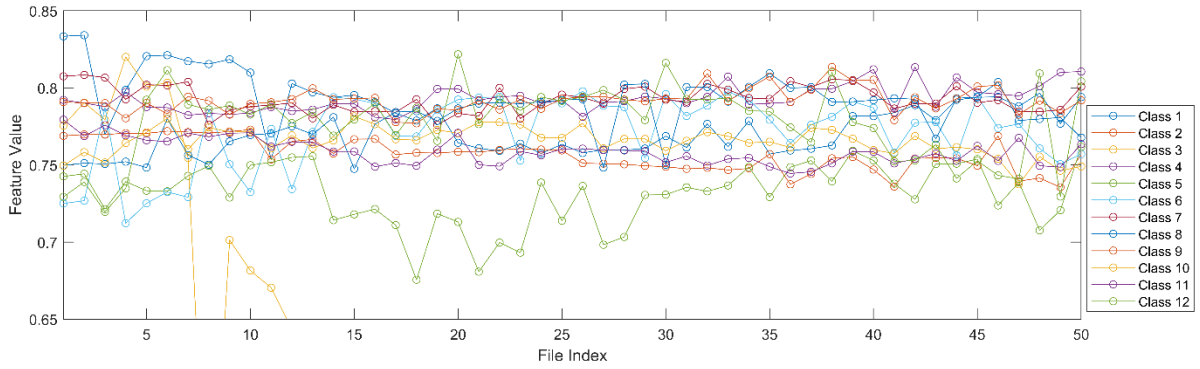


Fig. 5.24. Plot for sf_6

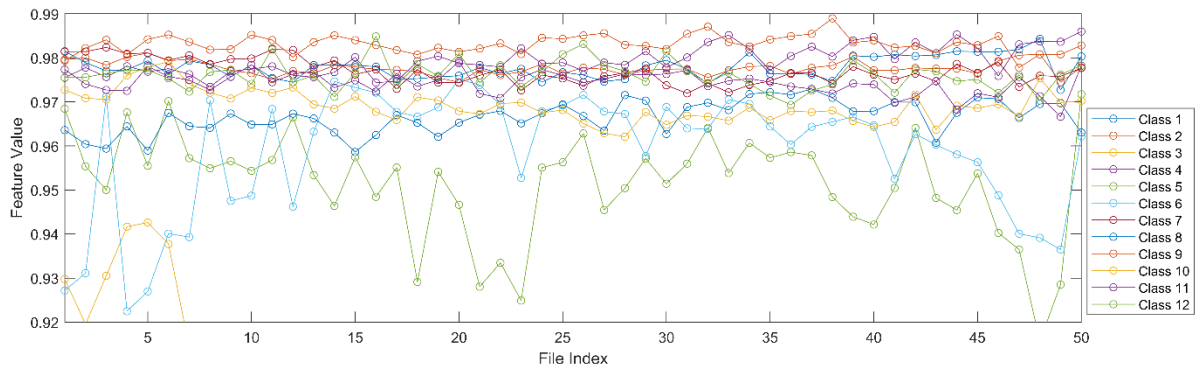


Fig. 5.25. Plot for sf_7

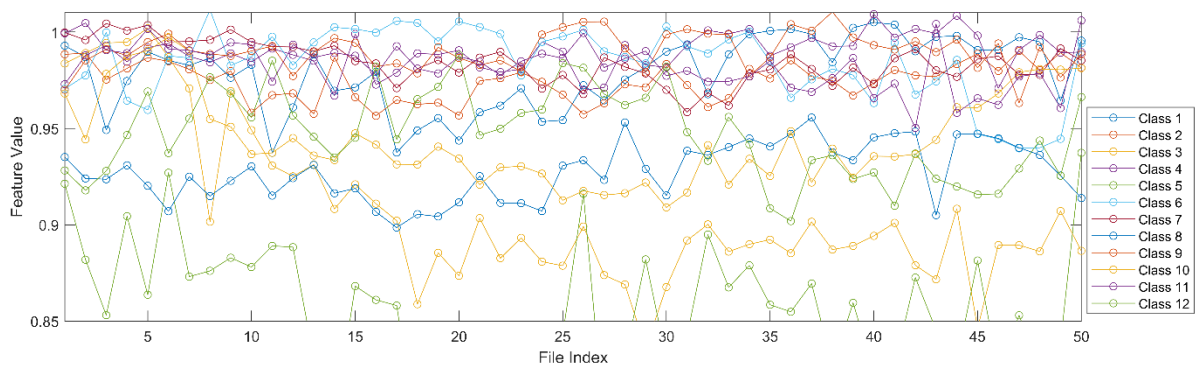


Fig. 5.26. Plot for sf_8

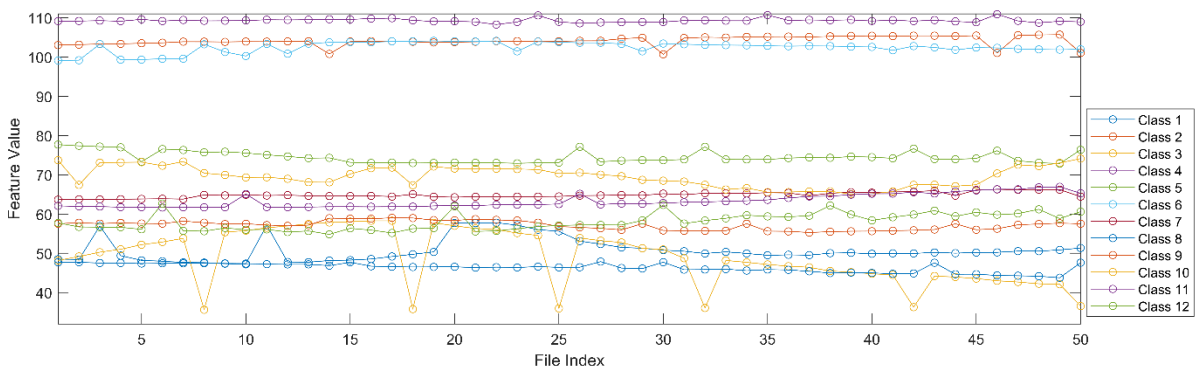


Fig. 5.27. Plot for σ_R

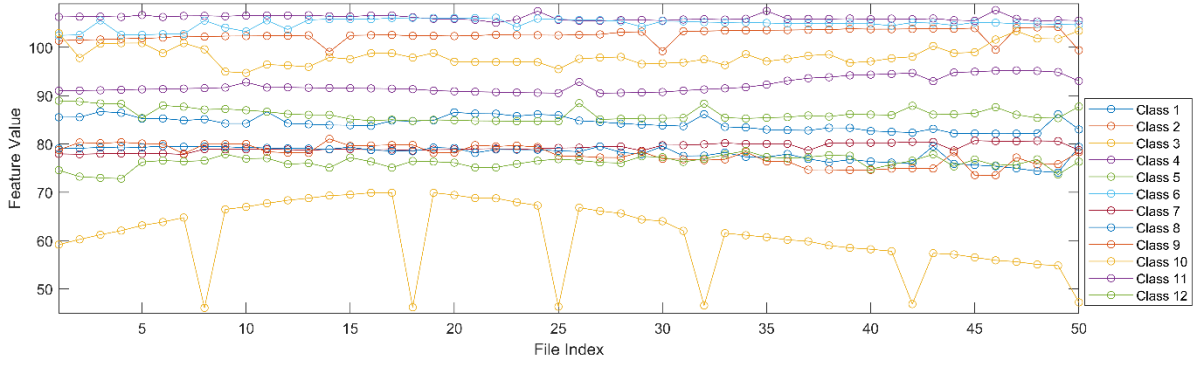


Fig. 5.28. Plot for σ_G

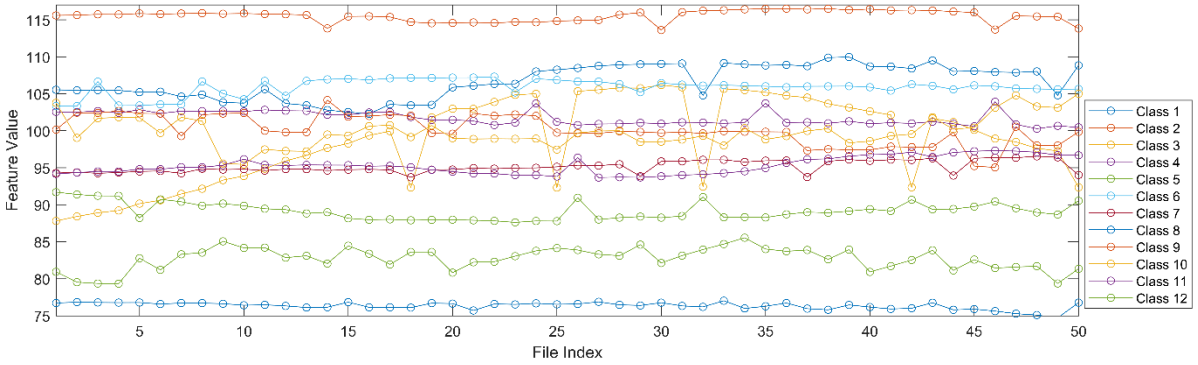


Fig. 5.29. Plot for σ_B

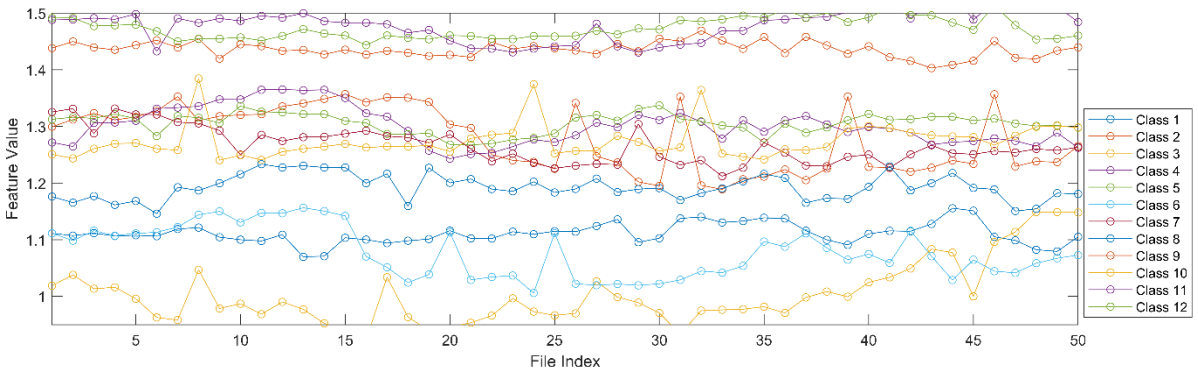


Fig. 5.30. Plot for D_1

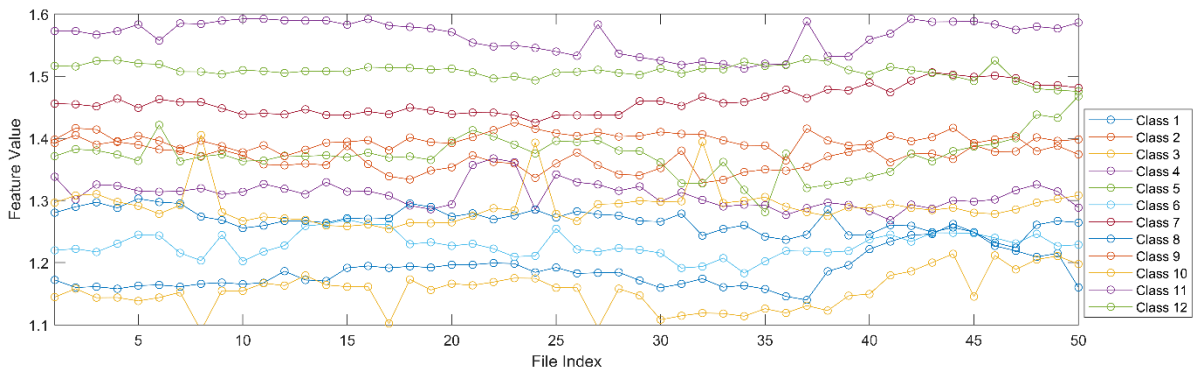


Fig. 5.31. Plot for D_2

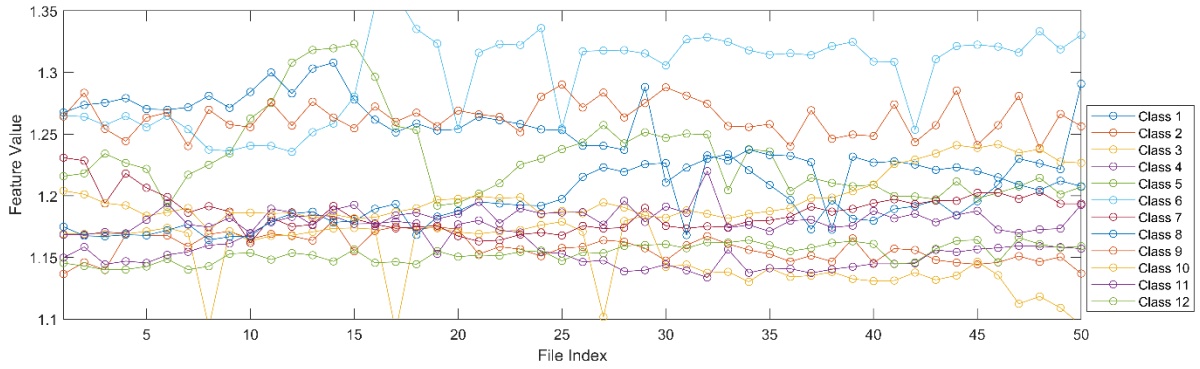


Fig. 5.32. Plot for D_3

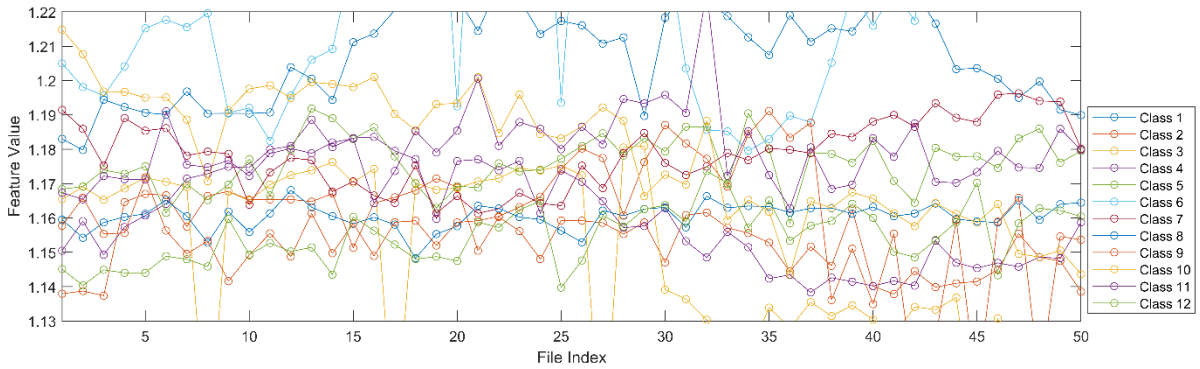


Fig. 5.33. Plot for D_4

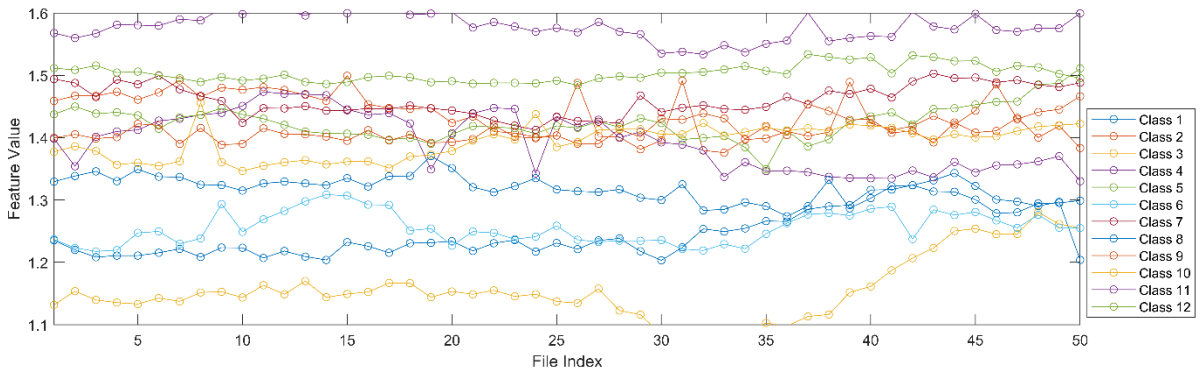


Fig. 5.34. Plot for D_5

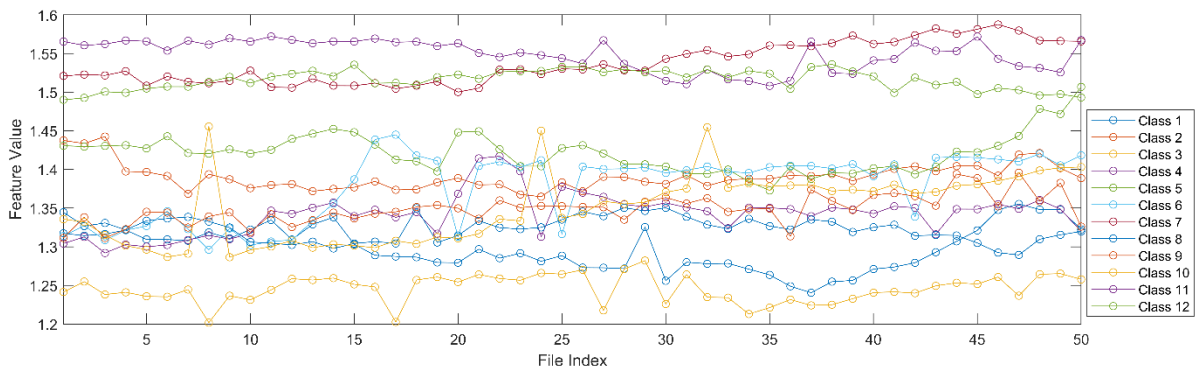


Fig. 5.35. Plot for D_6

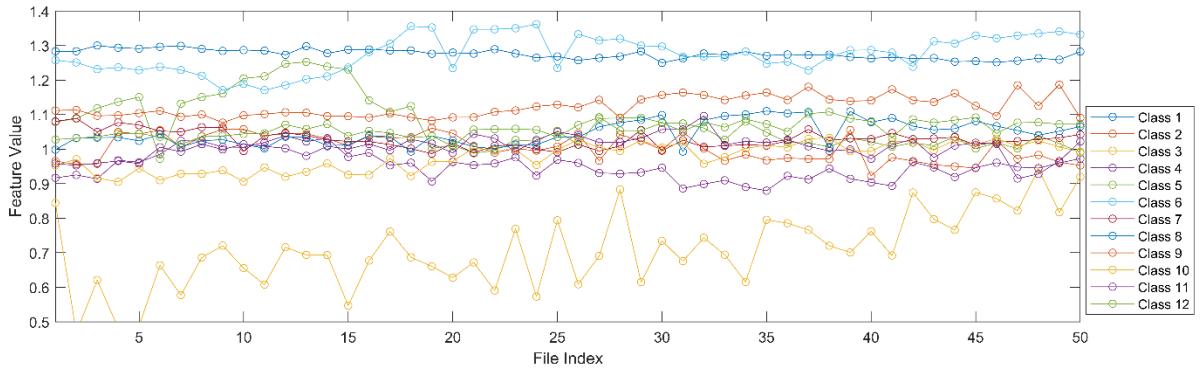


Fig. 5.36. Plot for D_7

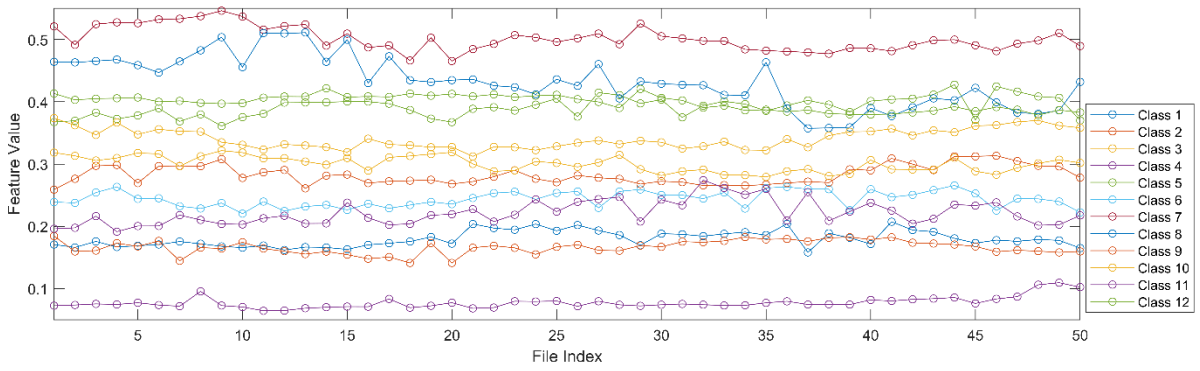


Fig. 5.37. Plot for Ct_{0°

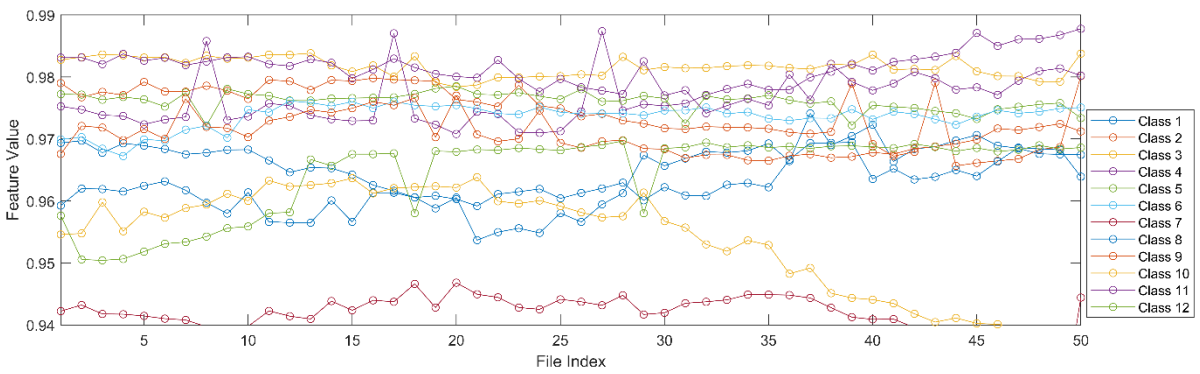


Fig. 5.38. Plot for Cn_{0°

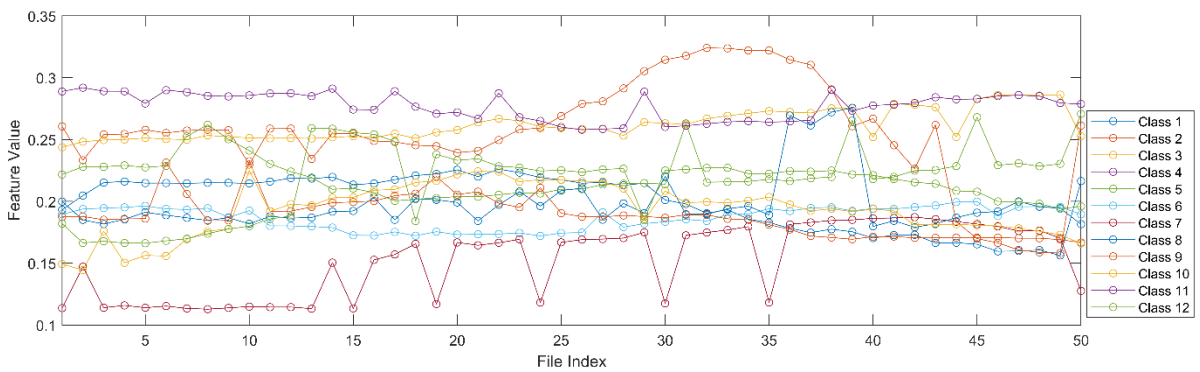


Fig. 5.39. Plot for Ey_{0°

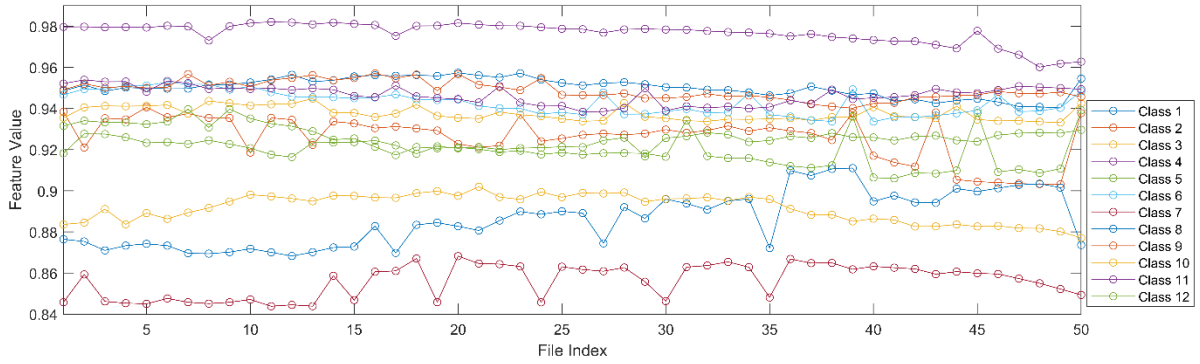


Fig. 5.40. Plot for $H\gamma_0^\circ$

After visualizing the feature plots on training samples, it can be said that the selected features are highly discriminating among different classes of fruits and vegetables. The parameters and configuration of classifiers are adjusted by the optimum results of the experimentations. The number of threshold levels is 4 for fractal dimension calculation. The predictor data has been standardized by the mean and standard deviation. The Euclidean distance is used here by the classification models for prediction. We have experimented with some other well-established classification algorithms in the proposed framework for a comparative study of the result. The classification algorithms are Naïve Bayes, Support Vector Machine (SVM), Discriminant Analysis, and Decision Tree. Different flows of the proposed approach have been tested separately. Fig. 5.41 shows overall accuracy using different classifiers on different flows of the proposed framework or with a different combination of features. The x -axis represents different classification algorithms and the y -axis represents overall classification accuracy. The figure also shows that overall accuracy improves irrespective of the classification algorithm when the features are merged. Different color represents different types or combination of features as mentioned in the legend. The overall accuracy is not fluctuating much with the change of classification algorithm in the complete proposed approach.

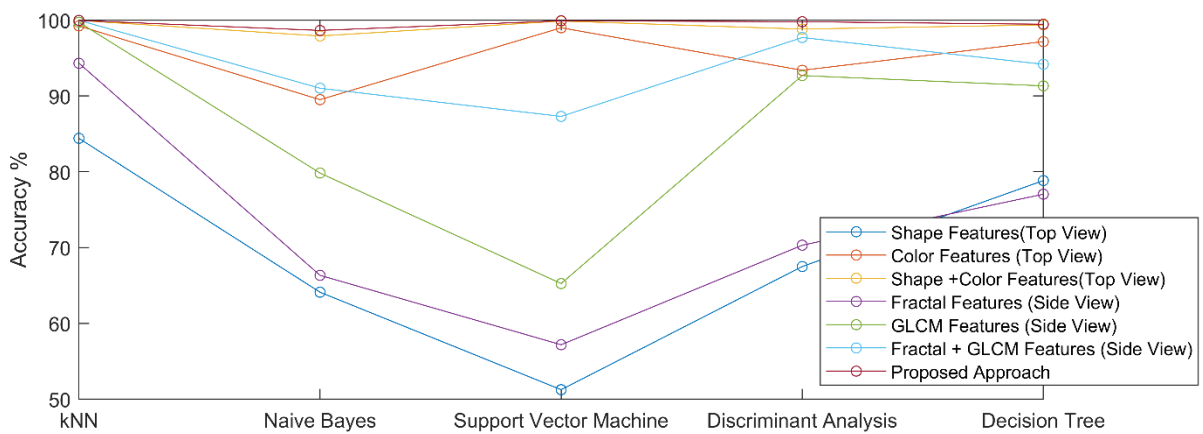


Fig. 5.41. Overall accuracy using different classifiers on different flows of the proposed framework

The unknown samples are classified separately from the top view image and side view image by the *CLM 1* and *CLM 2* respectively. Table 5.2 presents the class-wise accuracy as well as overall accuracy from the top view using different classifiers in *CLM 1*. Table 5.3 depicts the class-wise accuracy as well as overall accuracy from the side view using different classifiers in *CLM 2*. The final prediction module gives the final predicted label for an unknown test sample. Table 5.4 shows the class-wise as well as the overall accuracy of the complete proposed framework using different classifiers.

Table 5.2. Class-wise accuracy from the top view using different classifiers

Fruits and Vegetable Classes	k-NN	Naïve Bayes	SVM	Discriminant Analysis	Decision Tree
Apple Golden	100	88.20	100	94.94	100
Apple Red Delicious	100	97.19	100	100	100
Apricot	100	100	100	100	100
Avocado	100	100	100	100	100
Banana	100	100	100	100	100
Beetroot	100	100	100	100	98.00
Blueberry	100	100	100	100	100
Capsicum Green	100	100	100	100	100
Capsicum Red	100	97.26	99.32	94.52	100
Capsicum Yellow	100	98.63	100	98.63	100
Carambula	100	87.64	98.31	89.89	98.31
Cauliflower	99.38	99.38	100	100	100
Cherry	100	93.26	100	98.88	99.44
Cocos	100	100	100	93.82	100
Eggplant	100	100	100	100	100
Guava	100	97.19	100	100	100
Kiwi	100	99.31	100	100	99.31
Lemon	100	100	100	100	100
Limes	100	100	100	100	100
Lychee	100	100	100	100	100
Mango	100	100	100	100	100
Melon	99.44	94.94	100	100	95.51
Mulberry	100	100	100	100	100
Onion Red	100	100	100	100	100
Orange	100	97.52	100	100	99.38
Peach	100	92.70	100	93.82	98.88
Pear	100	100	100	100	100
Pineapple	100	100	100	100	99.44
Pomegranate	100	98.88	100	100	99.44
Potato Red	100	100	100	100	100
Potato White	100	100	100	100	96.00
Raspberry	100	100	100	100	100
Strawberry	100	100	100	100	100
Tomato Maroon	100	96.36	96.36	100	96.36
Tomato Red	100	96.89	100	100	94.41
Overall	99.96	97.91	99.88	98.82	99.38

Table 5.3. Class-wise accuracy from the side view using different classifiers

Fruits and Vegetable Classes	k-NN	Naïve Bayes	SVM	Discriminant Analysis	Decision Tree
Apple Golden	100	93.26	97.19	100	98.31
Apple Red Delicious	100	68.54	69.10	92.70	88.20
Apricot	100	96.63	82.02	100	95.51
Avocado	100	84.78	85.87	100	97.83
Banana	100	94.00	98.00	84.00	100
Beetroot	100	68.00	76.00	90.00	88.00
Blueberry	100	88.61	86.08	97.47	98.10
Capsicum Green	100	81.51	87.67	98.63	95.89
Capsicum Red	100	75.34	80.14	96.58	92.47
Capsicum Yellow	100	97.26	72.60	97.95	91.78
Carambula	100	92.70	85.39	98.31	91.01
Cauliflower	100	100	96.30	100	100
Cherry	100	86.52	86.52	99.44	86.52
Cocos	100	80.34	74.72	97.75	87.64
Eggplant	100	82.72	60.49	84.57	95.06
Guava	100	98.88	85.96	98.31	97.75
Kiwi	100	99.31	95.14	100	96.53
Lemon	100	100	98.31	100	100
Limes	100	97.06	90.59	100	96.47
Lychee	100	99.44	100	100	98.31
Mango	100	88.76	88.76	100	93.26
Melon	100	94.38	88.20	89.89	95.51
Mulberry	99.44	93.26	94.38	99.44	96.63
Onion Red	100	36.00	70.00	84.00	88.00
Orange	100	99.38	97.52	100	92.55
Peach	100	99.44	92.70	98.88	95.51
Pear	100	88.76	74.72	94.94	84.27
Pineapple	100	98.88	92.13	100	93.82
Pomegranate	100	98.31	93.82	98.31	89.89
Potato Red	100	88.00	88.00	100	94.00
Potato White	100	82.00	78.00	98.00	96.00
Raspberry	100	100	99.44	100	98.31
Strawberry	100	82.02	80.90	100	94.94
Tomato Maroon	100	96.36	96.36	100	96.36
Tomato Red	100	92.55	96.89	100	92.55
Overall	99.98	91.02	87.30	97.71	94.18

Table 5.4. Class-wise accuracy using different classifiers in the complete proposed framework

Fruits and Vegetable Classes	k-NN	Naïve Bayes	SVM	Discriminant Analysis	Decision Tree
Apple Golden	100	96.07	100	100	100
Apple Red Delicious	100	91.57	100	100	100
Apricot	100	100	100	100	100
Avocado	100	100	100	100	100
Banana	100	100	100	100	100
Beetroot	100	100	100	100	98.00
Blueberry	100	100	100	100	100
Capsicum Green	100	100	100	100	100

Capsicum Red	100	96.58	100	99.32	100
Capsicum Yellow	100	100	100	99.32	100
Carambula	100	99.44	99.44	98.31	98.31
Cauliflower	99.38	99.38	100	100	100
Cherry	100	89.33	100	100	96.63
Cocos	100	99.44	100	99.44	100
Eggplant	100	100	100	100	100
Guava	100	100	100	100	100
Kiwi	100	100	100	100	99.31
Lemon	100	100	100	100	100
Limes	100	100	100	100	100
Lychee	100	100	100	100	100
Mango	100	100	100	100	100
Melon	99.44	94.94	100	98.31	95.51
Mulberry	100	100	100	100	100
Onion Red	100	100	100	100	100
Orange	100	100	100	100	99.38
Peach	100	98.31	99.44	98.88	100
Pear	100	100	100	100	100
Pineapple	100	100	100	100	99.44
Pomegranate	100	100	100	100	99.44
Potato Red	100	100	100	100	100
Potato White	100	96.00	100	100	96.00
Raspberry	100	100	100	100	100
Strawberry	100	100	100	100	100
Tomato Maroon	100	100	96.36	100	92.73
Tomato Red	100	95.65	100	100	100
Overall	99.96	98.64	99.92	99.79	99.46

5.5 10-fold Cross-validation

Cross-validation is a process of validating the stability of classification performance on a dataset. Cross-validation is used to check some problems, which may occur during the training of the classification model i.e. biased selection or overfitting. It can be done by splitting the dataset randomly into multiple groups or folds. Select only one fold at a time as a test set and the remaining folds are used for training of the classification model. Here, the dataset has been split into 10 folds. Table 5.5 shows the class-wise distribution of samples over the folds. Table 5.6 shows the class-wise accuracy of different folds using the proposed approach. The minimum and maximum accuracy of 10-fold cross-validation is 99.90% and 100% respectively. Hence, it can be said that the performance of the proposed framework is stable.

Table 5.5. Distribution of images over multiple folds

Class Labels	Different Folds									
	1	2	3	4	5	6	7	8	9	10
Apple Golden	33	33	32	33	33	33	33	32	33	33
Apple Red Delicious	33	33	32	33	33	33	33	32	33	33

Apricot	33	33	32	33	33	33	33	32	33	33
Avocado	24	24	25	24	24	24	24	25	24	24
Banana	20	20	20	20	20	20	20	20	20	20
Beetroot	20	20	20	20	20	20	20	20	20	20
Blueberry	31	31	30	31	31	31	31	30	31	31
Capsicum Green	30	29	30	29	30	30	29	30	29	30
Capsicum Red	30	29	30	29	30	30	29	30	29	30
Capsicum Yellow	30	29	30	29	30	30	29	30	29	30
Carambula	33	33	32	33	33	33	33	32	33	33
Cauliflower	31	31	32	31	31	31	31	32	31	31
Cherry	33	33	32	33	33	33	33	32	33	33
Cocos	33	33	32	33	33	33	33	32	33	33
Eggplant	31	31	32	31	31	31	31	32	31	31
Guava	33	33	32	33	33	33	33	32	33	33
Kiwi	29	30	29	30	29	29	30	29	30	29
Lemon	33	33	32	33	33	33	33	32	33	33
Limes	32	32	32	32	32	32	32	32	32	32
Lychee	33	33	32	33	33	33	33	32	33	33
Mango	33	33	32	33	33	33	33	32	33	33
Melon	33	33	32	33	33	33	33	32	33	33
Mulberry	33	33	32	33	33	33	33	32	33	33
Onion Red	20	20	20	20	20	20	20	20	20	20
Orange	31	31	31	31	32	31	31	31	31	31
Peach	33	33	32	33	33	33	33	32	33	33
Pear	33	33	32	33	33	33	33	32	33	33
Pineapple	33	33	32	33	33	33	33	32	33	33
Pomegranate	33	33	32	33	33	33	33	32	33	33
Potato Red	20	20	20	20	20	20	20	20	20	20
Potato White	20	20	20	20	20	20	20	20	20	20
Raspberry	33	33	32	33	33	33	33	32	33	33
Strawberry	33	33	32	33	33	33	33	32	33	33
Tomato Maroon	21	20	21	20	21	20	21	20	21	20
Tomato Red	31	31	31	31	32	31	31	31	31	31

Table 5.6. Class-wise final accuracy on different folds using the proposed approach

Class Labels	Different Folds									
	1	2	3	4	5	6	7	8	9	10
Apple Golden	100	100	100	100	100	100	100	100	100	100
Apple Red Delicious	100	100	100	100	100	100	100	100	100	100
Apricot	100	100	100	100	100	100	100	100	100	100
Avocado	100	100	100	100	100	100	100	100	100	100
Banana	100	100	100	100	100	100	100	100	100	100
Beetroot	100	100	100	100	100	100	100	100	100	100
Blueberry	100	100	100	100	100	100	100	100	100	100
Capsicum Green	100	100	100	100	100	100	100	100	100	100
Capsicum Red	100	100	100	100	100	100	100	100	100	100
Capsicum Yellow	100	100	100	100	96.67	100	100	100	100	100
Carambula	100	100	100	96.97	100	96.97	100	100	100	100
Cauliflower	100	100	96.88	100	100	100	100	100	100	100
Cherry	100	100	100	100	100	100	100	100	100	100
Cocos	100	100	100	100	100	100	100	100	100	100
Eggplant	100	100	100	100	100	100	100	100	100	100

Guava	100	100	100	100	100	100	100	100	100	100
Kiwi	100	100	100	100	100	100	100	100	100	100
Lemon	100	100	100	100	100	100	100	100	100	100
Limes	100	100	100	100	100	100	100	100	100	100
Lychee	100	100	100	100	100	100	100	100	100	100
Mango	100	100	100	100	100	100	100	100	100	100
Melon	100	100	100	100	100	100	100	100	100	100
Mulberry	100	100	100	100	100	100	100	100	100	100
Onion Red	100	100	100	100	100	100	100	100	100	100
Orange	100	100	100	100	100	100	100	100	100	100
Peach	100	100	100	100	100	100	100	100	100	100
Pear	100	100	100	100	100	100	100	100	100	100
Pineapple	100	100	100	100	100	100	100	100	100	100
Pomegranate	100	100	100	100	100	100	100	100	100	100
Potato Red	100	100	100	100	100	100	100	100	100	100
Potato White	100	100	100	100	100	100	100	100	100	100
Raspberry	100	100	100	100	100	100	100	100	100	100
Strawberry	100	100	100	100	100	100	100	100	100	100
Tomato Maroon	100	100	100	100	100	100	100	100	100	100
Tomato Red	100	100	100	100	100	100	100	100	100	100
Overall	100	100	99.90	99.90	99.90	99.90	100	100	100	100

5.6 Analysis

The experimentation shows that the classification framework is very accurate and stable. The maximum and minimum accuracy of the proposed framework is 99.96% and 98.64% respectively. The classification accuracy using the kNN classifier is the maximum for the complete proposed framework as well as for both the views separately. The overall accuracy using Naïve Bayes, SVM, Discriminant Analysis, and Decision Tree is 98.64%, 99.92%, 99.79%, and 99.46% respectively. It shows that the performance of other classifiers in this framework is also good. It is observed that the overall classification accuracy from the top view is slightly better than the side view for most of the classifiers. The reason could be the merging of two types of features i.e. shape and color for classification from the top view whereas the side view uses only texture features for the classification. It is also observed that the prediction accuracy improves after passing through the final prediction module.

Some of the previous approaches have been implemented here to compare the effectiveness of the proposed approach. Table 5.7 gives the previous approaches and their techniques in brief. The approach of Roomi et al. is shape-based. Cornejo et al. proposed a histogram-based approach. The approach of Ninawe et al. is a mixture of shape, color, and texture features. Table 5.8 presents the class-wise as well as overall classification accuracy using previous approaches on both the top view and side view. Fig. 5.42 and Fig. 5.43 depict the comparison

of the proposed approach and previous approaches on the top view dataset and side view dataset respectively. The x -axis presents the index of different classes and the y -axis presents the percentage of accuracy. The different color presents different approaches as mentioned in the legend. The red line depicts that the classification accuracy has been improved using the proposed approach for all over the classes than the previous approaches on both the views.

Table 5.7. The details of the previous approaches implemented for comparison with the proposed approach

Sl. No.	Approach	Number of Classes	Features	Classifier
1	Roomi et al., 2012	3	Circulatory ratio, major axis and minor axis ratio, and eccentricity	Naïve Bayes
2	Ninawe et al. Approach, 2014	6	Perimeter, area, roundness, mean value of RGB channels, and grayscale entropy	kNN
3	Cornejo et al., 2016	15	Histogram of hue, saturation channel, and histogram of census transformed grayscale image	SVM

Table 5.8. Class-wise prediction accuracy using separate camera view and previous approaches

Class Labels	Roomi et al. Approach		Ninawe et al. Approach		Cornejo et al. Approach	
	Top View	Side View	Top View	Side View	Top View	Side View
Apple Golden	93.26	85.96	92.13	82.02	100	100
Apple Red Delicious	23.60	0	95.51	77.53	100	76.97
Apricot	15.17	69.10	98.88	85.96	100	89.33
Avocado	95.65	1.09	100	71.74	100	69.57
Banana	100	88	100	96	0	90
Beetroot	50	0	100	56	100	4
Blueberry	94.30	0	99.37	94.94	100	100
Capsicum Green	0.68	0.68	100	84.93	100	94.52
Capsicum Red	21.92	32.88	93.84	95.21	100	91.78
Capsicum Yellow	0	8.22	98.63	98.63	100	100
Carambula	53.37	13.48	80.34	86.52	7.87	89.89
Cauliflower	87.04	56.79	86.42	88.27	100	100
Cherry	43.82	53.93	95.51	89.33	100	100
Cocos	88.20	4.49	96.63	61.24	100	83.15
Eggplant	100	56.17	100	68.52	82.72	21.60
Guava	67.98	3.93	99.44	91.57	100	97.75
Kiwi	99.31	0	98.61	84.72	100	65.97
Lemon	41.01	0	85.39	94.38	100	100
Limes	5.88	54.12	77.06	87.65	100	100
Lychee	86.52	22.47	100	99.44	100	89.89
Mango	24.72	32.02	100	85.39	100	90.45
Melon	6.18	0	87.64	51.12	100	47.19
Mulberry	54.49	21.91	100	85.39	100	94.38
Onion Red	98	44	96	60	72	12
Orange	75.78	85.09	100	98.76	100	100

Peach	1.12	0	97.19	88.20	96.63	32.58
Pear	53.37	0	95.51	61.80	88.20	28.65
Pineapple	73.60	42.70	100	55.06	100	78.65
Pomegranate	58.43	1.12	87.64	88.76	97.19	44.94
Potato Red	90	58	100	88	100	64
Potato White	30	48	92	64	80	58
Raspberry	5.06	52.81	100	99.44	100	100
Strawberry	57.30	67.42	100	96.07	41.01	67.42
Tomato Maroon	50.91	16.36	96.36	98.18	100	100
Tomato Red	18.63	77.64	98.76	98.14	100	100
Overall	50.38	30.37	95.31	84.04	92.19	79.60

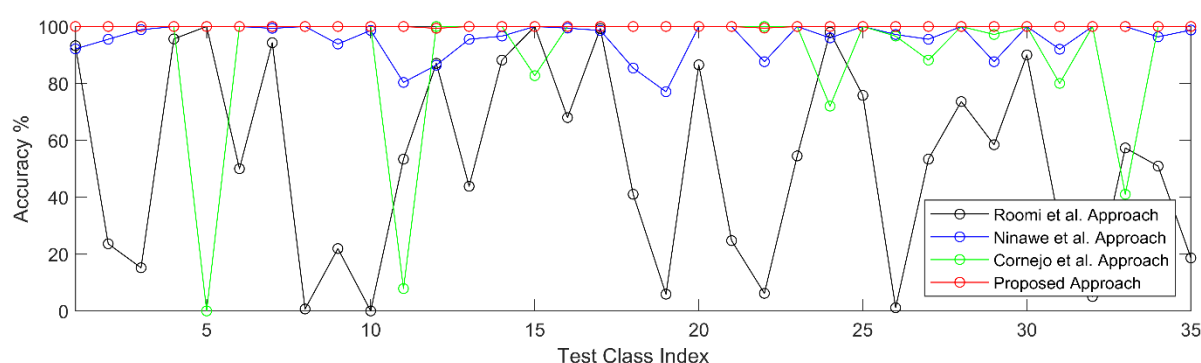


Fig. 5.42. Class-wise prediction accuracy comparison among previous approaches on top view images and the proposed approach

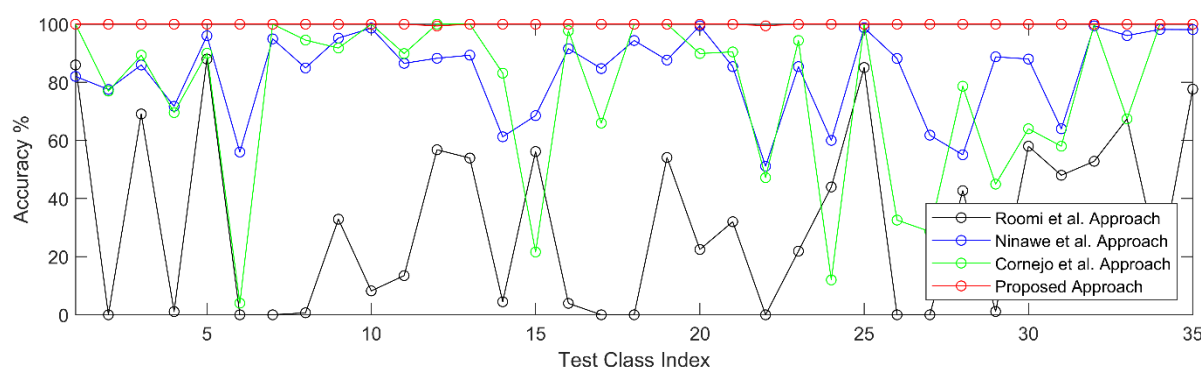


Fig. 5.43. Class-wise prediction accuracy comparison among previous approaches on the side view images and the proposed approach

5.7 Summary

The classification for all the types of fruits and vegetables, which are available in the world, is not easy using a single framework. This chapter proposes a framework for the classification of fruits and vegetables with their subtypes using the visual features (i.e. shape, color, and texture) and supervised machine learning techniques. The prior approaches mostly focus on a specific type of features for the classification of a small number of fruits and vegetable classes. It has been observed that the performance with a specific type of feature is very poor for a large number of classes. This framework will reliably classify 35 types of fruits and vegetables by

combining three types of features. The top view and side view of the fruits and vegetable surface are considered for predicting the class label of an unknown sample. The proposed framework has been divided into two-phase i.e. training and testing.

In the training phase, the imaging module captures the images from the top view and side view and sends them to the image repository. The features extraction module takes images from the repository and extracts shape, color, and texture features. The extracted features are stored in the feature database. The classification module contains classification models that are trained with the feature matrix from the feature database and class labels.

In the testing phase, the imaging module captures images of an unknown test sample and sends them to the feature extraction module for feature extraction. The extracted features are sent to the trained classification module as a query. The classification module does its prediction. The predicted class labels and corresponding scores are sent to the final prediction module. The intelligent decision-making system in the final prediction module does the final prediction of class labels for the unknown test sample. The major contributions of this chapter are-

- A complete framework is proposed here for automated fruits and vegetable classification in industries.
- The proposed framework will be able to predict the class label though it has been oriented arbitrarily at the time of passing through the conveyer belt.
- The specific features are identified for the particular view of fruits and vegetables.
- The extended shape features are independent of geometrical transformation, and different growth stages.
- The intelligent decision-making module takes the final decision of prediction by analyzing the top view as well as the side view of the image.

The limitation of this approach is that it will not be able to predict the type of fruit or vegetable which do not belong to the set of training classes.

6

Volume and Mass Estimation

6.1 Introduction

Grading [Nandi et al., 2016] of fruits and vegetables is very important for the marketing of agricultural products. The fruits and vegetables should be separated based on grade before placing them in the store for the customer. There are many parameters for the grading of fruits and vegetables. The parameters are shape, size, color, volume, and mass. The visual feature i.e. shape [Iqbal et al., 2015], size [Dang et al., 2010; Hu et al., 2015], and color [Lee et al., 2010] based grading has been explored a lot. The volume and mass are also important parameters for the packing of fruits and vegetables as well as for other edible products [Islamadina et al., 2018]. There is some conventional volume estimation technique for fruits and vegetables. These are mathematical modeling [Bozokalfa et al., 2010] and the water displacement method [Calbo et al., 1995]. Mathematical formulations are complex for irregular-shaped fruit and vegetable. The water displacement method may damage fruit or vegetables. It is also time-consuming for larger numbers of fruits and vegetables in production. Mass estimation is still done using the balance in the open market as well as in supermarkets. Those methods are destructive for the fruits and vegetables. It reduces the freshness and may cause a surface defect. An automated non-destructive approach could be suitable in this situation to measure volume and mass. Computer vision provides automated and non-destructive solutions in many problems of agricultural fields as well as industries based on agricultural products [Al Ohali et

al., 2011]. The volume and mass estimation by analyzing a single view image are very challenging. The previous approaches are mostly proposed for axisymmetric and regular-shaped fruits and vegetables i.e. spherical, cylindrical, elliptical, etc. The previous approaches fail to estimate volume and mass accurately when the fruits and vegetables are non-axisymmetric and irregular in shape. This chapter proposed a split and merge technique on a single view image to address those challenges.

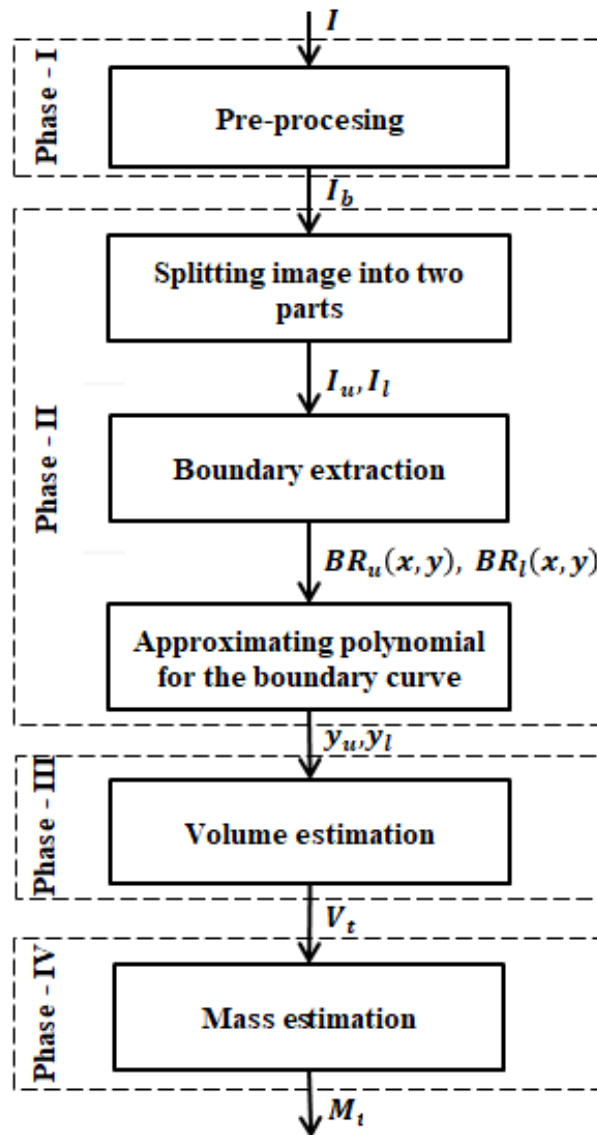


Fig. 6.1. Process flow of the proposed technique

6.2 Proposed Method

The volume and mass estimation of fruits and vegetables is the aim of this work. The proposed approach addresses the challenges and provides an accurate result by analyzing a single view image. The process flow of the proposed technique is shown in Fig. 6.1. The technique has

been structured in four phrases. In the first phase, the fruits or vegetable image (I) goes through some pre-processing techniques. Pre-processing is mandatory for segmenting the object from the background and removing noise. In the second phase, the pre-processed image (I_b) is rotated to align the major axis in the horizontal plane. Then the image has been split into two parts along the major axis. Two new images have been formed after the split i.e. the upper part (I_u) and the lower part (I_l). The coordinate of the boundary pixels is extracted for both parts. Then those coordinates are converted into real world distance units. A polynomial equation is approximated from the boundary points for each part. In the next phase, the radius is calculated from the approximated polynomial. The volume is calculated by integrating the cross-sectional area for each part along the major axis. The volumes of both parts are summed to estimate the complete volume (V_t). In the last phase, the mass (M_t) is estimated from the density-mass-volume relationship.

6.2.1 Pre-processing

The input image may contain noise or complex background. Some pre-processing steps are applied to segment the object correctly from the background as well as remove noise. RGB input image (I), Fig. 6.2(a), contains three color channels i.e. IR , IG , and IB . It is converted into a grayscale image (I_g), Fig. 6.2(b), using Eq. (6.1). The global threshold value (T) is calculated using the Otsu thresholding [Otsu, 1979] technique. The value of T is selected as optimum which has the maximum between-class variance and minimum within-class variance. The grayscale image is thresholded using Eq. (6.2) to generate the segmented fruit or vegetable object, Fig. 6.2(c). The thresholded image (I_{bw}) has the fruit or vegetable object region as black and background as white. The reason is that the experimentation setup contains a background with high intensity than the object. The thresholded binary image is complemented, Fig. 6.2(d), to make the object region as white and background as black using Eq. (6.3). Hence, the black pixels become white and white pixels become black. The white region with maximum area is considered an object region or region of interest (ROI). There may have some small white regions containing very few pixels apart from ROI. The small noise regions are discarded by region area thresholding. The binary image after removing noise, Fig. 6.2(e), contains only ROI. The ROI may contain some small holes. A morphological filling [Soille, 1999; Dougherty et al., 2003] operation is applied to fill the small holes inside the ROI. Refer to Eqs. (6.4) and (6.5). B in Eq. (6.4) is the 3×3 structuring element of this filling operation. Here, k ($k = 1, 2, 3, \dots$) is the counter of iteration. The iteration with Eq. (6.4) will

stop only when $X_k = X_{k-1}$. A union operation is performed with the binary image (I_{bw}) and X_k after the final iteration to generate the final filled image (I_b), Fig. 6.2(f).

$$I_g = 0.2989 \times IR + 0.5870 \times IG + 0.1140 \times IB \quad 6.1$$

$$I_{bw}(x, y) = \begin{cases} 1 & \text{if } I_g(x, y) > T \\ 0, & \text{otherwise} \end{cases} \quad 6.2$$

$$I_{bw}(x, y) = I_{bw}(x, y)' \quad 6.3$$

$$X_k = (X_{k-1} \oplus B) \cap I_{bw}^c \quad 6.4$$

$$I_b = I_{bw} \cup X_k \quad 6.5$$

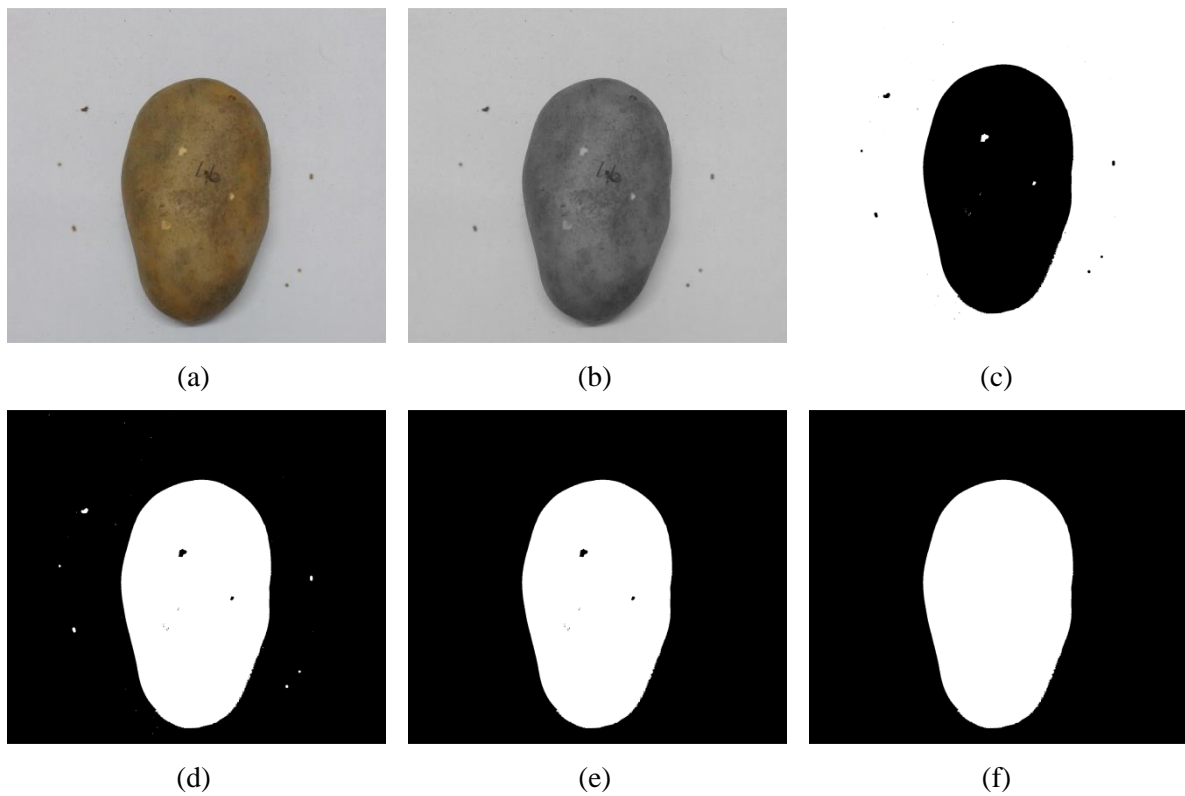


Fig. 6.2. (a) Input image (b) Grayscale version (c) Binary version (d) Complemented binary version (e) After the exclusion of noise regions (f) After filling holes

6.2.2 Splitting the image

The major axis is detected from the binary ROI in I_b . Most of the time, the major axis is not aligned accurately to the horizontal plane. The angle of inclination (θ) with the horizontal plane is measured for the major axis. The image (I_b) is rotated with that angle of inclination (θ) using

Eq. (6.6) to align the major axis to the horizontal plane. Fig. 6.3. The image (I_r) is split into two parts along the major axis. Refer to Eq. (6.7). The upper segment is flipped vertically to make the alignment similar for both parts. The segment above the major axis is marked as the upper part (I_u). Fig. 6.4(a). The segment below the major axis is marked as the lower part (I_l), Fig. 6.4(b).

$$I_r[x, y] = I_b[x, y] \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad 6.6$$

$$[I_u; I_l] = I_r \quad 6.7$$

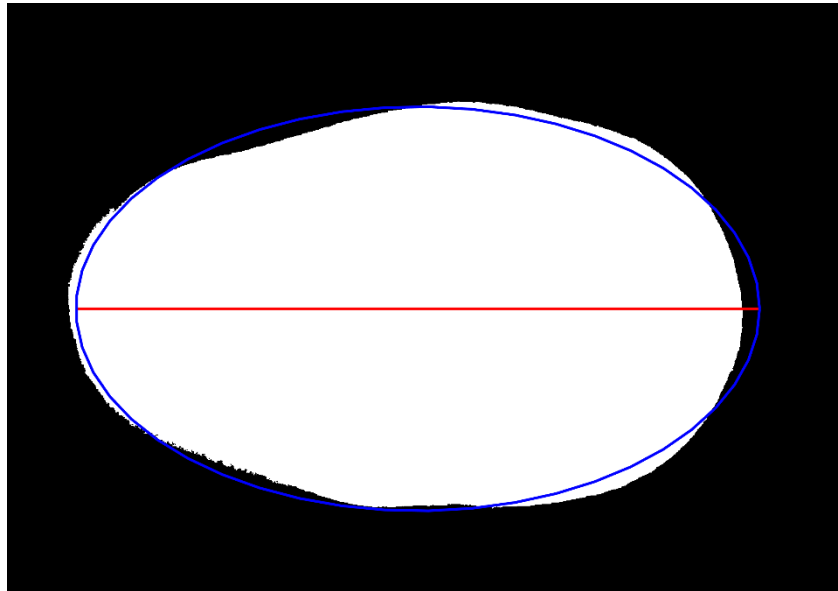


Fig. 6.3. Alignment of the major axis along the horizontal plane

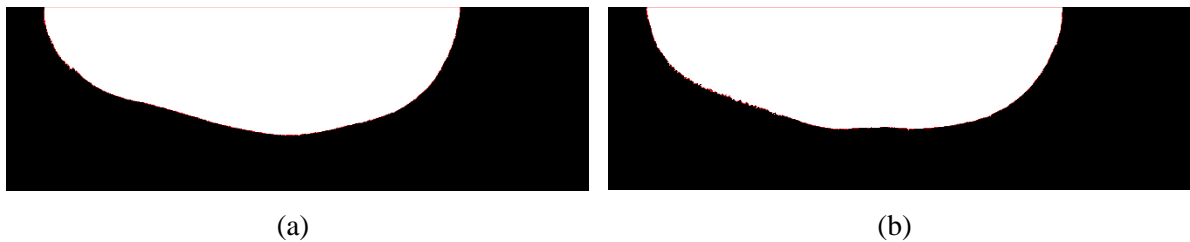


Fig. 6.4. After splitting the I_r into two parts (a) upper part (b) lower part

6.2.3 Boundary Extraction

The boundary extraction is quite challenging for irregularly shaped objects. Moore-Neighbor contour [Biswas et al., 2018] tracing algorithm, which is very popular for boundary point extraction, is used here for extracting the boundary of both the upper and lower part. The

foreground pixel value is represented by 1 and the background pixel value is represented by 0 in the binary image of fruit and vegetable object. The initial task is to find the pixel which is the starting point of the scan for a boundary. The binary image can be represented in the form of a two-dimensional matrix. The scanning will start from the bottom left corner of the image. The scan will progress from bottom to top in the vertical direction and from left to right in the horizontal direction. The pixel, which is detected as the first pixel with value 1 during scanning, is marked as the “*Start*” pixel. The first position of the boundary contour sequence is filled with the “*Start*” pixel. Now the search will be performed among the 8-neighborhood pixels of the “*Start*” pixel in the clockwise direction to find the next pixel with a value of 1. If any of the 8-neighbor pixels found with value 1, it is considered as the second pixel of the boundary contour sequence after the “*Start*” pixel. The second position of the boundary contour sequence is filled with the second pixel, which is found with value 1. Now, this second pixel is marked as the “*Top*” pixel. Again the same clockwise search is performed around the “*Top*” pixel. Assuming that the search will get a pixel with value 1. Now, this pixel is placed to the next position of the “*Top*” pixel in the boundary contour sequence and the “*Top*” marker is repositioned to this pixel. The contour search progresses in this way. The searching will end when $Top(x, y) = Start(x, y)$.

But, we have seen that this is not a very robust criterion for ending the searching process. Jacob’s stopping criteria [Seo et al., 2016] is added here to stop the contour search. The improved criterion is that the search will stop when the “*Start*” pixel is visited the second time as it was entered for the first time. As soon as the search ends successfully, it will return a set of boundary pixels $B(x, y)$ for both parts. There are some pixels in both of the parts which do not belong to the actual fruit or vegetable contour. These invalid boundary pixels are generated from major axis pixels. Hence, those invalid pixels are removed from the set $B(x, y)$ using Eq. (6.8) for both the parts of the object region. The new valid set is named $BV(x, y)$. Those valid pixels are converted into the real-world coordinate system with the help of the conversion factor (CF). Refer to Eq. (6.9). The value calculation for CF is mentioned in the *Dataset*. The $BR_u(x, y)$ and $BR_l(x, y)$ are the valid boundary points in the real coordinate system for I_u and I_l respectively. The valid boundary points for a sample are shown by the blue curve in Fig. 6.5(a) and Fig. 6.5(b) for the upper and lower part respectively.

$$BV(x, y) = \begin{cases} B(x, y) & \text{if } B_y > 0 \\ \text{discard } B(x, y), & \text{otherwise} \end{cases} \quad 6.8$$

$$BR(x, y) = BV(x, y)/CF$$

6.9

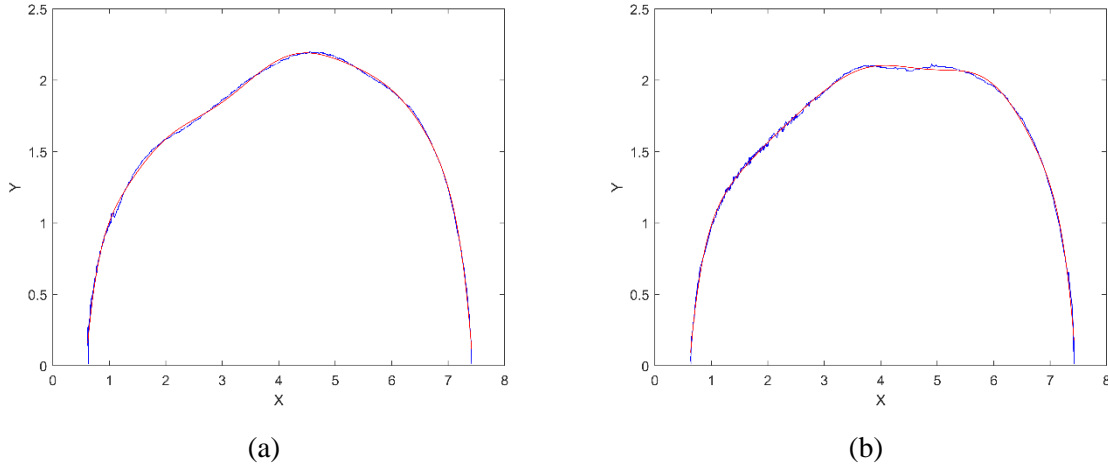


Fig. 6.5. Valid boundary points (blue) and approximated polynomial (red) for the (a) upper part and (b) lower part

6.2.4 Approximating Polynomial for Outer Boundary

The valid boundary contains m number of points in the real-world coordinate system. Refer to Eq. (6.10). Now, an n^{th} order polynomial [Ji et al., 2015; Leddy, 1997; Biswas et al., 2016] is approximated from those m number of valid boundary points in the real-world coordinate system. A Vandermonde matrix [Macon et al., 1958] $V_{m \times (n+1)}$ is formed from the x coordinates of boundary points. Refer to Eq. (6.11). A matrix, $Y_{m \times 1}$, is formed with the y coordinates of the boundary points as well. Refer to Eq. (6.12). The n^{th} order polynomial has $n+1$ number of coefficients. A matrix, $P_{(n+1) \times 1}$, is formed with the coefficients. Refer to Eq. (6.13). The coefficients will be calculated using Eq. (6.14). The set coefficients for the approximated polynomial of the upper part and lower part are $\{u_1, u_2, u_3, \dots, u_{n+1}\}$ and $\{l_1, l_2, l_3, \dots, l_{n+1}\}$ respectively. The approximated polynomial equation for the upper part and lower part are presented by Eq. (6.15) and Eq. (6.16) respectively. Those two approximated polynomials for the upper and lower part of a sample are depicted by the red curve in Fig. 6.5(a) and 6.5(b) respectively.

$$BR(x, y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \quad 6.10$$

$$V = \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & 1 \\ x_2^n & x_2^{n-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_m^n & x_m^{n-1} & \dots & 1 \end{bmatrix} \quad 6.11$$

$$Y = \begin{bmatrix} y_1 \\ y_1 \\ \vdots \\ y_m \end{bmatrix} \quad 6.12$$

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n+1} \end{bmatrix} \quad 6.13$$

$$V.P = Y \quad 6.14$$

$$y_u = u_1x^n + u_2x^{n-1} + u_3x^{n-2} + \dots + u_nx^1 + u_{n+1} \quad 6.15$$

$$y_l = l_1x^n + l_2x^{n-1} + l_3x^{n-2} + \dots + l_nx^1 + l_{n+1} \quad 6.16$$

6.2.5 Volume Estimation

It is assumed that the cross-section is nearly half-circular in shape for both the parts along the major axis. The cross-sections can be visualized as in Fig. 6.6. The radius of the cross-sections is changing arbitrarily as the fruit or vegetable is irregular in shape. The radius for each cross-section is computed by subtracting the equation of the x-axis ($y = 0$) from the approximated polynomial equation of the respective part. Refer to Eqs. (6.17) and (6.18) for the computation of cross-sectional radius of upper (R_u) and lower (R_l) part respectively. The area of a half-circle can be calculated very easily using a mathematical formula. Refer to Eqs. (6.19) and (6.20) for the cross-sectional area of upper (A_u) and lower (A_l) part respectively. The cross-sectional areas are integrated along the boundary curve to get the volume of the respective part. The equations for volume calculation of upper (V_u) and lower (V_l) part are Eqs. (6.21) and (6.22) respectively. The computed volume from both the parts are merged to get the entire volume of the fruit or vegetable object. Refer to Eq. (6.23).

$$R_u = y_u - X_{y=0} = y_u \quad 6.17$$

$$R_l = y_l - X_{y=0} = y_l \quad 6.18$$

$$A_u = \frac{\pi}{2} R_u^2 = \frac{\pi}{2} y_u^2 \quad 6.19$$

$$A_l = \frac{\pi}{2} R_l^2 = \frac{\pi}{2} y_l^2 \quad 6.20$$

$$V_u = \int_{x_1}^{x_m} A_u dx = \frac{\pi}{2} \int_{x_1}^{x_m} y_u^2 dx \quad 6.21$$

$$V_l = \int_{x_1}^{x_m} A_l dx = \frac{\pi}{2} \int_{x_1}^{x_m} y_l^2 dx \quad 6.22$$

$$V_t = V_u + V_l \quad 6.23$$

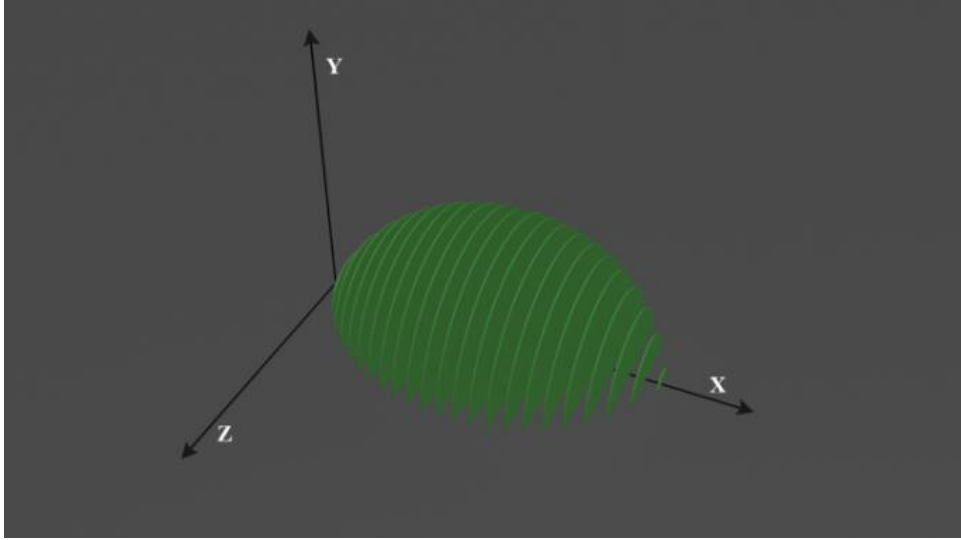


Fig. 6.6. A cross-sectional view along major-axis

6.2.6 Mass Estimation

The amount of material contained by any object is represented by Mass (M). Volume (V) represents the place occupied by the object in three-dimensional space. Mass and volume have a relation. The relationship is the measure of density (D). The density is the value of mass divided by the volume. The density of particular fruit and vegetable types is known. The volume has already been estimated in the previous section. Mass is estimated from the known density and estimated volume using Eq. (6.24).

$$M_t = D \times V_t \quad 6.24$$

6.3 Dataset

We do not find any dataset for experimenting with this approach and we do not have the scope to capture images from the conveyor belt. A simple setup was created by us to capture a single view image. The setup is created with white color background and by placing a smartphone camera with a clip and stand 20 cm above the background plane. Fig. 6.7 shows the imaging setup. The setup is placed in a closed room. The main light source is CFL slim light. The power of the light is 36W and 6500k color temperature. The samples are bought from the market.

There were 52 potatoes, 11 citruses, and 14 tomatoes. The samples are marked with a unique number. The potato samples are marked as P01, P02, ..., P52. The citrus samples are marked as C01, C02, ..., C11. The tomato samples are marked as T01, T02, ..., T14. Only a single view of fruit and vegetable objects is considered for the image. The images have been captured without a flashlight to minimize the shadow. The samples from the dataset are shown in Fig. 6.8.

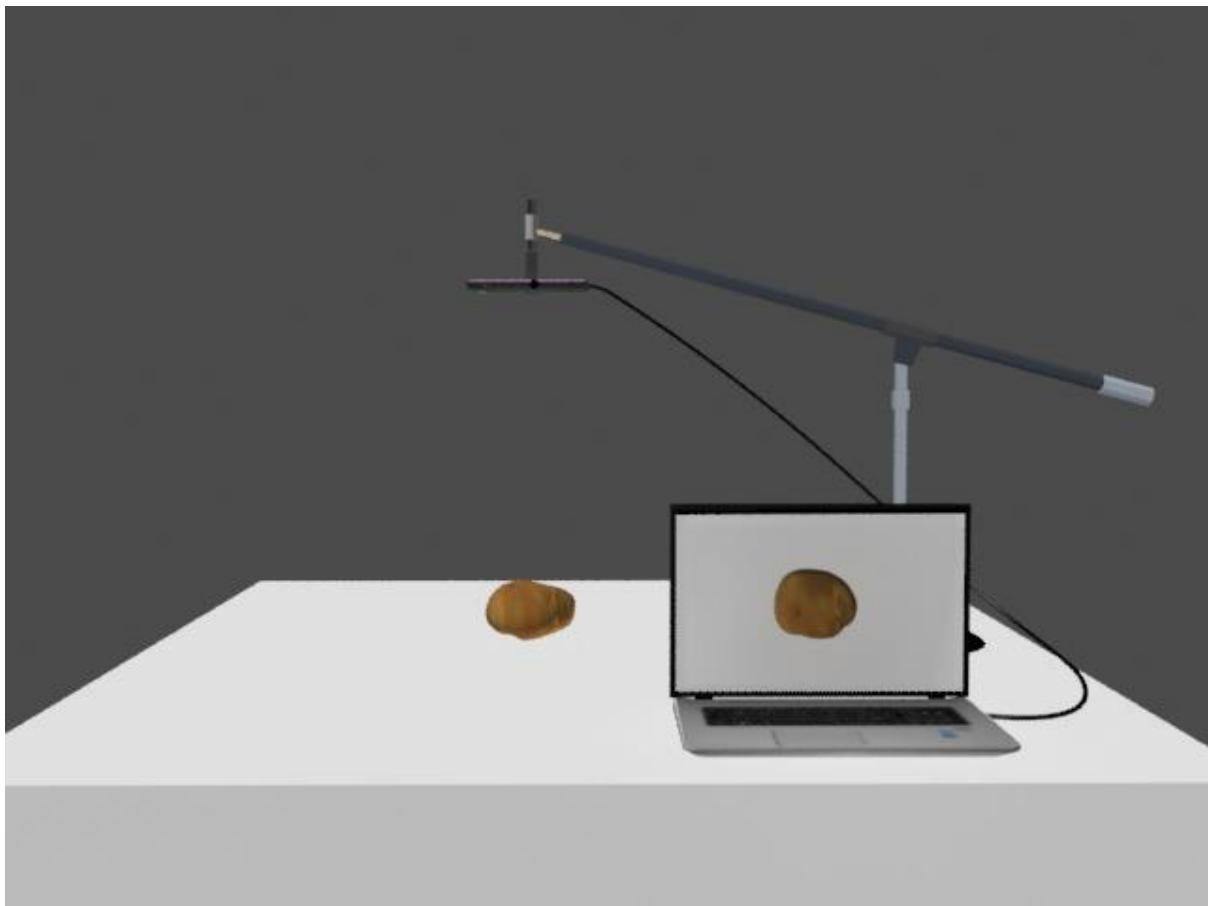


Fig. 6.7. The imaging setup



Fig. 6.8. Samples from the dataset

The actual height (major axis) and width (minor axis) are measured by a digital caliper, which is shown in Fig. 6.9(a). The volume (V) for all the samples is measured by using a standard water displacement method. The digital balance, which is shown in Fig. 6.9(b), is used to measure the mass (M) for all the samples. The distance parameters will be measured in pixels, so there should be a conversion factor from pixels to real-world units of distance. The conversion factor for each sample is computed from the major axis and minor axis measured by calipers in mm and major axis and minor axis from the image in $pixels$. The conversion will be in $pixels/mm$. The conversion factor is measured separately along the major axis and minor axis for each sample. The aggregate conversion factor (CF) was computed for each type of fruit or vegetable in $pixels/mm$ for a more accurate result. The conversion factor (CF) for potato samples along the major axis and the minor axis is $14.85\ pixels/mm$ and $14.08\ pixels/mm$ respectively. The CF for citrus samples along the major axis and the minor axis is $15.91\ pixels/mm$ and $14.54\ pixels/mm$ respectively. The CF along the major axis and minor axis for the tomato samples is $15.62\ pixels/mm$ and $14.05\ pixels/mm$ respectively.

The density (d) for each sample is computed from the measured volume using the water displacement method and mass using a digital balance. The density (d), mass (M), and volume (V) relationship ($d=M/V$) are taken from basic science. The average density (D) for each type of fruit and vegetable is also computed.



Fig. 6. 9. (a) Digital Calipers (b) Digital Balance

6.4 Experimentations & Results

The experimentation is performed on the dataset, which is created by us. The dataset contains 77 images of fruits and vegetables in total. The samples introduce shape varieties i.e. axisymmetric, non-axisymmetric, regular, and irregular, etc. The volume and mass estimation

for regular and axisymmetric fruits and vegetables has been explored a lot. Hence, this research is specially focused on non-axisymmetric and irregular-shaped fruits and vegetables. Potato is very irregular and non-axisymmetric in shape. This is the reason for incorporating a high number of potato samples in this dataset compared with other types of fruits and vegetables. The technique has been implemented in Matlab 2018a along with Windows 10 operating system, Intel Core i5, 3.00 GHz processor, and 4 GB RAM. We found that the result is optimum with the 10th order polynomial. Hence, the order of the polynomial is selected as 10. The volume (*milliliter*) is measured using the water displacement method to measure the accuracy of the estimated volume using the proposed image analysis technique. The estimated mass is validated with the measured mass (*gram*) using a digital balance. The unit of estimated volume and mass are milliliter (*ml*) and gram (*gm*) respectively.

The previous approaches in the literature survey are mostly proposed for the volume and mass estimation of fruits and vegetables, which has specific shape types i.e. spherical, elliptical, cylindrical, etc. Some of the previous approaches used the multiple-camera setup to capture views from a different angle. The multiple-camera setup gives multidimensional information about the parameters of volume. The volume and mass estimation technique, which is proposed here, is targeted for all kinds of shapes from a single view image using a single camera setup. Hence, the prior approaches will be tested on this dataset of single-view images. Venkatesh et al. [Venkatesh et al., 2014] propose an image-based volume and mass estimation for different types of shapes. Some shape-oriented parameters were measured from the boundary contour of the fruit object. They established three criteria to determine the shape of the fruit i.e. circular, parabolic, and elliptical. The extracted parameters determine the shape using those criteria. The volume was computed by the formula for that shape. The estimated volume was compared with the measured volume using the water displacement method. Mass is dependent on the estimated volume. Hence, the Venkatesh et al. approach is most suitable for comparing with the proposed approach. The Venkatesh et al. approach also developed in the same platform with the same configuration and tested on the same dataset which was used for validating the proposed approach. Table 6.1 shows the estimated volume and mass using the proposed approach as well as Venkatesh et al.'s approach along with the volume using water displacement and mass using digital balance for potato samples. Table 6.1 also shows the total execution time for a sample using the proposed approach. Tables 6.2 and 6.3 show the same entities for citrus and tomato samples respectively. The volume and mass error has been determined for both approaches by comparing with water displacement and digital balance respectively. The error

will be positive if the estimated volume or mass is more than the measured volume and mass using standard techniques. The error will be negative for the reverse scenario. Hence, an absolute percentage of error is calculated for avoiding this confusion.

Table 6.1. Volume and mass estimation results of potato samples using standard technique, proposed approach, and Venkatesh et al. approach

Samples	Volume (ml)			Mass (gm)			Execution Time using Proposed Approach (Sec)
	Water Displacement	Proposed Approach	Venkatesh et al. Approach	Digital Balance	Proposed Approach	Venkatesh et al. Approach	
P01	155	168.96	194.44	148	162.21	186.67	0.87
P02	115	122.64	127.43	113	117.74	122.33	0.90
P03	105	121.40	123.61	103	116.55	118.66	0.86
P04	115	122.95	152.78	114	118.03	146.67	0.84
P05	92	96.17	104.24	87	92.32	100.07	0.85
P06	105	108.80	144.56	104	104.45	138.77	0.87
P07	78	88.58	82.96	75	85.04	79.64	0.82
P08	90	102.81	103.09	88	98.70	98.97	0.84
P09	148	173.06	166.50	142	166.14	159.84	0.90
P10	182	192.98	195.53	172	185.26	187.71	0.92
P11	115	112.26	116.62	107	107.77	111.95	0.83
P12	140	156.56	207.85	133	150.30	199.53	0.88
P13	114	118.16	122.82	107	113.43	117.91	0.84
P14	168	200.55	276.68	164	192.53	265.61	0.89
P15	84	80.65	85.24	80	77.43	81.83	0.78
P16	113	115.53	117.50	104	110.91	112.80	0.84
P17	140	156.69	182.39	131	150.42	175.09	0.84
P18	116	117.01	161.68	95	112.33	155.22	0.81
P19	139	161.21	206.52	133	154.76	198.26	0.87
P20	77	70.09	73.81	67	67.29	70.86	0.88
P21	73	68.58	70.86	65	65.83	68.02	0.79
P22	121	128.86	132.84	116	123.71	127.53	0.84
P23	155	181.02	179.62	151	173.78	172.44	0.87
P24	170	189.34	197.63	159	181.76	189.73	0.91
P25	82	79.82	103.64	77	76.63	99.50	0.82
P26	135	149.55	156.04	132	143.57	149.80	0.88
P27	89	89.30	109.93	81	85.73	105.53	0.84
P28	92	92.89	122.50	86	89.18	117.60	0.76
P29	78	79.34	76.40	75	76.16	73.35	0.82
P30	77	82.42	112.77	74	79.13	108.26	0.83
P31	107	129.25	132.59	106	124.08	127.29	0.84
P32	100	86.24	95.20	96	82.79	91.39	0.82
P33	80	76.48	114.32	76	73.42	109.75	0.83
P34	114	120.10	126.26	112	115.30	121.21	0.88
P35	95	100.75	134.26	93	96.72	128.89	0.83
P36	107	100.12	105.69	102	96.11	101.46	0.94
P37	106	109.97	134.73	101	105.57	129.34	0.82
P38	173	194.32	191.73	173	186.55	184.06	0.91

P39	96	97.00	93.41	93	93.12	89.67	0.83
P40	103	103.04	99.43	102	98.92	95.46	0.84
P41	110	120.91	115.65	109	116.07	111.03	0.88
P42	109	120.14	124.17	109	115.33	119.20	0.84
P43	95	102.63	105.55	94	98.52	101.33	0.83
P44	79	77.07	100.43	76	73.98	96.42	0.82
P45	72	72.81	70.91	70	69.90	68.07	0.86
P46	70	66.05	105.54	67	63.40	101.32	0.81
P47	70	72.36	74.05	70	69.47	71.09	0.82
P48	74	67.55	71.76	72	64.84	68.89	0.85
P49	75	73.08	78.27	72	70.16	75.14	0.86
P50	79	82.54	84.93	76	79.24	81.54	0.81
P51	67	63.42	87.48	62	60.88	83.98	0.84
P52	72	65.12	102.34	69	62.52	98.25	0.81

Table 6.2. Volume and mass estimation results of citrus samples using standard technique, proposed approach, and Venkatesh et al. approach

Samples	Volume (<i>ml</i>)		Mass (<i>gm</i>)				Execution Time using Proposed Approach (<i>Sec</i>)
	Water Displacement	Proposed Approach	Venkatesh et al. Approach	Digital Balance	Proposed Approach	Venkatesh et al. Approach	
C01	144	130.30	136.03	133	119.88	125.15	0.88
C02	141	135.92	136.85	127	125.05	125.90	0.88
C03	128	105.06	117.25	114	96.66	107.87	0.86
C04	145	130.75	143.46	129	120.29	131.98	0.87
C05	142	124.45	131.35	130	114.49	120.84	0.87
C06	100	80.33	93.77	96	73.90	86.26	0.83
C07	87	90.30	75.75	81	83.08	69.69	1.00
C08	103	82.79	91.85	95	76.17	84.50	0.82
C09	99	102.44	86.61	92	94.25	79.68	0.84
C10	108	95.86	88.96	101	88.19	81.84	0.92
C11	97	85.38	87.56	89	78.55	80.55	0.83

Table 6.3. Volume and mass estimation results of tomato samples using standard technique, proposed approach, and Venkatesh et al. approach

Samples	Volume (<i>ml</i>)		Mass (<i>gm</i>)				Execution Time using Proposed Approach (<i>Sec</i>)
	Water Displacement	Proposed Approach	Venkatesh et al. Approach	Digital Balance	Proposed Approach	Venkatesh et al. Approach	
T01	114	105.00	107.24	100	93.45	95.45	0.86
T02	80	72.10	75.25	77	64.17	66.97	0.81
T03	73	65.12	68.70	68	57.96	61.14	0.82
T04	107	114.81	118.28	92	102.18	105.27	0.83
T05	74	66.35	67.13	66	59.05	59.75	0.81
T06	67	73.78	63.99	60	65.66	56.96	0.78
T07	46	34.44	33.92	41	30.65	30.19	0.76
T08	74	59.94	58.86	66	53.35	52.38	0.80
T09	79	67.17	79.88	68	59.78	71.09	0.80

T10	84	70.54	69.64	71	62.78	61.98	0.80
T11	98	90.53	90.37	86	80.58	80.43	0.84
T12	90	85.60	91.05	77	76.18	81.03	0.86
T13	87	87.04	88.86	81	77.47	79.08	0.83
T14	75	67.70	67.60	68	60.25	60.17	0.80

6.5 Analysis

The main motivation for doing automation is to reduce execution time compared with manual execution. The proposed approach is also an automated approach for volume and mass estimation. Hence processing time is very important in the context of this problem. The average volume and mass estimation time for a potato sample is 0.85 s using the proposed technique. The average processing time for citrus and a tomato sample is 0.87 s and 0.81 s respectively using the proposed approach. A human resource may take a few minutes to measure volume and mass for a fruits and vegetable sample whereas the automated system using the proposed approach does the same task just within a second.

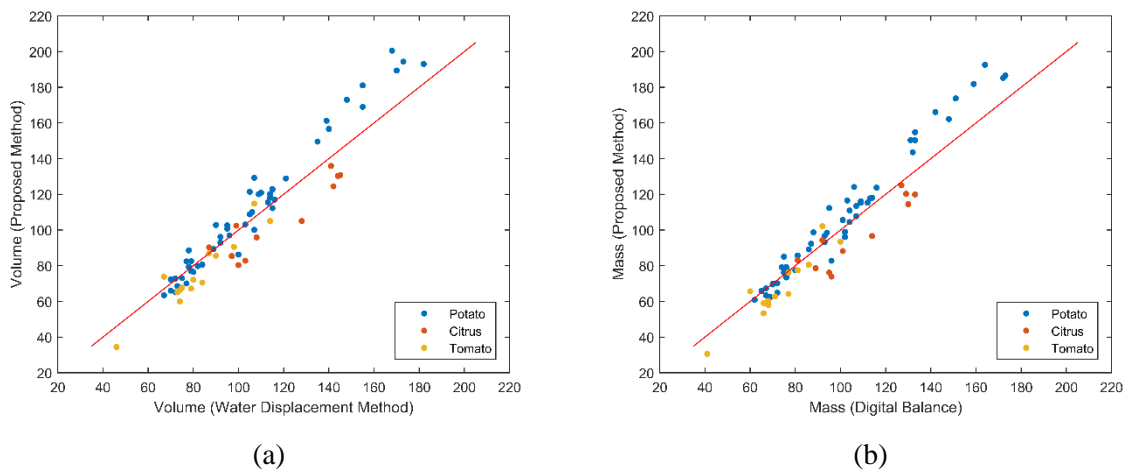


Fig. 6.10. The scatter plot between the proposed approach and standard technique for (a) Volume and (b) Mass

Fig. 6.10(a) shows the scatter plot between estimated volume using the proposed approach and measured volume using the water displacement approach for all the samples of the dataset. The colored dots represent a sample of a specific type of fruit and vegetable that has been mentioned in the legend. The significance of a more accurate volume estimation for a sample is that the dot for the corresponding sample will be closer to the red-colored line. The volume accuracy is very good for the sample with low volume. That means volume error is increasing with the increment of volume. Pearson's correlation coefficient is calculated between measured volume using the water displacement method and estimated volume using the proposed approach. The

correlation coefficient of volume estimation for potato, citrus, tomato is 0.99, 0.92, and 0.94 respectively. The overall correlation coefficient of volume estimation is 0.96 on the complete dataset.

Fig. 6.10(b) shows the scatter plot between the estimated mass using the proposed approach and the measured mass using digital balance for all the samples of the dataset. The samples are represented by a dot of a particular color for a specific type of fruit or vegetable mentioned in the legend. The estimated mass is dependent on the estimated volume. Hence, the performance of the mass deviation is almost similar to volume. Here also the Pearson's correlation coefficient is calculated between the measured mass using digital balance and estimated mass using the proposed approach. The correlation coefficient of the estimated mass for potato, citrus, and tomato is 0.99, 0.91, and 0.94 respectively. The overall correlation coefficient of mass estimation is 0.97 on the complete dataset. Here also, the percentage of error is increasing with the increment of mass. It is observed that both estimated volume and mass using the proposed approach are positively correlated with the measured volume and mass using standard approaches.

The average volume estimation error using the proposed approach is 7.46%, 11.18%, and 10.98% for potato, citrus, and tomato respectively. The average volume estimation error using Venkatesh et al.'s approach is 19.09%, 8.67%, and 9.14% for potato, citrus, and tomato respectively. The average mass estimation error using the proposed approach is 7.02%, 10.69%, and 11.44% for potato, citrus, and tomato respectively. The average mass estimation error using Venkatesh et al.'s approach is 19.60%, 8.96%, and 10.46% for potato, citrus, and tomato respectively. We can see that the result is significantly improved than the Venkatesh et al approach. Fig. 6.11 depicts a comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the potato samples. Fig. 6.12 and 6.13 depict the same for citrus and tomato samples respectively. It is visible in the plots that the volume estimation error using Venkatesh et approach is more than the volume estimation error using the proposed approach for most of the samples in the dataset. The volume estimation error is less than 10% for 71.15% of potato samples using the proposed approach whereas the volume estimation error is more than 10% for 61.54% of potato samples using the Venkatesh et al. approach. The potato is mostly irregular and non-axisymmetric in shape. The performance of the proposed approach on the potato sample is very good. Hence, it can be said that the proposed approach is working accurately and efficiently

for irregular and non-axisymmetric fruits and vegetables. The volume estimation error of citrus and tomato samples is not deviating too much between the proposed approach and the Venkatesh et al. approach. The citrus and tomato are mostly regular and axisymmetric in shape. This could be the reason for the closeness of volume estimation error of citrus and tomato samples using the proposed approach and Venkatesh et al. approach. The mass is calculated by multiplying the density with volume. Hence mass is dependent on volume. The mass estimation error will be nearly similar to the volume estimation error of the corresponding fruit and vegetable samples. It was observed that the mass estimation error is less than 10% for 75% of potato samples using the proposed approach whereas mass estimation error is more than 10% for 53.85% of potato samples using the Venkatesh et al. approach. The mass estimation error for citrus and tomato are very close using both the proposed and Venkatesh et al. approach.

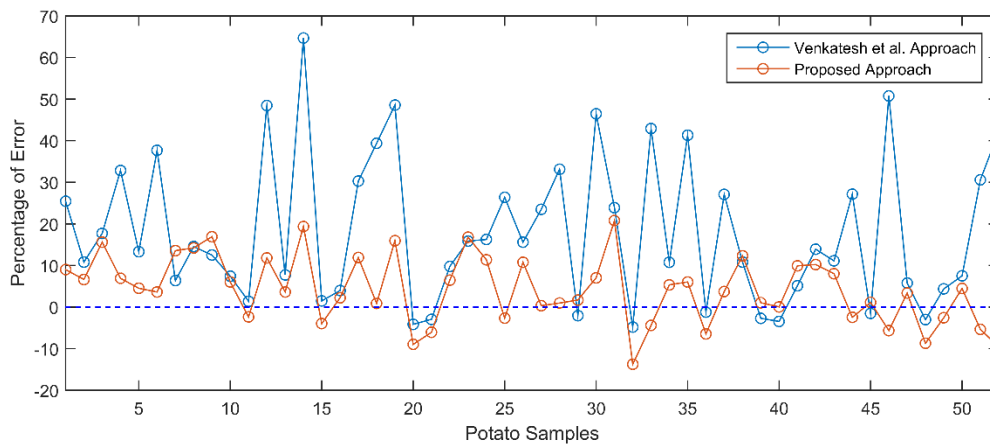


Fig. 6.11. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the potato samples

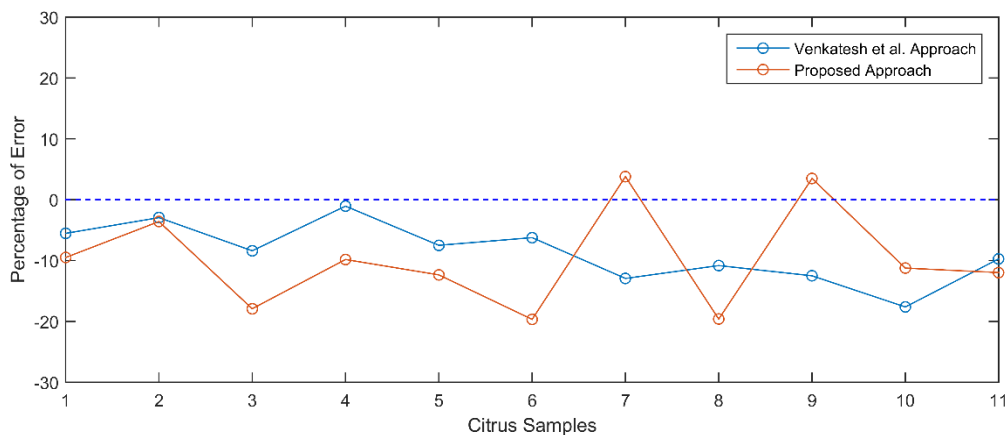


Fig. 6.12. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the citrus samples

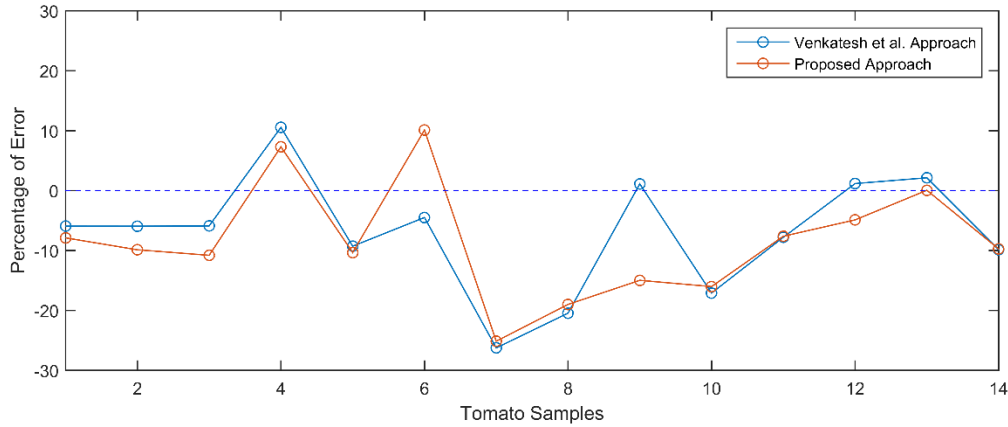


Fig. 6.13. Comparison of volume estimation error between the proposed approach and Venkatesh et al. approach with respect to the water displacement method for the tomato samples

Table 6.4. Comparison of absolute percentage of error for volume and mass estimation where the proposed approach overcomes the limitations of the Venkatesh et al. approach





Approach	Parameters				
Proposed Approach	Volume	2.66%	0.34%	2.45%	5.34%
	Mass	0.48%	5.84%	2.65%	1.80%
Venkatesh et al. Approach	Volume	26.39%	23.51%	27.13%	30.57%
	Mass	29.22%	30.28%	26.86%	35.46%

Table 6.4. shows images of a few potato samples where Venkatesh et al. approach suffers from highly erroneous volume and mass estimation but the proposed approach appears with a highly accurate estimation for the same. The reason behind this erroneous result is that those samples are highly irregular and non-axisymmetric in shape. This outcome states that the Venkatesh et al. approach struggles with irregular and non-axisymmetric fruits and vegetables whereas the proposed approach addresses the challenges with a *Split and Merge* technique.

6.6 Summary

This chapter estimates the very important grading parameters using image analysis. The proposed technique in this chapter estimates the volume and mass by analyzing an image of fruits and vegetables. The input image has been pre-processed to remove noise and segment the object properly. The image is rotated for aligning the major axis horizontally and the image is divided into two parts along the major axis. A polynomial equation is formed for each part from the boundary points of the corresponding part. The volume is calculated by integrating the polynomial along the boundary. The mass is computed from the density, mass, and volume relationship. The results are validated with the water displacement method for volume and

digital balance for mass. The accuracy of the estimated volume using the proposed approach is 92.54%, 88.82%, and 89.02% with respect to the water displacement method for potato, citrus, and tomato respectively. The accuracy of mass estimation using the proposed approach is 92.98%, 89.31%, and 88.56% with respect to digital balance for potato, citrus, and tomato respectively. The average execution time for a sample to estimate volume and mass is 0.84 s. It is satisfactory compared with manual execution time.

The major contributions of this chapter are –

- An image dataset of fruits and vegetables with variation in shape and size. The dataset contains 77 images of three types of fruits and vegetables as well as the volume using the water displacement method and mass using digital balance for each sample.
- An automated technique for volume and mass estimation of fruits and vegetables from a single view image. The technique will be equally effective, efficient, and accurate for regular and axisymmetric as well as irregular and non-axisymmetric shaped fruits and vegetables.

The limitation of this work is that the technique is not applicable for the fruits and vegetables which are available as a branch e.g. grape.

Supplier Selection

7.1 Introduction

Fruits and vegetables are a very profitable business in India. India exported fruits worth 668.75 USD Millions and vegetables worth 608.48 USD Millions in the year 2019-20. Indian fruits and vegetables are majorly exported in the countries like UK, UAE, Netherland, Oman, Qatar, Nepal, Sri Lanka, Bangladesh, etc. Now-a-day customers are very concerned about food safety. They always wish for fresh and quality fruits and vegetables at a lower price. The retailers of fruits and vegetables always want to maximize profit. This is a challenge from the managerial point of view to balance both the demand of the customer as well as the profit of the store. The appropriate supplier selection could solve the problem of fruits and vegetable stores. Another concern of fruits and vegetable stores is a large amount of wastage. The amount of fruits and vegetable wastage is nearly 30% in India [Negi et al., 2014]. Fig. 7.1 depicts the fruits and vegetables supply chain in India. Supermarkets are among the major retailers of fruits and vegetables in India. The wastage of fruits and vegetables also happens in supermarket stores. The reasons for fruits and vegetable wastage could be bad supplier selection, poor distribution network [Saeedi et al., 2019], poor storage facility, demand-supply gap, etc. The wastage should be reduced to assure a high profit. The need is to reduce wastage of fruits and vegetables. The need creates demand. The demand is an appropriate and robust supplier selection framework for fruits and vegetable stores.



Fig. 7.1. Fruits and vegetable supply chain in India

There is a lot of work that has been already published for supplier selection of different industries. In our literature study, we have seen that there is only one paper that directly deals with the supplier selection problem of fruits and vegetables. This chapter proposes a novel supplier selection framework for fruits and vegetable stores. A committee of experts is formed to analyze the problem and identify the conflicting criteria. The principal component analysis (PCA) is used to reduce the number of criteria as well as correlations among the criteria. The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is also used in this framework to evaluate the performance of the suppliers based on the reduced and uncorrelated criteria i.e. principal components.

7.2 Preliminaries

The proposed framework is an integrated framework. The techniques, which are used in this framework, are PCA, fuzzy set theory, TOPSIS. Those techniques are elucidated in this section for a better understanding of the complete framework.

7.2.1 Principal Component Analysis (PCA)

The principal component analysis is a very popular tool for multivariate data analysis. Karl Pearson [Pearson, 1901] proposed the basic concept of principal component analysis. Harold Hotelling [Hotelling, 1933] developed the technique in 1933 and the formulations in 1936 [Hotelling, 1936]. The main goal of PCA is to reduce dimension and identify the hidden pattern of a data set consisting of many variables correlated with each other. The detailed technique and formulations are described here.

Step 1: The raw dataset has m number of observations and n number of variables. The raw dataset ($D_{m \times n}$) contains the variable of different ranges or scales. This problem may lead to a biased result. The first task is to transform it to a comparable scale by normalizing the dataset. The dataset can be normalized using Eqs. (7.1), (7.2), and (7.3) by transforming in the same scale. d_{ij} represents i^{th} observation of j^{th} variable. r_{ij} is the normalized value of d_{ij} .

$$\bar{d}_j = \frac{1}{m} \sum_{i=1}^m d_{ij} \text{ where } i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad 7.1$$

$$\sigma_j = \sqrt{\frac{1}{m} \sum_{i=1}^m (d_{ij} - \bar{d}_j)^2} \text{ where } i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad 7.2$$

$$r_{ij} = \frac{d_{ij} - \bar{d}_j}{\sigma_j} \text{ where } i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad 7.3$$

A different normalization formula can be applied considering the type of criteria for multi-criteria decision making (MCDM) problems. The benefit criteria (higher the better) and cost criteria (lower the better) are normalized using Eq. (7.4) and Eq. (7.5) respectively. The normalized matrix is $ND_{m \times n}$.

$$d_i^+(j) = \frac{d_i(j) - [\min (d_i(j))]}{[\max (d_i(j))] - [\min (d_i(j))]} \quad 7.4$$

$$d_i^-(j) = \frac{[\max (d_i(j))] - d_i(j)}{[\max (d_i(j))] - [\min (d_i(j))]} \quad 7.5$$

Step 2: The covariance matrix is computed to identify the correlation among the variables. The covariance matrix is an $n \times n$ symmetric matrix if n is the number of variables. Eq. (7.6) is an example of an $n \times n$ covariance matrix.

$$C = \begin{bmatrix} cov(1,1) & cov(1,2) & \cdots & cov(1,n) \\ cov(2,1) & cov(2,2) & \cdots & cov(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(n,1) & cov(n,2) & \cdots & cov(n,n) \end{bmatrix} \quad 7.6$$

Step 3: Calculate the eigenvalues ($\lambda_1, \lambda_2, \dots, \lambda_n$) and the eigenvectors (v_1, v_2, \dots, v_n) from the covariance matrix. Refer to Eq. (7.7). Eigenvalues are simply the coefficients attached with eigenvectors, which represent the amount of variance of the corresponding principal component.

$$(C - \lambda I)v = 0 \quad 7.7$$

Step 4: A scree plot will be generated with the eigenvalues. The principal components will be selected from this scree plot. The selection of principal components should be done in such a way that the selected principal components will contain the maximum variance.

Step 5: Assume, k (where the value of $k < n$) number of eigenvectors are selected ($V_{n \times k}$) from n number of eigenvectors ($V_{n \times n}$). Now, compute the new dataset or decision matrix by performing the below steps-

- Transpose the selected Eigenvectors, $EV_{k \times n} = [V_{n \times k}]^T$
- Transpose the normalized dataset, $TND_{n \times m} = [ND_{m \times n}]^T$
- Compute $TD_{k \times m} = EV_{k \times n} \times TND_{n \times m}$
- Transpose $TD_{k \times m}$ to get a new dataset $TD_{m \times k} = [TD_{k \times m}]^T$, and rename the columns

7.2.2 Fuzzy Set Theory

A fuzzy set is a very popular mathematical tool to express vagueness and ambiguity. Prof. Lotfi Asker Zadeh introduced the fuzzy set in 1965 [Zadeh, 1965]. In a fuzzy set, if anyone asks you “How good you are at Driving?”. The answer may be ‘Very Good’, ‘Good’, ‘Average’, ‘Bad’, and ‘Very Bad’. This is the degree of membership. A fuzzy set \tilde{A} can be defined as in Eq. (7.8), where $\mu_{\tilde{A}}(x)$ is the membership function of x in the universe of discourse X .

$$A = \{(x, \mu_A(x)) | x \in X\} \quad 7.8$$

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } c \leq x \end{cases} \quad 7.9$$

Here, a triangular fuzzy membership function is used to represent the expert's rating in the decision matrix. Fig. 7.2 depicts the triangular fuzzy representation of linguistic terms. It is constructed with a triplet $\{a, b, c\}$. Refer to Eq. (7.9). Table 7.1 presents the linguistic rating and corresponding triangular fuzzy number as well as corresponding defuzzified crisp value. The center of gravity (COG) method is used for the defuzzification of the triangular fuzzy number.

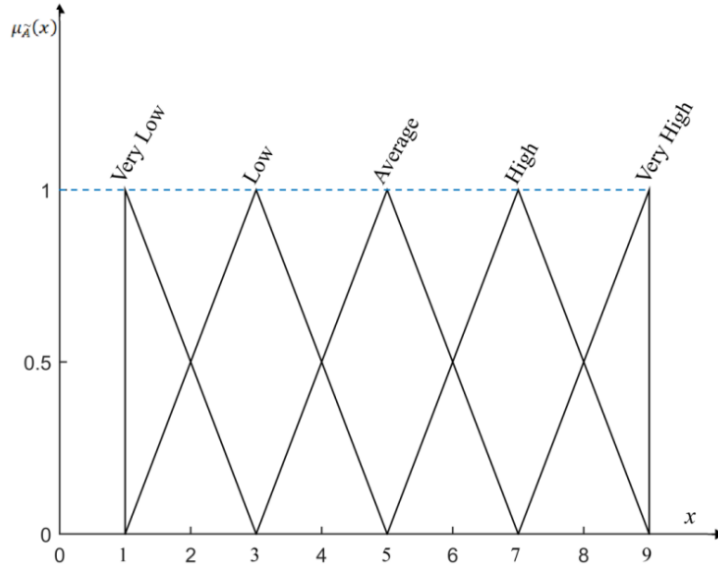


Fig. 7.2. Fuzzy membership functions of linguistic terms for expressing the rating

Table 7.1. Linguistic rating, fuzzy number, and defuzzified crisp value

Linguistic Terms	Abbreviation	Triangular fuzzy numbers	Defuzzified crisp number
Very Low	VL	(1,1,3)	1.66
Low	L	(1,3,5)	3
Average	A	(3,5,7)	5
High	H	(5,7,9)	7
Very High	VH	(7,9,9)	8.33

7.2.3 Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

The TOPSIS is a multi-criteria decision-making method. It was originated by Hwang and Yoon [Hwang et al., 1981]. Later some modifications were made by Yoon [Yoon, 1987]. The main idea behind this technique is that the best alternative should have a minimum distance from the positive ideal solution (PIS) and the maximum distance from the negative ideal solution (NIS). The detailed process of TOPSIS is described below.

Step 1: The computations of TOPSIS begin with the forming of a decision matrix (D). If there are m number alternatives and n number of criteria, then D will be an $(m \times n)$ matrix. Refer to Eq. (7.10). The element d_{ij} in D will represent the score of i^{th} alternative and j^{th} criteria.

$$D = \begin{matrix} & C_1 & C_2 & \cdots & C_n \\ A_1 & \left[\begin{matrix} d_{11} & d_{12} & \cdots & d_{1n} \end{matrix} \right. \\ A_2 & \left[\begin{matrix} d_{21} & d_{22} & \cdots & d_{2n} \end{matrix} \right. \\ \vdots & \left[\begin{matrix} \vdots & \vdots & \ddots & \vdots \end{matrix} \right. \\ A_m & \left[\begin{matrix} d_{m1} & d_{m2} & \cdots & d_{mn} \end{matrix} \right. \end{matrix} \quad 7.10$$

Step 2: The decision matrix is normalized and scaled within 0 to 1 using Eq. (7.11). R in Eq. (7.12) denotes the normalized matrix of D .

$$r_{ij} = \frac{d_{ij}}{\sqrt{\sum_{i=1}^m d_{ij}^2}} \quad 7.11$$

$$R = [r_{ij}]_{m \times n} \quad 7.12$$

Step 3: Construct a weighted normalized decision matrix by multiplying a weight with the corresponding criteria. $S_{m \times n}$ is the weighted normalized decision matrix computed using Eq. (7.13).

$$s_{ij} = w_j * r_{ij} \quad 7.13$$

Step 4: There are mainly two types of criteria i.e. benefit criteria and cost criteria. The highest score for benefit criteria and the lowest score for cost criteria are selected as the positive ideal solution (PIS). The lowest score for benefit criteria and the highest score for cost criteria are selected as the negative ideal solution (NIS). The PIS & NIS can be selected by following Eq. (7.14) and Eq. (7.15) respectively.

$$A^+ = (s_1^+, s_2^+, \dots, s_n^+) \text{ where } s_j^+ = \left\{ \left(\max_i s_{ij} \mid j \in J \right) \mid i = 1, 2, \dots, m \right\} \quad 7.14$$

$$A^- = (s_1^-, s_2^-, \dots, s_n^-) \text{ where } s_j^- = \left\{ \left(\min_i s_{ij} \mid j \in J \right) \mid i = 1, 2, \dots, m \right\} \quad 7.15$$

Step 5: Calculate the n -dimensional Euclidean distance from the positive ideal solution (DPIS) as well as from the negative ideal solution (DNIS) using Eq. (7.16) and Eq. (7.17) respectively.

$$E_i^+ = \sqrt{\sum_{j=1}^n (s_{ij} - s_j^+)^2}, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad 7.16$$

$$E_i^- = \sqrt{\sum_{j=1}^n (s_{ij} - s_j^-)^2}, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad 7.17$$

Step 6: Calculate the closeness co-efficient (CC) for each alternative. Refer to Eq. (7.18). Closeness co-efficient depicts that the alternative is how much close towards the PIS and far from NIS.

$$CC_i = \frac{E_i^-}{E_i^+ + E_i^-}, \quad i = 1, 2, \dots, m \quad 7.18$$

Step 7: Once the closeness co-efficient is computed the rank of the alternatives is done based on the closeness co-efficient in descending order.

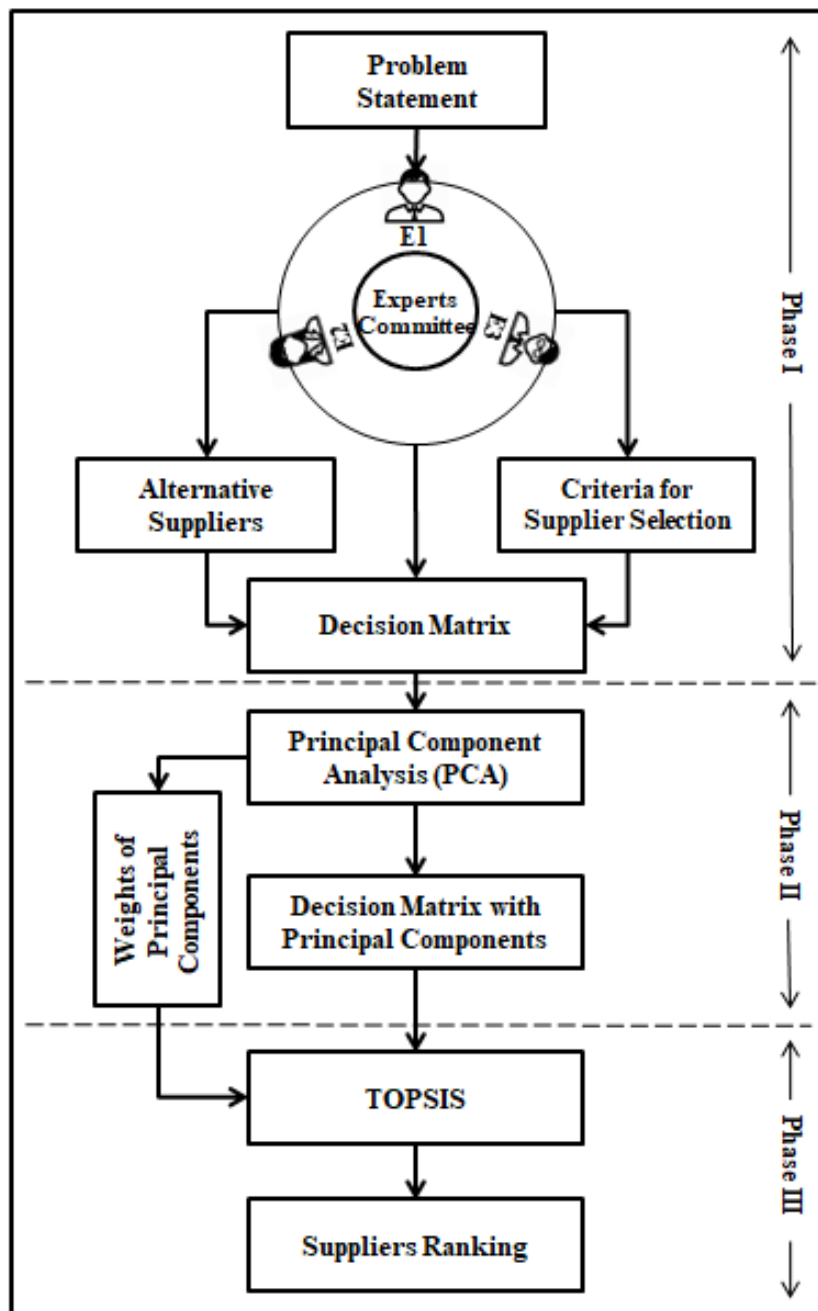


Fig. 7.3. The proposed framework for supplier selection of fruits and vegetable




7.3 Proposed Framework

Supplier selection is an existing problem for all industries. It has been addressed by different researchers with different techniques. Fruits and vegetables are highly perishable though it is stored in a freezer. The supplier selection problem became more challenging for perishable products. The supplier selection for the fruits and vegetable store is not explored enough. This chapter proposed a framework that is specially designed for the evaluation of fruits and vegetable suppliers. The proposed framework consists of three phases. Fig. 7.3 shows the complete flow of the proposed framework.

Phase I: The first task in this framework is to form a committee of experts. The committee consists of three experts with different gender, qualification, experience, area of expertise. Table 7.2 shows the details about the experts in the committee. A brainstorming session has been arranged among the experts to find the criteria for supplier selection. The expert's committee will give a rating for all the criteria of a supplier. The rating will be given based on their observation, understanding, and expertise. The expertise earned by acquiring explicit knowledge as well as tacit knowledge, which is like the tip of the iceberg.

The decision matrix (D) contains some alternatives i.e. suppliers and all the identified criteria. The decision matrix is filled with the expert's ratings. The ratings are given in linguistic terms by following Table 7.1. The linguistic inputs will be transformed into triangular fuzzy numbers. The fuzzy numbers are defuzzified using the center of gravity (COG) method for further processing.

Table 7.2. Details of the experts in the committee (Explicit Knowledge)

Experts			
	E1	E2	E3
Gender	Male	Female	Male
Age	35 Years	41 Years	62 Years
Qualification	BE	ME	Ph.D.
Experience	12 Years	18 Years	32 Years
Area of Expertise	Agricultural Engineer	Finance & Strategic Planning	Supply Chain Management

Phase II: In the second phase, the decision matrix is normalized. The principal component analysis is applied to the normalized decision matrix. The dimension of criteria will be reduced in this phase. The output of this phase is a new set of criteria, which are formed with the uncorrelated principal components. A new uncorrelated decision matrix is established with principal components and alternatives. The weights of the new criteria will be determined by the corresponding eigenvalues. The normalized eigenvalues of selected principal components will act as the weight for the corresponding criteria in the next phase.

Phase III: In this phase, TOPSIS is applied to the uncorrelated decision matrix. The weighted normalized decision matrix is computed by multiplying weights with the corresponding criteria in the normalized uncorrelated decision matrix. The next task is to find a positive ideal solution and a negative ideal solution. The closeness coefficient for all the suppliers will be determined by the distance from PIS as well as NIS. The suppliers are ranked by the descending order of the closeness coefficient.

7.4 Identification of Criteria

The criteria for the supplier selection of fruits and vegetables are identified in the first phase of the proposed framework. The expert committee has identified five top-level criteria. These are cost, quality, service, delivery, and supplier profile. All the top-level criteria have been divided into multiple sub-criteria. Table 7.3 shows the five top-level criteria as well as the sub-criteria. There are two types of criteria i.e. benefit criteria and cost criteria. The benefit criteria i.e. higher the better are marked with a '+' sign. The cost criteria i.e. lower the better are marked with a '-' sign. The first top-level criteria have been divided into three sub-criteria i.e. product price, discount on purchase, and payment policy. The price is the most important criterion for supplier selection of any product. The price is determined by the quality of the product. The price is proportional to the quality of the product. The discount on purchases attracts retailers for making deals with suppliers. Different supplier accepts different payment policy. The on-spot payment is always preferred by most of the suppliers. There is a trade-off between the discount on purchase and payment policy. The payment policy is very important for making a long-term deal with suppliers.

There are five sub-criteria of quality criteria. There are different quality parameters for different products. The quality is determined here by the product's visual appearance, packaging quality, matching with sample, consistency, and safety. The quality of fruits and vegetables cannot be

easily determined by brands like garments or other products. The visual appearance is the most important quality criterion for fruits and vegetables. The surface color determines the maturity of most fruits and vegetables. The shape, size, and weight determine the cost of fruits and vegetables. The packaging quality is also very important for suppliers because good packaging assures the reduction of damage at the time of transport. Good packaging is always in high demand for fruits and vegetable stores. The delivered product should match the sample shown at the time of placing the order. The consistency of the fruits and vegetable quality is also in high demand for the managerial of fruits and vegetable store. The match with sample and consistent product quality encourages the buyer to reorder and build trust in the supplier. Nowadays, the safety of edible products is a global concern. The farmers are using chemical pesticides and fertilizers to increase production in a shorter time. The fruits and vegetables produced by using chemical pesticides and fertilizers are very harmful to our health. Hence, the customers are preferring organic products over the product produced by using chemical pesticides and fertilizer.

Table 7.3. Hierarchical presentation of criteria

Serial No.	Criteria	Sub-criteria
1	Cost	Product price (-) Discount on purchase (+) Payment policy (+)
2	Quality	Surface appearance (+) Packaging quality (+) Matching with sample (+) Consistency (+) Safety (+)
3	Service	Service after the sale (+) Return and refund policy (+)
4	Delivery	Supplier's location (-) Damage during delivery (-) Delivery interval (-) Delivery efficiency (+) Timing flexibility (+) Quantity flexibility (+)
5	Supplier profile	Financial status (+) Cooperation (+) Storage condition (+) Storage capacity (+) Labor skill (+) Labor experience (+)

The service after the sale and return-refund policy of suppliers are two sub-criteria under the service criteria. The service after-sale help to build long-term relationships. The supplier with a smooth and flexible return refund policy gets more preference than other suppliers.

The delivery criteria have six sub-criteria. These are the supplier's location, damage during delivery, delivery interval, delivery efficiency, timing flexibility, and quantity flexibility. Supplier location is the most important criteria for a perishable product like fruits and vegetables. Transportation damage has a huge impact on the business of fruits and vegetables. The supplier of nearby locations is always preferred by the fruits and vegetable store because the local supplier reduces the transportation cost as well as reduces the chance of damage. The damage can also be reduced by using a freezer in the truck. But the availability of this kind of facilitated truck is very less in India. The minimum delivery interval is preferred because the customer always prefers fresh fruits and vegetables. Delivery efficiency is required for the delivery of any product. The flexibility of delivery time means the delivery in the desired time slot impresses the managers of the fruits and vegetable store. The flexibility of product quantity in order is also adding points to the supplier. The quantity flexibility satisfies the current demand for fruits and vegetable stores.

The supplier profile is also very important for making an order. There are six attributes of supplier profile that have an impact on supplier selection. These are financial status, cooperation, storage condition, storage capacity, labor skill, and labor experience. Financial status represents the stability of a supplier. Financial stability gives confidence in ordering. Cooperation from the supplier is always expected from the managerial personnel of any retail store. The hygienic condition, capacity, and other conditions of the store will be considered at the time of order. Storage hygiene has a direct impact on the safety of edible items like fruits and vegetables. The skill and experience of labor are a valuable addition to the supplier profile.

7.5 Numerical Example

The proposed framework has been demonstrated with a suitable numerical example. There are 22 criteria in this framework for supplier selection of fruits and vegetables. In phase I, six alternatives are taken in this example i.e. A1, A2, A3, A4, A5, and A6. The members of the expert's committee have a different level of emotional intelligence. The committee members are come up with a rating for each alternative based on each criterion. This rating will generate a decision matrix of 6×22 size. The ratings will be given in linguistic terms following Table 7.1. The linguistic rating will be converted to a triangular fuzzy number. The fuzzy number will be defuzzified by using the center of gravity method. The mapping of those conversions is already mentioned in Table 7.1. The defuzzified decision matrix is now ready for further computations in this framework.

In phase II, the decision matrix is normalized using Eqs. (7.4) and (7.5) for the higher the better criteria and lower the better criteria respectively. Table 7.4 shows the defuzzified and normalized decision matrix. The steps of PCA will be applied one by one on this defuzzified and normalized decision matrix. A covariance matrix will be computed from this defuzzified and normalized decision matrix. The dimension of the covariance matrix will be 22×22 . Then eigenvalues and eigenvectors are computed from this covariance matrix. There will be 22 eigenvectors and one eigenvalue for each eigenvector. The amount of variance of an eigenvector will be represented by the corresponding eigenvalue.

Table 7.4. The defuzzified and normalized decision matrix

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
A1	0.62	0.20	0.75	0.00	0.20	1.00	0.80	0.80	1.00	0.80	0.00
A2	1.00	0.50	1.00	0.38	0.50	0.38	0.20	0.00	0.75	0.50	0.00
A3	0.00	0.80	0.38	0.75	0.20	1.00	1.00	0.80	0.38	1.00	0.62
A4	0.00	1.00	0.00	0.75	0.80	0.38	0.50	0.20	0.00	0.00	0.25
A5	1.00	0.00	1.00	0.38	0.00	0.00	0.80	0.80	1.00	0.80	1.00
A6	0.25	0.50	0.38	1.00	1.00	0.38	0.00	1.00	0.38	0.80	0.62
	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
A1	0.75	0.40	0.75	0.25	0.50	0.80	1.00	0.00	1.00	0.20	0.00
A2	0.75	1.00	1.00	1.00	0.00	0.50	0.38	1.00	1.00	0.00	0.63
A3	0.37	0.40	1.00	0.00	0.50	1.00	1.00	0.75	0.63	1.00	1.00
A4	1.00	0.00	0.75	1.00	1.00	0.80	0.38	0.38	1.00	0.50	1.00
A5	0.00	0.40	1.00	0.63	0.50	0.00	0.00	0.38	0.25	0.20	0.63
A6	0.75	1.00	0.00	0.63	1.00	0.00	0.38	0.75	0.00	0.50	0.63

All the eigenvalues are plotted in a scree plot. Fig. 7.4 shows the scree plot of eigenvalues. The plot depicts that there is no variance from the sixth eigenvector. The eigenvalues from the sixth position are nearly zero. The value of k is five here. Hence, all the eigenvectors from the sixth position will be discarded. The first five eigenvectors are selected as the five principal components. The principal components will act as a criterion for upcoming steps. The criteria have been reduced from 22 to 5. The reduced uncorrelated decision matrix will be computed by following *Step 5* of PCA. Table 7.5 shows the reduced and uncorrelated decision matrix. The eigenvalues will be considered as the weight of the corresponding principal components. The weights are normalized for further processing. Table 7.6 presents the normalized weights of principal components.

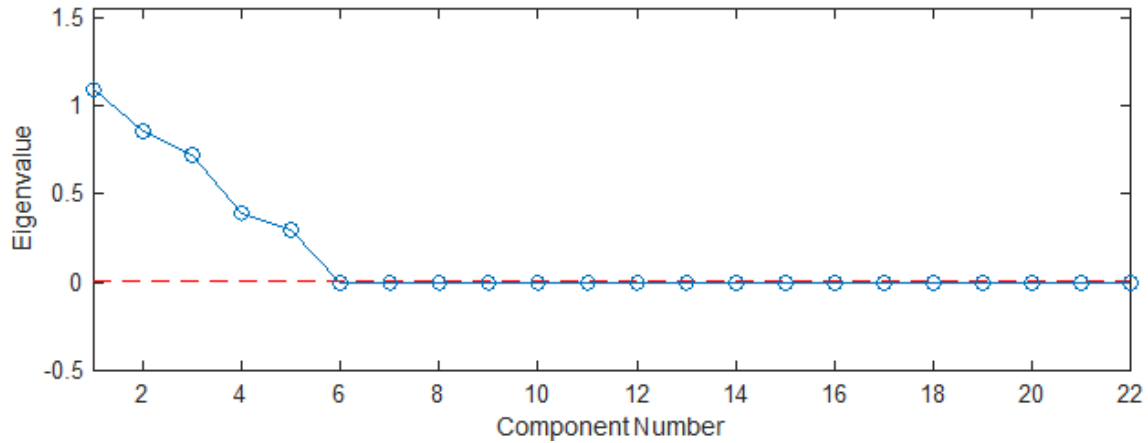


Fig. 7.4. Scree plot of the eigenvalues

Table 7.5. Decision matrix with uncorrelated principal components

	PC1	PC2	PC3	PC4	PC5
A1	-0.49	1.59	0.00	0.73	0.42
A2	-0.46	0.17	-1.17	0.13	1.61
A3	0.79	1.45	0.95	-0.24	1.61
A4	1.63	0.44	-0.63	-0.49	0.45
A5	-1.15	-0.09	0.64	-0.74	0.67
A6	0.76	-0.82	0.74	0.76	0.97

Table 7.6. Normalized weights of principal components

PC1	PC2	PC3	PC4	PC5
0.32	0.26	0.21	0.12	0.09

In phase III, the steps of the TOPSIS are applied one by one on the reduced and uncorrelated decision matrix. This matrix is the input decision matrix to the TOPSIS as mentioned in *Step 1* of TOPSIS. The weights are multiplied with the corresponding principal components on the normalized decision matrix of size 6×5 . Table 7.7 shows the weighted normalized decision matrix with principal components. The positive ideal solution (PIS) and negative ideal solution (NIS) are identified from the weighted normalized decision matrix. The initial decision matrix is transformed into the decision matrix of principal components. The type of criteria was considered at the time of this transformation. Hence, all the principal components are treated as higher the better criteria. The PIS and NIS are selected accordingly. The distance from the positive ideal solution (DPIS) and distance from the negative ideal solution (DNIS) is computed by using Eq. (7.16) and Eq. (7.17) respectively. The closeness coefficients (CC) are computed using Eq. (7.18) for all the alternatives. The closeness to PIS and farness from NIS is represented by this closeness coefficient. Table 7.8 shows the DPIS, DNIS, and CC for all the alternatives. The value of the closeness coefficient determines the rank of the alternatives.

The supplier is considered as the best supplier who has the maximum closeness coefficient. The rank of all the suppliers is given in descending order of closeness coefficient. Table 7.8 also shows the rank of the suppliers.

Table 7.7. Weighted normalized decision matrix with principal components

	PC1	PC2	PC3	PC4	PC5
A1	-0.07	0.17	0.00	0.06	0.01
A2	-0.06	0.02	-0.13	0.01	0.05
A3	0.11	0.16	0.11	-0.02	0.05
A4	0.22	0.05	-0.07	-0.04	0.02
A5	-0.16	-0.01	0.07	-0.06	0.02
A6	0.10	-0.09	0.08	0.06	0.03

Table 7.8. DPIS, DNIS, CC, and Rank

	DPIS	DNIS	CC	Rank
A1	0.31	0.33	0.52	4
A2	0.41	0.16	0.29	6
A3	0.14	0.44	0.75	1
A4	0.24	0.41	0.63	2
A5	0.44	0.22	0.33	5
A6	0.29	0.36	0.55	3

The result of this demonstration depicts that the 3rd supplier appears as the best among the six suppliers considered for evaluation. The complete rank of the suppliers is **A3 > A4 > A6 > A1 > A5 > A2**. Sensitivity analysis in the MCDM problem checks the stability of the decision. It is done to see the change in the final result if the weights of the criteria are interchanged. C_2^n number of combinations will be generated if there are n number of criteria and the weights between two factors are exchanged at a time. In this example, we have five criteria after the reduction of criteria in the decision matrix using PCA. Hence, 10 combinations can be possible by swapping two weights at a time. There will be 11 combinations of weights in total including the base scenario. Fig. 7.5 is the radar diagram of this sensitivity analysis. It shows the variation of the closeness coefficient for different combinations of weights. The plot also shows that supplier 3 (A3) comes first and supplier 2 (A2) comes last for most of the combinations of weights. The other alternatives are not deviating from their position for different combinations of weights as well. This analysis signifies that the evaluation is stable using this framework under different conditions.

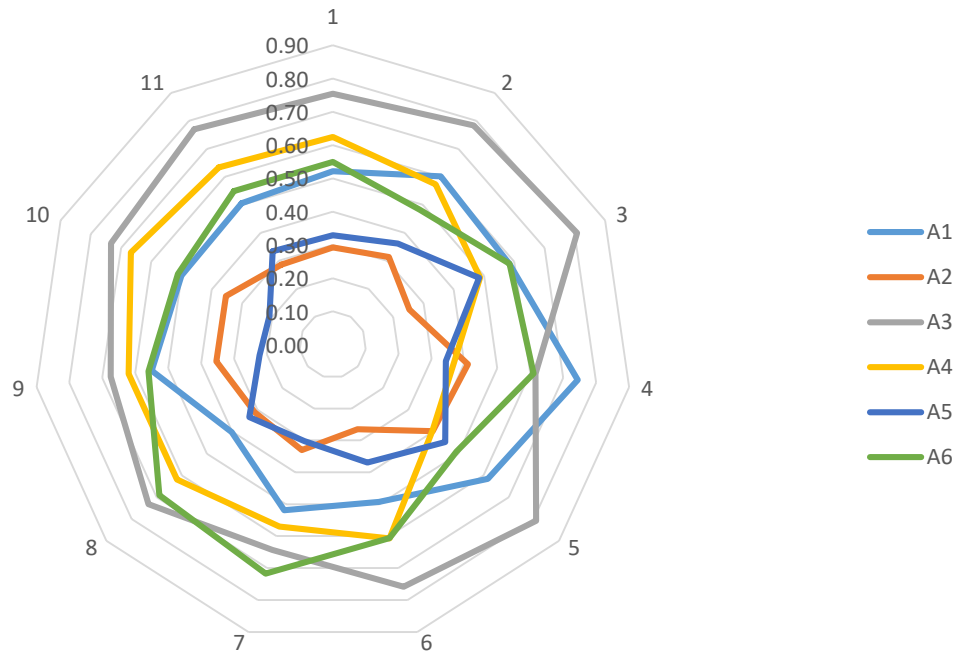


Fig. 7.5. Variation of closeness coefficients with a different combination of weights

7.6 Summary

This chapter proposes a holistic framework for supplier selection of fruits and vegetable stores. As per our literature survey, the proposed framework is the first complete framework for supplier selection of fruits and vegetables. The framework integrates the principal component analysis and technique for order of preference by similarity to ideal solution. The previous framework for supplier selection of different industries suffers from a lot of computations when the number of criteria is large. The proposed framework finds twenty-two criteria for supplier selection. There may have correlations among the criteria. There is a need to remove correlations among criteria as well as reduce computation. The proposed framework applied principal component analysis to remove the correlations among criteria as well as reduce the number of criteria by generating a decision matrix with principal components. In the previous frameworks, the weights are provided by experts or computed by using a separate technique. The experts do this based on their cognitive intelligence as well as emotional intelligence. Hence, the weights may be biased. This framework does this in the second phase by computing eigenvalues. The eigenvalues act like the weights of the corresponding principal components.

There is no importance of using principal component analysis in this framework if the number of criteria is very small. The framework can be tested by applying it to different industrial sectors for its supplier selection problem.

Conclusions & Future Scopes

8.1 Conclusions

The motivation of this work was to propose an end-to-end framework for the processing of fruits and vegetables. The processing includes segmentation and detection from the natural environment, sorting of fresh and non-fresh, classification among different types, volume and mass estimation. Supplier selection is required on top of all the processing for a supermarket. This thesis proposes an integrated framework addressing the challenges of those processing using image analysis and machine learning.

The segmentation and detection from the natural environment are very challenging because of the complex background. Here, the GrabCut technique has been adopted for segmenting fruit or vegetable objects from the background. A mincut algorithm is also used to separate the foreground and background pixels from the graph of pixels. An intelligent ROI marking technique has been introduced to address the conflict of choosing the ROI from the multiple foreground regions. The proposed technique is working better than the prior approaches in this context. It uses initial hard labeling which acts as a training to the system whereas most of the prior approach works on the global image.

The detection of non-fresh (rotten/defective) fruits and vegetables is very important. It should be done as soon as possible. The different types of fruit and vegetables have different types of

rot or defect pattern. The change in visual appearance (color, texture), flavor, and nutritional value are the symptoms of non-fresh fruits and vegetables. The flavor and nutritional value can not be perceived through computer vision. Hence, it is very tough to separate fresh and non-fresh relying only on visual appearance. Here, a convolutional neural network architecture has been proposed to separate fresh and non-fresh for any type of fruit and vegetables. The experimentation result shows the performance is better with the proposed approach than the prior approaches.

The classification of different types of fruits and vegetables is needed in many aspects of processing fruits and vegetables. There is a large number of publications for this problem in the literature. Still, there is a good number of challenges as well. A novel framework has been proposed here to address the challenges mentioned in chapter 1 for the classification problem. The framework extracts different shape features, color features from the top view of fruits and vegetables. The texture has been extracted from the side view of fruits and vegetables. There is a separate classification model for each view of fruit and vegetable. But, the final prediction module will give the final label for unknown fruit or vegetable sample. The framework has been tested for 35 classes of fruits and vegetables. The 10-fold cross-validation accuracy ranges from 99.90% to 100%. Some of the prior approaches have been implemented and tested in this context. The proposed approach outperforms every aspect of the previous approaches.

The grading of every product before selling is mandatory to get the proper price. The grading is done based on some predefined set of criteria. The grading parameters for fruits and vegetables are shape, size, color, volume, mass, etc. The estimation of volume and mass from the single view image of fruit and vegetable objects is very difficult than the estimation of other parameters. The prior volume and mass estimation techniques were mostly proposed for regular and axisymmetric fruits and vegetables. Here, a split and merge technique has been proposed for regular and axisymmetric as well as irregular and non-axisymmetric fruits and vegetables. The object has been divided into two-part then a polynomial equation has been formed for the boundary of each part. The volume is computed from both the polynomial. Finally, both the volume are merged to get the complete volume. The mass, volume, and density relation is used to estimate the mass. The result has been validated with the measured volume and mass by water displacement technique and digital balance respectively. The proposed methods outperform the prior approach for both the volume and mass estimation.

The supplier selection problem indirectly helps to reduce the burden of every stage of fruits and vegetable processing. The appropriate selection of suppliers is mandatory to reduce wastage and maximize profit for a fruits and vegetable store located in a supermarket. The proposed framework identifies the criteria and alternatives to make a decision matrix. It also removes correlation and reduces the number of criteria from the decision matrix. The proposed framework also contains an integrated weight generation technique for the reduced set of criteria. The final evaluation is done by generating the rank of the suppliers.

In summary, the achieved objectives are-

- A technique for fruits and vegetable segmentation and detection from the natural environment using image analysis and graph theory.
- A convolutional neural network architecture for sorting fresh and non-fresh fruits and vegetables using image analysis and deep learning.
- A framework for the classification of fruits and vegetables using image analysis and machine learning.
- A technique for the volume and mass estimation of fruits and vegetables using image analysis.
- A framework for the supplier selection of fruits and vegetables using multi-criteria decision making, fuzzy set, and machine learning.

8.2 Future Scopes

The agricultural application of computer vision and machine learning is a thrust area of research in the twenty-first century. There are many challenging problems in the agricultural field as well as in the agricultural industry that can be automated with the help of computer vision. The proposed framework addresses some challenges of existing problems of fruits and vegetable processing using image analysis and machine learning techniques. There are still some challenges for working in the future. The overlapping of fruits and vegetables creates challenges for the detection and counting of fruits and vegetables. The detection of the type of defect for a particular fruit or vegetable can be a value addition after marking it as non-fresh. The improvement in the proposed framework can be done to classify or grade the fruits and vegetables that are available as a branch. The research is a never-ending process because always there is a scope for improvements. Hence, further research can also be done to address the limitations of the proposed approach.

References

- Jidong, L., Wei, J., Fengyi, C., Dean, Z., & Bo, X. (2012, July). Research on the recognition method for obscured apple in natural environment. In Proceedings of the 31st Chinese Control Conference (pp. 3932-3937). IEEE.
- Jidong, L., De-An, Z., Wei, J., & Shihong, D. (2016). Recognition of apple fruit in natural environment. *Optik*, 127(3), 1354-1362.
- Xiang, R., Ying, Y., & Jiang, H. (2013a, December). A recognition algorithm for occluded tomatoes based on circle regression. In 2013 6th International Congress on Image and Signal Processing (CISP) (Vol. 2, pp. 713-717). IEEE.
- Xiang, R., Ying, Y., & Jiang, H. (2013b, December). Tests of a recognition algorithm for clustered tomatoes based on mathematical morphology. In 2013 6th International Congress on Image and Signal Processing (CISP) (Vol. 1, pp. 464-468). IEEE.
- Gatica, G., Best, S., Ceroni, J., & Lefranc, G. (2013). Olive fruits recognition using neural networks. *Procedia Computer Science*, 17, 412-419.
- Nanaa, K., Rizon, M., Rahman, M. N. A., Ibrahim, Y., & Aziz, A. Z. A. (2014, July). Detecting mango fruits by using randomized hough transform and backpropagation neural network. In 2014 18th International Conference on Information Visualisation (pp. 388-391). IEEE.
- Xiang, R., Jiang, H., & Ying, Y. (2014). Recognition of clustered tomatoes based on binocular stereo vision. *Computers and Electronics in Agriculture*, 106, 75-90.
- Lv, J., Wang, F., Ma, Z., & Rong, H. (2015, August). Yellow apple recognition method under natural environment. In 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics (Vol. 1, pp. 46-49). IEEE.
- Meng, J., & Wang, S. (2015, August). The recognition of overlapping apple fruits based on boundary curvature estimation. In 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA) (pp. 874-877). IEEE.

- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8), 1222.
- Toon, O. P., Zakaria, M. A., Nasir, A. F. A., Majeed, A. P. A., Tan, C. Y., & Ng, L. C. Y. (2019). Autonomous Tomato Harvesting Robotic System in Greenhouses: Deep Learning Classification. *Mekatronika*, 1(1), 80-86.
- Feng, Q., Wang, X., Wang, G., & Li, Z. (2015, August). Design and test of tomatoes harvesting robot. In *2015 IEEE International Conference on Information and Automation* (pp. 949-952). IEEE.
- Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3), 309-314.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- Chandini, A. A., & Maheswari B., U. (2018). Improved Quality Detection Technique for Fruits Using GLCM and MultiClass SVM. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.
- Karakaya, D., Ulucan, O., & Turkan, M. (2019). A Comparative Analysis on Fruit Freshness Classification. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-4). IEEE.
- Gómez-Sanchis, J., Martín-Guerrero, J. D., Soria-Olivas, E., Martínez-Sober, M., Magdalena-Benedito, R., & Blasco, J. (2012). Detecting rottenness caused by *Penicillium* genus fungi in citrus fruits using machine learning techniques. *Expert Systems with Applications*, 39(1), 780-785.
- Kamalakaran, A., & Rajamanickam, G. (2012, December). Surface defect detection and classification in mandarin fruits using fuzzy image thresholding, binary wavelet transform and linear classifier model. In *2012 Fourth International Conference on Advanced Computing (ICoAC)* (pp. 1-6). IEEE.
- Capizzi, G., Sciuto, G. L., Napoli, C., Tramontana, E., & Woźniak, M. (2015, September). Automatic classification of fruit defects based on co-occurrence matrix and neural networks. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 861-867). IEEE.
- Ranjit, K. N., Naveen, C., & Chethan, H. K. (2019). Fruit Disease Categorization based on Color, Texture and Shape Features. *International Journal of Computer Applications*, 178(49), 16-19.
- Dubey, A. K., Ratan, R., & Rocha, A. (2019). Computer vision based analysis and detection of defects in fruits causes due to nutrients deficiency. *Cluster Computing*, 1-10.
- Roy, K., Chaudhuri, S. S., Bhattacharjee, S., Manna, S., & Chakraborty, T. (2019a, March). Segmentation Techniques for Rotten Fruit detection. In *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)* (pp. 1-4). IEEE.

- Roy, K., Ghosh, A., Saha, D., Chatterjee, J., Sarkar, S., & Chaudhuri, S. S. (2019b, March). Masking based Segmentation of Rotten Fruits. In 2019 International Conference on Opto-Electronics and Applied Optics (Optronix) (pp. 1-4). IEEE.
- da Costa, A. Z., Figueroa, H. E., & Fracarolli, J. A. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering*, 190, 131-144.
- Maslow, A. H. (1943). A theory of human motivation. *Psychological review*, 50(4), 370.
- Singh, D., & Sharma, R. R. (2018). Postharvest diseases of fruits and vegetables and their management. In *Postharvest Disinfection of Fruits and Vegetables* (pp. 1-52). Academic Press.
- Cao, J., Wang, C., Xu, S., Chen, Y., Wang, Y., Li, X., & Sun, C. (2019). The effects of transportation temperature on the decay rate and quality of postharvest Ponkan (*Citrus reticulata* Blanco) fruit in different storage periods. *Scientia horticultrae*, 247, 42-48.
- Barrett, D. M., Beaulieu, J. C., & Shewfelt, R. (2010). Color, flavor, texture, and nutritional quality of fresh-cut fruits and vegetables: desirable levels, instrumental and sensory measurement, and the effects of processing. *Critical reviews in food science and nutrition*, 50(5), 369-389.
- Gould, W. A. (1977). *Food quality assurance*. AVI Pub. Co.
- Kramer, A. M. I. H. U. D. (1965). Evaluation of quality of fruits and vegetables. American Association for the Advancement of Science, Washington, DC, 9-18.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- Seng, W. C., & Mirisae, S. H. (2009, August). A new method for fruits recognition system. In 2009 International Conference on Electrical Engineering and Informatics (Vol. 1, pp. 130-134). IEEE.
- Roomi, S. M. M., Priya, R. J., Bhumes, S., & Monisha, P. (2012, December). Classification of mangoes by object features and contour modeling. In 2012 International Conference on Machine Vision and Image Processing (MVIP) (pp. 165-168). IEEE.
- Cornejo, J. Y. R., & Pedrini, H. (2016, November). Automatic fruit and vegetable recognition based on CENTRIST and color representation. In *Iberoamerican Congress on Pattern Recognition* (pp. 76-83). Springer, Cham.
- Rocha, A., Hauagge, D. C., Wainer, J., & Goldenstein, S. (2010). Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1), 96-104.
- Arivazhagan, S., Shebiah, R. N., Nidhyandhan, S. S., & Ganesan, L. (2010). Fruit recognition using color and texture features. *Journal of Emerging Trends in Computing and Information Sciences*, 1(2), 90-94.
- Zhang, Y., & Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *sensors*, 12(9), 12489-12505.

- Dubey, S. R., & Jalal, A. S. (2012). Robust approach for fruit and vegetable classification. *Procedia Engineering*, 38, 3449-3453.
- Haidar, A., Dong, H., & Mavridis, N. (2012, October). Image-based date fruit classification. In *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems* (pp. 357-363). IEEE.
- Chowdhury, M. T., Alam, M. S., Hasan, M. A., & Khan, M. I. (2013). Vegetables detection from the glossary shop for the blind. *IOSR Journal of Electrical and Electronics Engineering*, 8(3), 43-53.
- Ninawe, P., & Pandey, S. (2014). A completion on fruit recognition system using k-nearest neighbors algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3(7), 2352-2356.
- Wang, X., Huang, W., Jin, C., Hu, M., & Ren, F. (2014, November). Fruit recognition based on multi-feature and multi-decision. In *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems* (pp. 113-117). IEEE.
- Zawbaa, H. M., Hazman, M., Abbass, M., & Hassanien, A. E. (2014, December). Automatic fruit classification using random forest algorithm. In *2014 14th International Conference on Hybrid Intelligent Systems* (pp. 164-168). IEEE.
- Vogl, M., Kim, J. Y., & Kim, S. D. (2014, December). A fruit recognition method via image conversion optimized through evolution strategy. In *2014 IEEE 17th International Conference on Computational Science and Engineering* (pp. 1497-1502). IEEE.
- Kim, J. Y., Vogl, M., & Kim, S. D. (2014, October). A code based fruit recognition method via image conversion using multiple features. In *2014 International Conference on IT Convergence and Security (ICITCS)* (pp. 1-4). IEEE.
- Naskar, S., & Bhattacharya, T. (2015). A fruit recognition technique using multiple features and artificial neural network. *International Journal of Computer Applications*, 116(20).
- Kuang, H. L., Chan, L. L. H., & Yan, H. (2015, July). Multi-class fruit detection based on multiple color channels. In *2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)* (pp. 1-7). IEEE.
- Rachmawati, E., Khodra, M. L., & Supriana, I. (2015a, August). Histogram based color pattern identification of multiclass fruit using feature selection. In *2015 International Conference on Electrical Engineering and Informatics (ICEEI)* (pp. 43-48). IEEE.
- Rachmawati, E., Khodra, M. L., & Supriana, I. (2015b, October). Toward new fruit color descriptor based on color palette. In *2015 International Conference on Science in Information Technology (ICSITech)* (pp. 315-320). IEEE.
- Al-falluji, R. A. A. (2016). Color, shape and texture based fruit recognition system. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(7).

- Ronald, M., & Evans, M. (2016). Classification of selected apple fruit varieties using Naive Bayes. *Indian Journal of Computer Science and Engineering (IJCSE)*, 7(1), 13-19.
- Hossain, M. S., Al-Hammadi, M., & Muhammad, G. (2018). Automatic fruit classification using deep learning for industrial applications. *IEEE Transactions on Industrial Informatics*, 15(2), 1027-1034.
- Desai, M. (2019, September). A Fruit Recognition Approach for Refrigerator Inventory Management. In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (Vol. 1, pp. 1-6). IEEE.
- Pennington, J. A., & Fisher, R. A. (2009). Classification of fruits and vegetables. *Journal of Food Composition and Analysis*, 22, S23-S31.
- Soille P. (1999) Application Fields. In: Morphological Image Analysis. Springer, Berlin, Heidelberg.
- Dougherty, E. R., & Lotufo, R. A. (2003). Hands-on morphological image processing (Vol. 59). SPIE press.
- Mandelbrot, B. (1967). How long is the coast of Britain? Statistical self-similarity and fractional dimension. *science*, 156(3775), 636-638.
- Koch, H. V. (1904). Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire. *Arkiv for Matematik, Astronomi och Fysik*, 1, 681-704.
- Costa, A. F., Humpire-Mamani, G., & Traina, A. J. M. (2012, August). An efficient algorithm for fractal analysis of textures. In 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (pp. 39-46). IEEE.
- Feng, J., Lin, W. C., & Chen, C. T. (1996, August). Fractional box-counting approach to fractal dimension estimation. In Proceedings of 13th international conference on Pattern recognition (Vol. 2, pp. 854-858). IEEE.
- Liao, P. S., Chen, T. S., & Chung, P. C. (2001). A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.*, 17(5), 713-727.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6), 610-621.
- Mureşan, H., & Oltean, M. (2018). Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 10(1), 26-42.
- Vivek Venkatesh, G., Iqbal, S. M., Gopal, A., & Ganesan, D. (2015). Estimation of volume and mass of axi-symmetric fruits using image processing technique. *International journal of food properties*, 18(3), 608-626.
- Uluişik, S., Yildiz, F., & Özdemir, A. T. (2018, April). Image processing based machine vision system for tomato volume estimation. In 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT) (pp. 1-4). IEEE.

- Sabliov, C. M., Boldor, D., Keener, K. M., & Farkas, B. E. (2002). Image processing method to determine surface area and volume of axi-symmetric agricultural products. *International Journal of Food Properties*, 5(3), 641-653.
- Omid, M., Khojastehnazhand, M., & Tabatabaeefar, A. (2010). Estimating volume and mass of citrus fruits by image processing technique. *Journal of food Engineering*, 100(2), 315-321.
- Khojastehnazhand, M., Omid, M., & Tabatabaeefar, A. (2010). Determination of tangerine volume using image processing methods. *International Journal of Food Properties*, 13(4), 760-770.
- Iqbal, S. M., Gopal, A., & Sarma, A. S. V. (2011, November). Volume estimation of apple fruits using image processing. In *2011 International Conference on Image Information Processing* (pp. 1-6). IEEE.
- Siswantoro, J., Prabuwo, A. S., & Abdulah, A. (2013). Volume measurement of food product with irregular shape using computer vision and Monte Carlo method: a framework. *Procedia Technology*, 11, 764-770.
- Sa'ad, F. S. A., Ibrahim, M. F., Shakaff, A. M., Zakaria, A., & Abdullah, M. Z. (2015). Shape and weight grading of mangoes using visible imaging. *Computers and Electronics in Agriculture*, 115, 51-56.
- Gokul, P. R., Raj, S., & Suriyamoorthi, P. (2015, April). Estimation of volume and maturity of sweet lime fruit using image processing algorithm. In *2015 International Conference on Communications and Signal Processing (ICCSP)* (pp. 1227-1229). IEEE.
- Dhameliya, S., Kakadiya, J., & Savant, R. (2016). Volume estimation of mango. *International Journal of Computer applications*, 143(12), 11-16.
- Siswantoro, J., & Asmawati, E. (2016, October). A new framework for measuring volume of axisymmetric food products using computer vision system based on cubic spline interpolation. In *2016 2nd International Conference on Science in Information Technology (ICSITech)* (pp. 74-78). IEEE.
- Concha-Meyer, A., Eifert, J., Wang, H., & Sanglay, G. (2018). Volume estimation of strawberries, mushrooms, and tomatoes with a machine vision system. *International journal of food properties*, 21(1), 1867-1874.
- Nandi, C. S., Tudu, B., & Koley, C. (2016). A machine vision technique for grading of harvested mangoes based on maturity and quality. *IEEE sensors Journal*, 16(16), 6387-6396.
- Iqbal, S. M., Gopal, A., Sankaranarayanan, P. E., & Nair, A. B. (2015, March). Estimation of size and shape of citrus fruits using image processing for automatic grading. In *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)* (pp. 1-8). IEEE.
- Dang, H., Song, J., & Guo, Q. (2010, August). A fruit size detecting and grading system based on image processing. In *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics (Vol. 2, pp. 83-86)*. IEEE.

- Hu, M. H., Dong, Q. L., Malakar, P. K., Liu, B. L., & Jaganathan, G. K. (2015). Determining banana size based on computer vision. *International journal of food properties*, 18(3), 508-520.
- Lee, D. J., Archibald, J. K., & Xiong, G. (2010). Rapid color grading for fruit quality evaluation using direct color mapping. *IEEE transactions on automation science and engineering*, 8(2), 292-302.
- Islamadina, R., Pramita, N., Arnia, F., & Munadi, K. (2018, March). Estimating fish weight based on visual captured. In *2018 International Conference on Information and Communications Technology (ICOIACT)* (pp. 366-372). IEEE.
- Bozokalfa, M. K., & Kilic, M. (2010). Mathematical modeling in the estimation of pepper (*Capsicum annum* L.) fruit volume. *Chilean Journal of Agricultural Research*, 70(4), 626-632.
- Calbo, A. G., & Nery, A. A. (1995). Methods for measurement of gas volume of fruits and vegetables. *Journal of the American Society for Horticultural Science*, 120(2), 217-221.
- Al Ohali, Y. (2011). Computer vision based date fruit grading system: Design and implementation. *Journal of King Saud University-Computer and Information Sciences*, 23(1), 29-36.
- Biswas, S., & Hazra, R. (2018). Robust edge detection based on modified Moore-neighbor. *Optik*, 168, 931-943.
- Seo, J., Chae, S., Shim, J., Kim, D., Cheong, C., & Han, T. D. (2016). Fast contour-tracing algorithm based on a pixel-following method for image sensors. *Sensors*, 16(3), 353.
- Ji, C., Yang, X., & Wang, W. (2015, December). A novel method for image recognition based on polynomial curve fitting. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)* (Vol. 1, pp. 354-357). IEEE.
- Leddy, J. (1997). A Review of: "Review of Practical Handbook of Curve Fitting, Sandra L. Arlinghaus, Editor, CRC Press, Boca Raton, 1994".
- Biswas, S., Ghoshal, D., & Hazra, R. (2016). A new algorithm of image segmentation using curve fitting based higher order polynomial smoothing. *Optik*, 127(20), 8916-8925.
- Macon, N., & Spitzbart, A. (1958). Inverses of Vandermonde matrices. *The American Mathematical Monthly*, 65(2), 95-100.
- Lin, P. C., & Wu, L. S. (2011). How supermarket chains in Taiwan select suppliers of fresh fruit and vegetables via direct purchasing. *The Service Industries Journal*, 31(8), 1237-1255.
- Mohammed, A., Filip, M., Setchi, R., & Li, X. (2017, September). Drafting a fuzzy TOPSIS-multi-objective approach for a sustainable supplier selection. In *2017 23rd International Conference on Automation and Computing (ICAC)* (pp. 1-6). IEEE.
- Krishankumar, R., Ravichandran, K. S., & Saeid, A. B. (2017). A new extension to PROMETHEE under intuitionistic fuzzy environment for solving supplier selection problem with linguistic preferences. *Applied Soft Computing*, 60, 564-576.

- Mananawigapol, O., Jaturanonda, C., & Somboonwivat, T. (2018, April). Supplier selection considering sustainability criteria by using a hybrid evaluation method in printing business. In 2018 5th International Conference on Industrial Engineering and Applications (ICIEA) (pp. 548-554). IEEE.
- Koganti, V., Menikonda, N., Anbuudayasankar, S., Krishnaraj, T., Athhukuri, R., & Vastav, M. (2019). GRAHP TOP model for supplier selection in Supply Chain: A hybrid MCDM approach. *Decision Science Letters*, 8(1), 65-80.
- Negi, S., & Anand, N. (2014). Supply chain efficiency: an insight from fruits and vegetables sector in India. *Journal of Operations and Supply Chain Management*, 7(2), 154-167.
- Saeedi, F., Teimoury, E., & Makui, A. (2019). Redesigning fruit and vegetable distribution network in Tehran using a city logistics model. *Decision Science Letters*, 8(1), 45-64.
- Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6), 417.
- Hotelling, H. (1936). Relation between two sets of variates. *Biometrika*.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3), 338-353.
- Hwang, C. L., & Yoon, K. (1981). Methods for multiple attribute decision making. In *Multiple attribute decision making* (pp. 58-191). Springer, Berlin, Heidelberg.
- Yoon, K. (1987). A reconciliation among discrete compromise solutions. *Journal of the Operational Research Society*, 38(3), 277-286.
- Marsh, M., "GrabCut" - Interactive Foreground Extraction using Iterated Graph Cuts, <https://www.cs.ru.ac.za/research/g02m1682/> [Last Accessed: June, 2020].
- Kalluri, S. R.: Fruits fresh and rotten for classification, <https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification> [Last Accessed: May, 2020].
- Siddiqi, R.: apple-defect-detection, <https://github.com/raheelsiddiqi2013/apple-defect-detection> [Last Accessed: June, 2020].
- Francis, M., Galileo's Pendulum- Fractals for Fun, <https://galileospendulum.org/2012/01/31/fractals-for-fun/> [Last Accessed: June, 2018].
- National Horticulture Board (NHB), <http://nhb.gov.in/> [Last Accessed: April, 2019].
- Agriculture & Processed Food Products Export Development Authority (APEDA), http://www.apeda.gov.in/apedawebsite/six_head_product/FFV.htm [Last Accessed: April, 2019].