Classification of Vegetable Images Using Texture and Color Features

Irene Kagombe Atamba

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2023
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Zeki Bayram
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Önsen Toygar
Supervisor

Examining Committee

1. Prof. Dr. Önsen Toygar
2. Assoc. Prof. Dr. Yıltan Bitirim
3. Assoc. Prof. Dr. Mehtap Köse Ulukök

ABSTRACT

In this thesis, the aim to use vegetable images and implement a computationally cheap system to automatically classify vegetables using their texture and color features. In this respect, Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) approaches are used to classify vegetable features. Feature extraction is done based on three color space channels; XYZ color space, HSV color space and RGB color space. It generates the features using color space channels. The classifier is then utilized once the vegetable features have been created for each image. Experiments are conducted on Kaggle Vegetable Image Dataset using 15 different varieties of popular vegetables found all over the world that include bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, radish and tomato, and the results will be presented at the end of the thesis. Comparison of the effect of SIFT and SURF methods on different color space channels for vegetable classification is demonstrated.

Keywords: Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Vegetable Image Classification, Feature Extraction.

ÖZ

Bu tezde, sebze görüntülerini kullanmayı ve sebzeleri doku ve renk özniteliklerini kullanarak otomatik olarak sınıflandırmak için hesaplama açısından ucuz bir sistem uygulanması amaçlanmaktadır. Bu bağlamda, sebze özniteliklerini sınıflandırmak için Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) ve Hızlandırılmış Sağlam Öznitelikler (SURF) yaklaşımları kullanılmıştır. Öznitelik çıkarma işlemi üç renk uzayına göre yapılır; XYZ renk uzayı, HSV renk uzayı ve RGB renk uzayı kullanılarak renk uzayı kanalları ile öznitelikler oluşturulur. Sınıflandırıcı, her görüntü için bitkisel öznitelikler oluşturulduktan sonra kullanılabilir. Kaggle Sebze Görüntü Veri Kümesi üzerinde, tüm dünyada bulunan ve fasulye, acı su kabağı, şişe kabağı, brinjal, brokoli, lahana gibi 15 farklı popüler sebze çeşidi kullanılarak deneyler yapılmıştır. Kapsikum, havuç, karnabahar, salatalık, papaya, patates, balkabağı, turp ve domates ile elde edilen sonuçlar tez sonunda sunulmuştur. SIFT ve SURF metotlarının farklı renk uzayı kanalları kullanılarak sebze sınıflandırması üzerindeki etkisinin bir karşılaştırması gösterilmistir.

Anahtar Kelimeler: Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT), Hızlandırılmış Sağlam Öznitelikler (SURF), Sebze Görüntü Sınıflandırması, Öznitelik Çıkarma.

DEDICATION

This work is dedicated to all large and small-scale farmers in various countries throughout the world who have continuously provided the nations with foods and have helped put an end to Global Hunger.

My heart goes out to the "KITOOMA MIXED FARMERS" in my hometown village of Mbarara, Uganda, as well as to my father Mr. CHARLES KAGOMBE, who pioneered the various mixed farming projects in the community, and my mother Mrs. EDRAE NAAKUNDA KAGOMBE, who has devoted 26 years of her life to Crop farming and Animal farming by giving her family the best, most nourishing foods from her garden every day and supplying schools with healthy foods to incorporate into their diet.

ACKNOWLEDGMENT

My sincere gratitude to God Almighty who has given me the wisdom, knowledge, strength and has granted me sufficient grace to bring this work to completion. All the glory belongs to him.

I am incredibly grateful to my supervisor, Prof. Dr. Önsen Toygar, for all of her support and understanding while I was conducting this study. I would also want to express my heartfelt thanks to the Computer Engineering department staff; Dr. Felix Babalola, Dr. John Olaifa for their mentorship, support and encouragement all through my journey as a student in the computer engineering department.

And to my lovely family in Uganda "THE KAGOMBES" who have been my rock, and have continuously been my cheer from a thousands miles away.

To my Cypriot family, friends and Work family (EMU Registrar's office) that has made me fall in love with this country through their love, kind gestures, inclusivity, support and to many beautiful memories that we have created together you will forever hold a special place in my heart.

Lastly but not least to My Little Warrior GABRIEL WYATT ATAMBA who started this journey with me, thank you for all of the memories we've had to give each other throughout the years. You, son, are the most important piece of my life, and I love you. I am excited for your future and so thankful for you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
BIC	Border/Interior
CCVs	Color Coherence Vectors
CNN	Convolutional Neural Networks
DoG	Difference-of-Gaussian
FD	Fourier Descriptors
FMCG	Fast Moving Consumer Goods
GLCM	Gray-level Co-occurrence Matrix
HIS	Hue, Saturation, Intensity
HOF	Histograms of Optical Flow
HOG	Histogram of Oriented Gradient
HSV	Hue-Saturation-Value
KNN	K Nearest Neighbors
LBP	Local Binary Pattern
LoG	Laplacian of Gaussian
LSB	Least Significant Bits
MBH	Motion Boundary Histograms
PR	Progressive Randomization
RGB	Red, Green, Blue
SDA	Spatial Domain Analysis
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

SVMs	Support Vector Machine

Chapter 1

INTRODUCTION

One of the pillars of human nutrition and a necessity for a nutritious and well-balanced diet are vegetables. According to a recent study by a research specialist in agriculture and FMCG, the market for processing fruits and vegetables worldwide was expected to reach a value of over 304 billion US dollars in 2021. Around 898 million tons of vegetables, or nearly 80% of the world's total production, were grown in Asia in 2020. The three vegetables that are produced the most globally are tomatoes, onions, and cucumbers. People can satisfy their varied vitamin and mineral requirements with this kind of food from the food pyramid due to the fact that the manufacturing and distribution of vegetables involve certain manual activities. The categorization of vegetables is seen to be a crucial problem.

The similarities between various vegetables, namely color, texture, shape and size, can often make automatic categorization systems difficult to use. This complicates production and sales because a lot of time is spent labeling and counting the vegetables, which an automated system could do in less time. As a result, automated categorization of vegetables based on their features has become a critical component in vegetable classification and identification.

1.1 Vegetable Classification

Vegetables are best characterized or categorised based on the parts eaten or consumed for food. When different sections of a plant are edible, certain vegetables may fall into more than one category. There are two basic groups of vegetables: those with consumable vegetative parts, for example, stem, root and leafy vegetables, as well as those containing edible reproductive parts such as flower, fruit, and seed vegetables.

1.2 Types of Vegetables

Turnips, beets, carrots, radishes and sweet potatoes are examples of root vegetables. Sparagus and kohlrabi are examples of vegetables with stems. Potatoes are a kind of edible underground stem. Vegetables with leaves and leafstalks include Brussels sprouts, cabbage, celery, lettuce, rhubarb and spinach. Garlic, onions, and leeks are examples of bulb vegetables. Artichokes, broccoli, and cauliflower are among the plants that sprout from a head or flower. Because of how they are used, fruits like tomatoes, cucumbers, eggplant, okra, sweet corn, squash, peppers are commonly viewed as vegetables. Seed vegetables sometimes include legumes like peas and beans.

1.3 Steps to Perform Vegetable Classification

It is important to note that while classifying images, a specific image may be divided into n separate classes. It can take a long time to manually analyze and categorize images, especially when there are many of them. For this reason, it would be quite helpful if we could use computer vision to completely automate the procedure.

1.3.1 Image Pre-processing

This is the first stage in image classification, its primary goal is to enhance the image by emphasizing the interest parts or characteristics so that computer vision models can operate with the improved image. These steps are as follows: the image is read, resized, grayscaled, reflected, blurred with a Gaussian blur, histogram equalized, rotated, and translated.

1.3.2 Detection of an Object

Second stage is image segmentation or localisation of the position of the object of interest. Image segmentation is the method of breaking a digital image into discrete regions or segments that are represented as sets of pixels. It helps with the labeling process so that pixels with the same label have similar attributes. It’s also worth noting that image segmentation's main goal is to make an image's representation more concise, relevant, and understandable. Additionally, it helps in the recognition of boundaries and objects in images, including lines and curves.

Image texture can be used in image processing to describe areas into segments. Based on image texture, there are two forms of segmentation. Region Based segmentation, Boundary Based segmentation. While Boundary Based works to cluster pixels based on edges linking pixels with various texture features, Region Based strives to cluster pixels based on texture attributes. Despite not being considered a perfect metric for segmentation, image texture performs better when used in conjunction with other measures like color. When an image or picture is segmented, the results will display a collection of segments that cover the entire object or collection of features taken from the object.

1.3.3 Feature Extraction and Training

Feature extraction is the most crucial process since it is helpful for finding the most intriguing patterns in the image, characteristics that could be exclusive to a given class and subsequently be helpful in differentiating across classes. The technique through which the model absorbs the features from the dataset is referred to as model training.

1.3.4 Classification of an Object

In the last stage of image classification, the identified objects are assigned to the proper classes by comparing the image patterns to the target patterns in a suitable classification algorithm.

1.4 Machine Learning Algorithms

The techniques utilized to train the models are known as machine learning algorithms. Machine learning algorithms fall into three categories: supervised learning which employs methods like Regression and Classification on labeled data, unsupervised learning which uses techniques like dimensionality reduction and clustering on unlabeled data, and reinforcement learning algorithm where the model gains knowledge from each action it takes.

Currently, the market has a wide variety of machine learning algorithms, and with the quantity of research being done in this area, this number will only grow. Data scientists often start off with linear and logistic regression methods before moving on to more complex ones. Some examples of machine learning algorithms are given in the following subsections.

1.4.1 K-Nearest Neighbors

Classification and regression issues can be resolved using machine learning algorithms. Based on its K Nearest Neighbors (KNN), this technique evaluates the value of a new piece of data. K is frequently chosen as an odd number to minimize disagreement. The class with the highest mode among the neighbors is utilized to categorize a new data point. The mean is used as the value in the regression problem.

1.4.2 Support Vector Machines (SVMs)

Support Vector Machine is a classification approach in which two classes are divided by a hyperplane. In a binary categorization problem, the support vectors are two vectors from two different classes and the hyperplane is constructed at the greatest distance between the support vectors.

1.4.3 Naive Bayes

It is founded on the Bayes Theorem, which determines the likelihood of an event given specific factual circumstances.

The Bayes Theorem is stated as: The strategy is known as Naive because it presumes that all variables exist independently and that their existence has no influence on that of the other variables, which is never the case in real situation. Naive Bayes may therefore be used to classify information and spam emails.

1.4.4 Decision Tree

A decision tree is an example of supervised learning algorithm used for classification and regression modeling. Because regression is a predictive modeling tool, these trees are used to categorize data or anticipate what will happen next.

Decision trees like flowcharts in that they begin at the root node with a specific data inquiry and lead to branches that include potential responses. The branches then lead to decision (internal) nodes, which ask additional questions and provide more results. This continues until the data reaches a terminal (or "leaf") node and stops.

1.4.5 Random Forest

Random forest is a predictive modeling and behavior analysis approach based on decision trees. It is made up of several decision trees, each of which depicts a unique occurrence of how the input data was classified by the random forest. The forecast that obtains the most votes is chosen using the random forest approach, which analyzes each occurrence in isolation.

Samples from the original dataset are sent to each classification tree. The attributes are then chosen randomly and utilized to build the tree at each node. No tree in the forest may be cut down before the exercise is finished and the forecast has been definitively proved. In this regard, the random forest enables any classifier with weak correlations to produce a powerful classifier.

1.4.6 K-means Clustering

K-means Clustering is one of the unsupervised machine learning (ML) methods that data analysts utilize the most frequently and fundamentally. Clustering approaches, such as K-means, attempt to find commonalities within a dataset by putting data points into groups where those in one cluster are much more related to one another than those in another. In order to create clusters, several criteria are used, including the shortest distances, data point density, graphs and other statistical distributions. The K-means algorithm locates a certain number of centroids inside a data set, where a centroid is defined as the arithmetic mean of all data points in a given cluster. The algorithm then allocates each data point to the nearest cluster, with the goal of keeping clusters as small as possible.

1.4.7 Deep Neural Network (DNN)

Deep learning is one of the most common machine learning approaches today, in which computers are trained to execute certain activities that humans do instinctively.

Deep Learning, also known as Deep Neural Network (DNN), is characterized by a deep hierarchy that connects numerous internal layers for feature recognition and representation learning. The goal of representation learning is to learn how to express vital information extracted from observation data in the actual world. Feature extraction has traditionally required trial and error through artificial processes; however, Deep Learning employs a pixel level of the picture as input value, and learns to acquire and identify the characteristic that is most suited. A single layer perceptron network is the most basic type of neural network, consisting of a single layer of output and inputs that are supplied straight to the outputs. In this sense, it is the most basic type of feed-forward network.

1.4.8 Convolutional Neural Network (CNN)

CNN is a sophisticated and high-potential variant of the traditional artificial neural network model. It is designed to handle increasing levels of complexity, preprocessing, and data compilation. Layers are classified into three categories: width, height, and depth. The neurons in one layer do not link to all of the neurons in the following layer, but rather only a subset of the layer's neurons. The result is a single vector of probability scores grouped along the depth dimension. CNNs are one of the most efficient and adaptable models for specialized in image and non-image data. CNNs are capable of image recognition, image analysis, image segmentation, video analysis and natural language processing. CNN has been used effectively for object recognition.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Many methods for identifying and classifying fruits and vegetables utilizing color, shape, and texture attributes have been presented in previous years. However, within the food sector, several images could have similar color, shape, and texture features. Image analysis techniques are utilized in the agriculture and food industries for recognition and categorization reasons. Color, shape, texture and various defects of vegetables are highly essential features of vegetables for categorization and identification. In recent years, automated machine/computer vision systems have replaced manual vegetable categorization and recognition tasks as a result of improvements in machine/computer vision and the accessibility of affordable hardware and software.

In this chapter, some of the work of earlier researchers are outlined who have made significant contributions and innovations in the area of fruit and vegetable classification categorization and recognition using color, shape, size and texture image analysis techniques and some extraction techniques like Speeded Up Robust Features, Scale Invariant Feature Transform, Histogram of Oriented Gradient and Local Binary Pattern. K-nearest neighbor, Support Vector Machine, Artificial Neural Networks and Convolutional Neural Networks are examples of machine learning techniques.

2.2 Previous Work

In recent years, several image analysis techniques have been utilized in the field of Image Categorization to evaluate agricultural data for recognition and classification. Bolle et al. [1] pioneered image categorization by employing a fruit and vegetable identification algorithm. Texture, color, and density are all used in the system. Because it was established a few years ago, the system does not integrate the most recent advances. The reported accuracy was about 95% in several instances, however it was achieved by using the top four responses. While the Veggie-vision date set featured some more classes, the equipment that collects the pictures provided less specular lighting with more consistent color. The image data set collected in the supermarket exhibits higher lighting variations and significant color fluctuation between distinct photos.

Sego and Mirisaee [2] described a fruit identification system that can recognize seven distinct fruits. First, they use the KNN approach to identify photos of fruits based on the fruits' mean color, form area, roundness and perimeter values. Euclidean distance is utilized to determine the nearest fruit class by measuring the distance between the unidentified fruit's feature values and the saved feature values of each fruit class. Their recognition outcomes were up to 90% accurate.

To improve overall categorization accuracy, Rocha et al. [3] suggested a technique for classifying fruits and vegetables that considers a range of factors and employs the best classifier for each one. Vegetables and fruits are categorized using this technique according to their appearance, texture, and color. The appropriate classification method receives each feature after it has been concatenated.

Aibinu et al. [4] presents another method that employs a hybrid technique to automatically recognize and classify fruit using Artificial Neutral Network, Fourier Descriptors, and Spatial Domain Analysis. Fruit form identification is based on shape boundaries and signatures utilizing the Spatial Domain Analysis and Fourier Descriptors techniques. Color information collected throughout the training phase from an Artificial Neural Network is used to detect the fruit color. Then, for detecting and sorting fruits, fruit form recognition and color recognition routes are coupled. This method was tested using photos of apples, bananas, and mangos, yielding 99.1% accuracy.

Rocha et al. [5] describe a method for detecting vegetables and fruits in supermarkets. First, the color of an image is determined using global measurements such as histogram, means, contrast, homogeneity, energy, variance, correlation, and entropy over the histograms for each color channel. Color coherence vectors are utilized to create the pictures (CCVs). The image border is classified using the border pixel classifier (BIC). The look feature is created by employing a lexicon of components obtained through the use of K-means and a bottom-up clustering technique.

In addition, Heidemann [6] describes a method for automatically establishing image categories utilizing histograms, shape, and color descriptors using an unsupervised learning algorithm.

Zawbaa et al. [7] introduced a method for automatically recognizing fruit images, including three stages: pre-processing, feature extraction, and classification. The proposed approach was evaluated with 55 strawberries and 46 photos of oranges. The experimental findings show that the degree of variation between fruit types has an impact on categorization accuracy. The shape group's similarity accuracy records the lowest accuracy of the three groups. The highest results obtained while using the SVM classifier are 90.91% for apples and 78.89% for oranges. Classifying comparable fruits based on color yields results for apple of 96.97% and strawberry of 85.71% using SIFT as the feature extraction method. With unique fruits in both form and color, the highest accuracy may be achieved; it archives 100% of the time.

Arivazhagan et al. [8] suggested an effective merging for color and texture features for fruit detection using a minimal distance classifier built on statistical and co-occurrence information generated from Wavelet transformed sub-bands. The investigation was carried out on a supermarket Produce database that had around 2635 fruits from 15 distinct groups. Using 50% of the images from each group to train the system and the remaining images as the testing set, the fruit images are divided into training and testing sets. First, the fruit recognition system examined the color and texture features separately; it obtained 45.5% accuracy using color features, 70.8% accuracy using texture features, and 86% accuracy using both. Since color and texture information compliment one another, using them jointly improved categorization outcomes.

Using a deep learning architecture, Zhu et al. [9] proposed a high performance method for classifying vegetable images. The vegetable picture data set was trained using Caffe's AlexNet network model. The ImageNet dataset of vegetable images was compiled and divided into training and testing data sets. A total of 24000 photos were added to the set of images used in the experiment, with the training data set making up 80% (19200) and the test data set making up 20%. (4800). Five different types of vegetables were included in the visual data collection: cucumber, broccoli, cauliflower, mushrooms, and pumpkin. Each vegetable class has an equal number of training and testing datasets. The experiment contains 8000 iterations in total, with an increase in the number of repetitions, a drop in the loss rate, and a rise in the accuracy rate. From a dataset of 24,000 vegetable photographs, 10,000, 5,000, 1000 images were randomly selected and trained on the AlexNet network to examine the link between the accuracy rate of vegetable image classification and the number of image datasets. The 10,000, 5,000, and 1,000 vegetable picture data sets have accuracy and loss rates of 90.7%, 0.305, 86.6%, and 0.381, and 81.5%, 0.699, respectively. The standard BP neural network and the SVM classifier are used in a comparative experiment that the authors also run. The deep learning method's classification accuracy rate was clearly superior when the two approaches were employed to extract image color and shape attributes for image classification. The SVM classifier's multi classifier uses the one to one approach, and the experiment results validate the suggested method's efficacy.

Jana et al [10] suggested a technique that accounts for various types of fruits with the goal of properly and effectively identifying these distinct types of fruits. To differentiate the fruit in the foreground from the background, images must first go through preprocessing. The segmented image is used to extract texture features from the Gray-level Co-occurrence Matrix (GLCM) and statistical color features. A single feature description combines two types of features. These feature descriptors retrieved from the training dataset are used to train a Support Vector Machine (SVM) classification model. The suggested technique, when evaluated on both the training and validation sets, demonstrates that color features generate superior classification accuracy than texture features. 69.78% and 32.29%, respectively. Combining the proposed texture and color characteristics for categorization results in an increased overall accuracy of 83.33%. The main contributions of this research is the segmentation method that works well for varied colored fruit objects in the natural environment and an enhanced classification and identification methodology for fruits and vegetables.

A vegetable category recognition system utilizing deep neural network is suggested by Sakai et al. [11]. The authors employed the Deep Neural Network (DNN) for object category classification by extracting and learning the object. They investigated the Convolutional Neural Network and used deep learning to vegetable object recognition (CNN). According to the assessment results, 3 million iterations were appropriate for the vegetable recognition learning process using CNN. The results of the learning rate and recognition rate were 99.14% and 97.58%, respectively.

A fruit and vegetable categorization system employing Convolutional Neural Networks and image saliency is proposed by Zeng and Guoxiang [12]. They offer an effective method for classifying fruits and vegetables that makes use of visual saliency to delineate object areas and Convolutional Neural Network models to extract and apply image data. In accordance with the saliency map, the primary saliency zones are chosen using image saliency. The categorization of fruits and vegetables is trained using a VGG model. The research also establishes a library of images of fruits and vegetables that spans 26 categories and represents the main varieties found in nature. Their categorization method gets an exceptional accuracy rate of 95.6%, according to experiments done on their own database.

Femling et al. [13] method described in this research uses computer vision to automate the process of fruit and vegetable identification via self-service systems in the retail sector. The development of a user-friendly system, as determined by usability research, was another objective. Researchers tested two Convolutional Neural Network architectures (Inception and MobileNet) as fruit and vegetable classifiers for ten distinct varieties. Fast identification results with precise predictions were obtained by MobileNet. The disparity between the propagation durations is greater than the variances in accuracy between the networks, though. Images spread much more quickly with roughly the same precision using MobileNet. The top 3 accuracy for MobileNet is 97%. Despite the top 3 accuracy being excellent, MobileNet still has trouble predicting clementines and kiwis. A fresh collection of photos might be made for this type of kiwi to increase the accuracy.

In order to solve the multi-class and multiple fruit identification problem Kuang et. al [14] proposed a unique strategy based on efficient image region selection, the unique method for multi-class fruit detection proposed in this study makes use of efficient image region selection and enhanced object suggestions. To increase the detection accuracy, five complimentary features are used: local binary patterns (LBPs), global color histograms, global shape features, LBP based on magnitude of Gabor feature (GaborLBP), and histograms of oriented gradient (HOGs). For multi-class fruit detection, a new fruit dataset with five fruit classes has been created. The three main components of the suggested methodology are better suggestions, score-level feature fusion, and picture region selection. The suggested multi-class fruit detection is capable of detecting several fruits of different classes in a range of sizes, backgrounds, angles, locations, and picture circumstances.

In order to increase the precision of automatic vegetable detection and classification, Li et al. [15] proposes a deep learning-based approach for doing so. Using the open source Caffe deep learning framework, the enhanced VGG network model was trained using a collection of vegetable picture data. The output characteristic of the first two completely linked layers should be combined, as we suggest (VGG-M). To increase the VGG-M network's convergence speed and accuracy, Batch Normalization layers are added (VGG-M-BN). According to the experimental validation, the technique has a classification and recognition accuracy rate of up to 96.5% when compared to the VGG network (92.1%) and AlexNet network (86.3%).

In order to classify fruits effectively, Joseph et. al. [16] a Convolutional Neural Network was used in deep learning. It makes use of the 131 distinct fruit and vegetable classifications in the fruits 360 dataset to categorize photos. TensorFlow's backend was used to create the model. It received a 94.35% accuracy rating after 50 training epochs.

Yuhui et al. [17] develops a deep convolution neural network-based system for autonomous fruit and vegetable detection. A neural network is built with fewer parameters using depth wise separable convolution as opposed to the conventional standard convolution, making it ideal for hardware with constrained resources. For training and testing purposes, a small data set consisting of 12 common fruit varieties and 8 popular vegetable varieties is produced. This data is downloaded from the network and physically shot. The findings of the experiment indicate that the recognition accuracy is 95.67%.

CNN is used by the researchers Latha et al. [19] to classify fruits. In the categorization of fruits, the greatest accuracy was 97.4% with a 19.5 ms response time. Neural networks are typically used to build the deep learning models. The suggested method builds a model to categorize fruits using distinct CNN layer outputs and fully linked layers. Compared to the average pooling approach, max pooling removes and resizes the picture more effectively.

To categorize apples, colorful peppers, lemons, oranges, pomegranates, and tomatoes, Yuesheng et al. [20] proposed a "Circular fruit and vegetable classification based on optimized GoogLeNet." Through testing, the researchers discovered that Google Net can effectively satisfy the practical requirements. Its training accuracy is 96.88% and testing accuracy is 96.00%. the researchers observe that the improved model outperforms GoogLeNet in terms of training accuracy (98.82%), training speed (33.68 sheets/sec), training accuracy improvement (1.94%), testing accuracy improvement (2%), and other metrics.

Bayğın [21] presented a deep learning method to automatically classify 15 distinct varieties of vegetables. The study's extensive dataset includes 21,000 photos of vegetables. The dataset's photos are broken up into three categories for training, testing, and validation purposes. All of these groups were pooled within the parameters of the study, and a sizable dataset was produced. In the created machine learning model based on deep learning, feature extraction is done using the SqueezeNet architecture. Additionally, the most important elements were chosen using the ReliefF approach, and the least important aspects were eliminated. Linear Discriminant Analysis (LDA) was chosen as the classification approach for the application that was created. Hold-Out and 10-fold cross-validation procedures were applied in this investigation. Both validation procedures yielded accuracy values of about 99%. The study's findings demonstrate the viability of using the suggested strategy for automated vegetable categorization.

Ahmed et al. [22] made an effort is to classify vegetable images accurately. For this classification, a dataset of 21,000 photos from 15 classifications is used. The researchers test the effectiveness of CNN for classifying images of vegetables by building a CNN model from scratch. In order to evaluate the accuracy with the usual CNN, multiple pre-trained CNN architectures utilizing transfer learning are used. This paper suggests comparing common CNNs with their designs (VGG16, MobileNet, InceptionV3, ResNet, etc.) in order to determine which method would be most accurate and efficient when used with fresh picture datasets. All of the CNN architectures that have been suggested have experimental results. Additionally, a comparison between created CNN models and pre-trained CNN architectures is conducted. And the study demonstrates that the transfer learning approach may outperform classic CNN with a small dataset by making use of prior knowledge obtained from relevant large-scale studies. The proposed 6-layer CNN is adjusted and optimized for the working vegetable dataset, and it provides an accuracy of 97.5%, the highest compared to all prior work carried out by developing a model from scratch

Table 2.2 summarises the studies in the literature about vegetable and fruit classification with all the details such as method, type of images/classes, feature extraction methods used, etc.

Table 2.2: Literature review on vegetable and fruit classification

Author(s)	Year	Method	Type	Extracted Features	Dataset Information	Accuracy
Bolle et. al [1]	1996	Image Categorization (Veggie Vision)	Mixed	Texture Color Density	5000 and 150 classes	95%
Heidemann et al [6]	2005	Unsupervised learning Algorithm		Histograms Shape Color	3000 images	72.9%
Rocha et al [5]	2008	CCVs, BIC, Unser, K-means clustering	Mixed	Color, Texture, Appearance	2078 images and 11 classes	96%
Sego and Mirisaee[2]	2009	KNN	Fruit	Color	50 images	90%
Rocha et al.[3]	2010	SVM, LDA, Classification trees, K-NN and Ensembles of Tree	Mixed	Appearance, Texture Color	2633 images and 15 classes	
Arivazhagan et al. [8]	2010	Minimal distance classifier	Fruit	Color	2635 and 15 classes	45.5%
				Texture		70.8%
				Combined		86%
Aibinu et al [4]	2011	ANN, FD, SDA	Fruit	Shape, color		99.1%
Zawbaa et al. [7]	2014	SVM, K-NN SIFT	Fruit Apples Oranges	Shape, Color	178 images	90.91% 78.89%
			Apples Strawberry			96.97% 85.71%
Yuki et al.[11]	2016	DNN	Vegetable		200 and 8 classes	97.38%
Jana S et al [10]	2017	SVM GLCM	Fruit	Color	240 images and 8 classes	69.78%
				Texture		32.29%
				Combined		83.33%
Guoxiang Zeng [12]	2017	VGG	Mixed		3678 and 26 classes	95.6%
Zhu et al. [9]	2018	Caffe's AlexNet	Vegetable	Color, Shape	24000 (5 classes)	92.1%
		BP neural network				78%
		SVM				80.5%
Femling et al.[13]	2018	MobileNet Inception V3	Mixed	Color Appearance	4300 and 10 classes	96% 97%
Om et al.[23]	2018	Image preprocessing and InceptionV3	Vegetable		4 class and 1200 images	99%

Kuang et. al [14]	2018	Fused HOG, Local Binary Pattern(LBP) and GaborLBP	Fruits		5 class and 20433 images	99.5%
Li et al.[15]	2020	VGG VGG-BN	Vegetable		10 classes and 12000	95.8% 96.5%
Joseph et. al [16]	2021	Custom designed CNN	Mixed		131 class and 90483 images	94.35%
Yuhui et al. [17]	2021	Custom designed deep CNN	Mixed		20 class and 10756 images	95.67%
Bhavya et. al [18]	2021	CNN	Mixed		24 class and 3924	95.5%
R. S. Latha et al. [19]	2021	CNN	Mixed		12 classes and 6783 images	97.4%
Yuesheng et al.[20]	2021	GoogleNet based CNN	Mixed		6 class and 6600 images	98.82%
Bayğın M[21]	2022	Deep feature extraction(SqueezeNet), ReliefF, LDA	Vegetable		21000 3000 3000 images And 15 classes	99.69% 99.40% 99.33%

Chapter 3

FEATURE EXTRACTION AND CLASSIFICATION

3.1 What is Feature Extraction?

In computer vision and image processing, a feature is a piece of data that describes the content of an image; generally, whether a certain patch of an image has certain attributes. Features in an image might be distinct structures such as points, edges, or objects.

Feature Extraction is essentially a dimensionality reduction method in which raw data is sorted into linked manageable categories. In simplest terms, every pixel in a picture is a bit of data, and image processing works to extract just the most valuable information from the image while maintaining the pixels that characterize the image's features.

Feature extraction is a low-level image processing application. Color, size, shape and texture are a few elements that may be extracted from a picture.

3.2 Feature Extraction Techniques

Color, size, shape and texture are the most noticeable visual characteristics of vegetables.

A feature descriptor is a description of an image or a part of it that extracts valuable data while discarding irrelevant data. Its primary applications are image recognition and object identification.

3.3 Types of Feature Descriptors

A descriptor encodes an image to enable comparison and matching with other images. Local and global features are the two types of features that may be extracted from a picture or images. Global features are useful for low level applications such as object identification and classification, whereas local features are useful for higher level applications such as object recognition. It is worth mentioning that combining global and local information improves recognition accuracy. It is critical to understand the distinction between detection and identification. The process of locating something or identifying the presence of an object in a picture is referred to as detection. The process of establishing the identity of a detected object is known as recognition.

3.3.1 Local Descriptors

A local descriptor can be thought of as a patch, or more specifically, as the important areas of an image. This means that an image's texture is represented by local characteristics. It is also vital to remember that employing numerous local descriptors to match an image yields more precise outcomes and improved functionality. SIFT, SURF, LBP and GLCM are a few instances of local descriptors, demonstrate in next chapter.

Scale Invariant Feature Transform (SIFT) is a computer vision approach used for image recognition and feature description, as well as identifying locations and scales that may be assigned to the same object from different viewpoints. It can also recognize and characterize local elements in images. Despite the fact that SIFT has shown to be quite effective for object recognition applications, demand a significant amount of computer complexity, which is a significant disadvantage, particularly for real-time applications.

Speed up Robust Features (SURF) is a technique based on multi-scale space theory and the feature detector is based on the Hessian matrix, it offers excellent performance and accuracy. SURF is a more advanced method of Scale invariant feature transform descriptor. Although it closely resembles SIFT, the SURF technique outperforms SIFT without compromising the accuracy of the identified points.

Local Binary Patterns (LBP) is a simple grayscale invariant texture descriptor measure for categorization. In LBP, a binary code is created for each pixel by thresholding its neighboring pixels to either 0 or 1 dependent on the pixel's center value. Its primary goals are to improve precision and response time.

Gray level Co-occurrence Matrix (GLCM) is utilized for texture analysis. Two pixels are checked at once, the neighboring pixel and the reference pixel. Before computing the GLCM, a certain spatial connection is specified between the reference and neighboring pixels. Figure 3.1 depicts an example of GLCM computation. The image matrix is shown on the right and GLCM is shown on the left side. A red arrow represents the pixel pair (2, 2) with distance '1' and angle 0 in the first matrix, and this pixel pair occurs three times in the initial matrix, resulting in a number '3' in the GLCM at location (2, 2). GLCM is calculated in the same way for other pixel pairs.

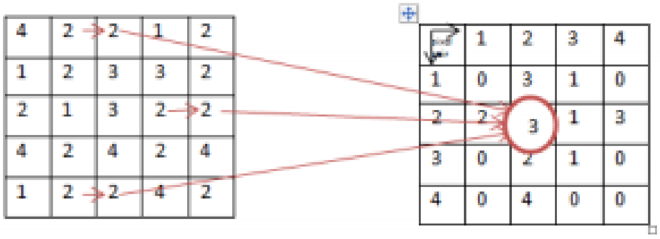

Figure 3.1: An example of GLCM computation. [24]

Gabor Filter (also known as Edge Detector): is a linear filter named after Dennis Gabor that is used in image processing to analyze texture. It evaluates whether or not the picture has any certain frequency content in specific directions within a constrained region surrounding the point or region of examination.

3.3.2 Global Descriptors

A global descriptor describes the whole image, that is to say it generalizes a whole complete image. Examples of global features may include; shape descriptors, texture features, contour representations. Global descriptor examples include Shape Matrices, Invariant Moments, Histogram Oriented Gradients, Histograms of Optical Flow, Co-HOG and motion boundary histograms.

3.4 Color Feature Extraction

Color is one of the most visually appealing aspects of any picture, and as such, it plays a vital part in the classification and identifying of vegetables. Color images consist of 3 channels namely; Red, Green and Blue. Using MATLAB code the color space of an image can be altered to any of the available possibilities, including XYZ, LAB, Grayscale, YCrCb, HIS, HSV.

A brief explanation for each color space is given and illustrated below.

3.5 Color Models

Color models are a means to define the visible color spectrum using either numerical values or color components. There are several different models of color spaces which are described in the following subsections.

3.5.1 RGB

Recognizing that an RGB image is just a composite of three different grayscale images that represent the intensity of red, green, and blue light in varying proportions to produce a wide spectrum of colors is essential to comprehending RGB image processing. In sensor and image-processing applications, the RGB model has shown to be incredibly effective.

3.5.2 XYZ

The initial model used by the CIE was the XYZ color space. The brightness of a color is shown in the Y channel. Although the Z value in the XYZ color space is different from the B value in the RGB color space, the Z channel often correlates to how much blue is present in a picture as shown in Figure 3.2. No distinct color equivalent exists for the X channel. When seen as a 3-D coordinate system, the X values in the XYZ color space are located along the axis that is orthogonal to the Y (luminance) and Z axes.

Figure 3.2: Image in RGB Color Space converted into XYZ Color Space

3.5.3 HSI

Hue, saturation, and intensity are abbreviated as HSI. It's vital to note that the HSI model represents color in a way that is far more in line with how people see it visually as shown in Figure 3.3. Hue is the actual color, whereas saturation is used to describe how intense the hue is in the light that reaches your eyes. Brightness is essentially intensity. Similar separation of intensity and color is accomplished using the HSI color model.

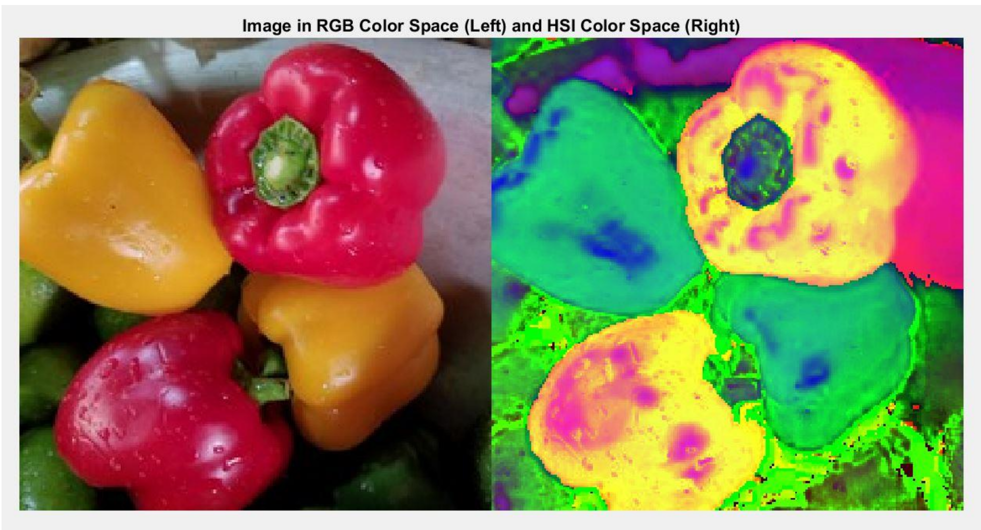

Figure 3.3: Image in RGB Color Space converted into HSI Color Space

3.5.4 HSV

The channel for defining color is called hue and it describes the prevailing wavelength. Saturation is a phrase used to describe the purity and colors of a color. The term value refers to the intensity of a color. Figure 3.4 shows an example.

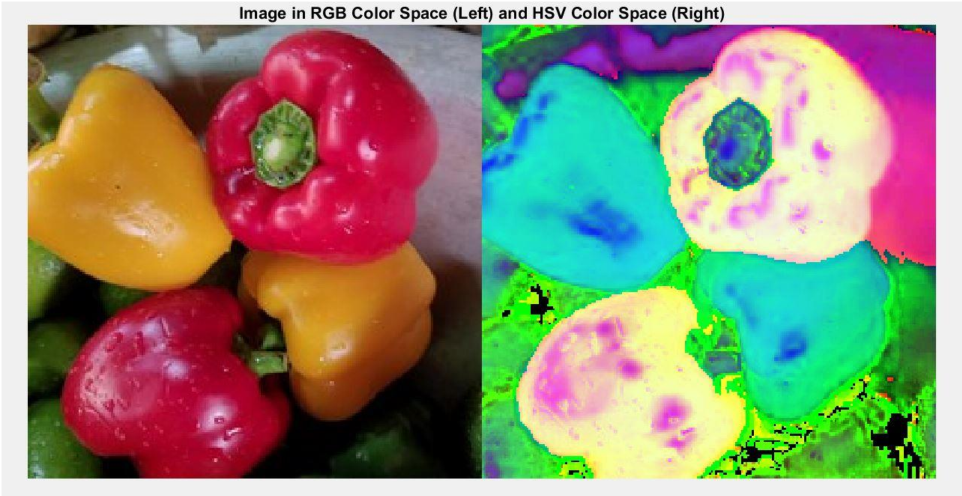

Figure 3.4: Image in RGB Color Space converted into HSV Color Space

3.5.5 LAB

There are three elements to the Lab color space as follows:

L: Lightness (Intensity).

a: a group of colors from Magenta to Green.

B: is a group of colors that ranges from Blue through Yellow.

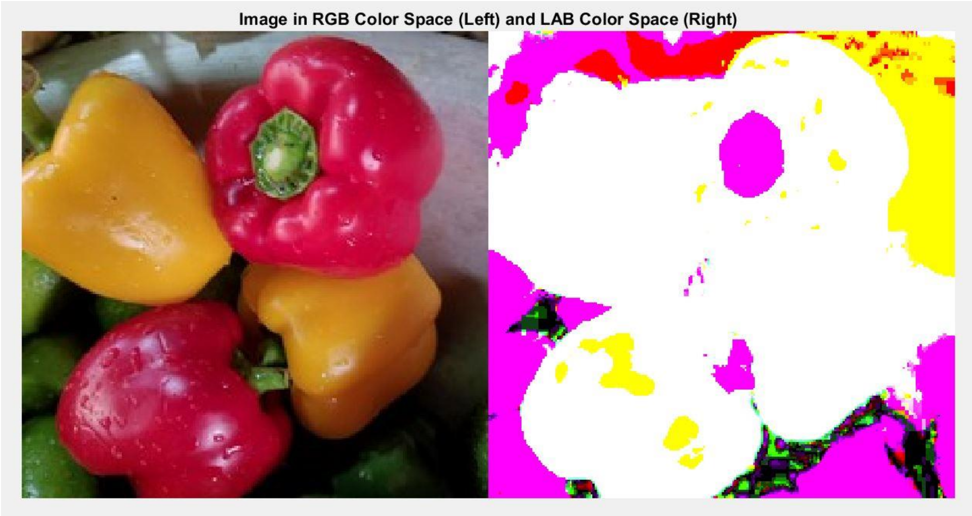

Figure 3.5: Image in RGB Color Space converted into LAB Color Space

The Lab color space differs greatly from the RGB color space as shown in Figure 3.5. Color information is divided into three channels in RGB color space, however the same three channels also contain brightness information. In Lab color space, on the other hand, the L channel is color-independent and solely encodes brightness. Color is encoded via the other two channels.

3.5.6 YCbCr

To reduce color, YCbCr use the RGB color model. The YCbCr algorithm splits visual data into three channels: luminance (Y), chroma blue (Cb), and chroma red (CR) (Cr). The luminance (Y) data of each red, green, and blue channel is retrieved and separated from the chroma data. To create a single luminance channel, RGB luminance data is encoded independently (Y). This data alone generates a complete and appropriate black and white picture. Figure 3.6 shows an example.

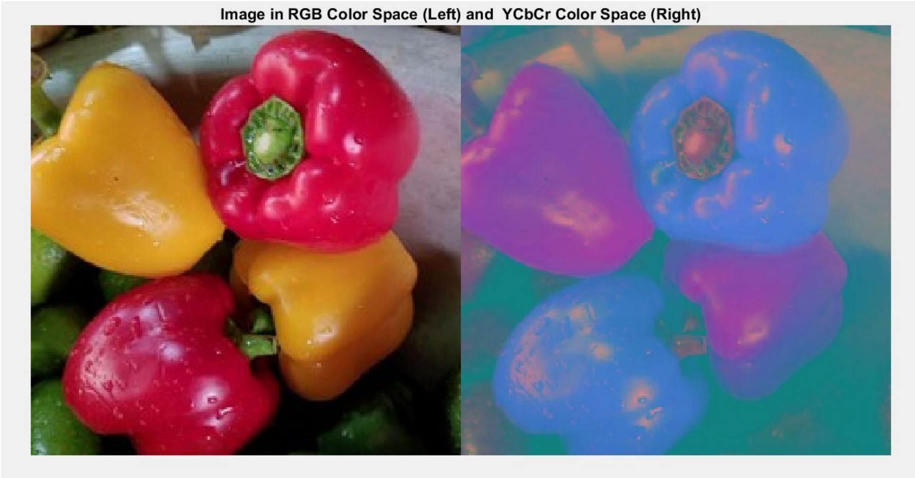

Figure 3.6: Image in RGB Color Space converted into YCbCr Color Space

The distinctions between these models are rather minor; at this time, the most crucial thing is to be aware that all three models are utilized and share a similar approach to characterizing color.

3.5 Image Matching

Image matching is a significant area for study in image processing and computer vision. Furthermore, it is a prerequisite for resolving many actual issues. A wide range of algorithms have been suggested by several academics who are committed to enhancing the effectiveness of image matching algorithms.

3.6 Image Matching Algorithms

There are two kinds of image matching algorithms: global feature-based matching algorithms and local feature-based matching algorithms.

3.6.1 Global Feature-Based Matching Algorithms

Global feature based matching algorithms extract typical global deep features from multi-modal images using deep convolutional networks.

3.6.2 Local Feature-Based Matching Algorithms

Local feature-based matching outperforms global feature-based matching in terms of dependability. They have been effectively used in a variety of real-world applications, including generating panoramas, object category classification, texture recognition, image retrieval, robot localisation, and object recognition. The following appropriate qualities should be present in good local features. Both speed and repeatability are quite high for feature detection. The minimal feature dimension of feature description makes it simple to match features quickly and provide robustness against changes in lighting, rotation, and perspective.

Local feature based matching techniques involve the identification and description of interest points.

3.6.3 Interest Points

An interest point is when two or more edge segments come together or where the border of the object abruptly changes direction.

The placement of interest points inside the image space is well-defined or well-localized. Even though size, rotation, and light vary both locally and globally, they remain stable. Consequently, it is essential that we are able to compute the interest points precisely, with great repeatability and that they enable efficient detection. Key points and interest points are the same thing.

3.6.4 Detection

Feature detection is the process of detecting essential image features (Identify the Interest Point) such as edges, corners, ridges, and blobs.

3.6.5 Description

Each feature point's local appearance is specified in a way that remains constant no matter how the illumination, translation, scale, or in-plane rotation is altered. Normally, each feature point results in a descriptor vector.

Chapter 4

METHODOLOGY

This chapter describes how the algorithms used in this thesis were implemented. It begins with setting up the datasets for training the system and progresses through how features were created by each method before presenting the performance evaluation of the suggested system. A block diagram is shown in Figure 4.1.

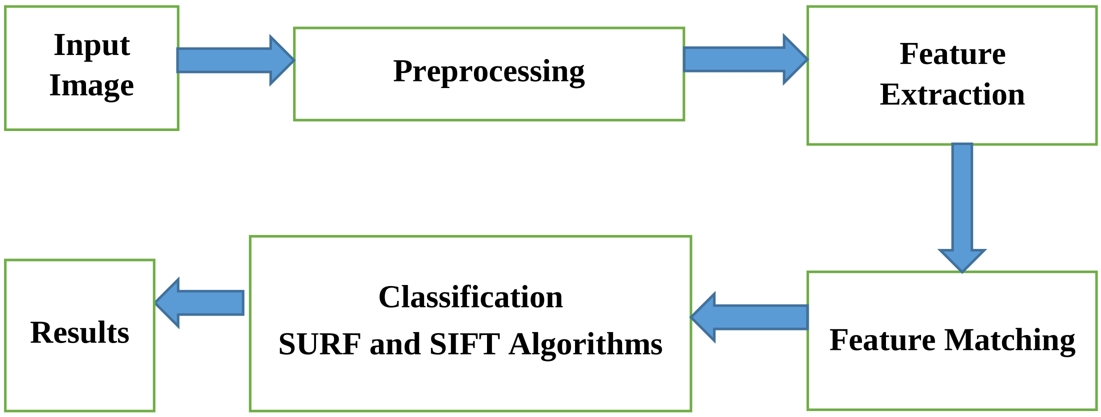

Figure 4.1: Overview of a Machine Learning System

4.1 SIFT Algorithm Overview

SIFT is a scale-space based image local feature description technique. SIFT offers a wide range of applications due to its great matching capabilities, including image retrieval, image stitching, and machine vision. Lowe [25] proposed the SIFT technique in 2004, it has great resilience and can tackle problems with image scaling, image rotation, and affine deformation, perspective shift, noise, and lighting variations. With regard to local features, image data is transformed into scale-invariant coordinates using the Scale Invariant Feature Transform technique. This approach creates a significant amount of features that cover the image densely at all sizes and locations, which is an essential component of the method. A normal 500x500 pixel picture will provide roughly 2000 stable features however, this number fluctuates according to the parameters used and the content of the image. When it comes to object recognition, the amount of features is very crucial since each item must have at least three correctly matched characteristics in order to be reliably identified when it is difficult to see them against a busy background. SIFT attributes are first retrieved from a collection of reference imagess that are kept in a database for the purposes of image matching and recognition. Individual image comparisons are used to match a new image.

4.1.1 Four Steps of SIFT Algorithm

In a SIFT algorithm, there are four steps: Scale Space Extrema Detection, Key point Localization, Orientation Assignment and Description Generation. Key points are identified and scaled utilizing scale space extrema in DoG (Difference-of-Gaussian) functions with different values in the first phase. The maxima and minima of Difference of Gaussian (DoG) images must be determined first. Eight neighbors at the same scale and nine neighbors at surrounding scales are taken into account when comparing each pixel in the DoG images. Pixels are considered as possible keypoints if they represent a local maximum or minimum. Orientation of the key point is then established using the local image gradient in the subsequent orientation assignment stage. The local image descriptor for each key point is calculated during the description generation stage using the magnitude and direction of the image gradient at each image sample location in the area focused around the key point. Each sample includes an 8-bin orientation grid and a 4-by-4 array of location bins, constructing a threedimensional histogram of gradient location and orientation. Its key point descriptor has a 128-element dimension.

Figure 4.2 displays the computation for the key point descriptor. The magnitudes and orientations of the image gradients are first sampled around the position of the key point before deciding how much Gaussian blur should be applied to the image. After the coordinates of the descriptor are changed, the gradient orientations are rotated in reference to the key point orientation to provide orientation invariance.

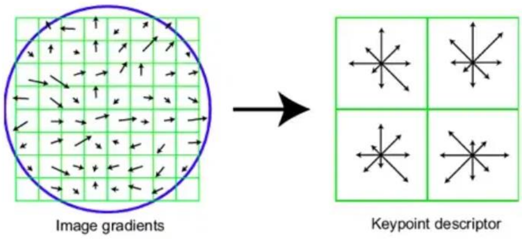

Figure 4.2: SIFT Descriptor Generation [24]

Figure 4.2 shows the keypoint description on the right side. By creating orientation histograms covering 4x4 sample patches. It enables for substantial changes in gradient positions. Each orientation histogram is represented by eight arrows, with the length of each arrow matching to the size of the histogram value. A gradient sample on the left can change sample locations up to four times while still contributing to the same histogram on the right. Each sample has a 4×4 array location grid as well as 8 orientation bins. Its key point descriptor has a 128-element dimension.

4.2 SURF Algorithm Overview

The SURF method is based on multi-scale theory, and the feature detector is based on the Hessian matrix. Analysis of SURF. While SURF and SIFT share a similar fundamental concept, they employ different techniques for location detection and descriptor generation. Bay [26] invented Speed Up Robust Features (SURF) to boost the efficiency of extreme point recognition and description since an image database contains a lot of information and SIFT takes a lot of time. For detection, SURF employs a fast Hessian matrix, which offers competitive benefits in terms of speed and accuracy.

SURF initially defines into multiple of 4× 4 square sub-regions around each extreme point in its close surroundings. Then, for each sub-region, a Haar wavelet response is computed. A four-dimensional vector is connected to each response. Each response is linked to a four-dimensional vector. An detailed 64-dimensional feature description is provided for each key point of every sub-region.

4.3 Detailed Comparisons of SIFT and SURF

Lowe [25] presented SIFT for extracting distinctive invariant features from images that can be invariant to image scale and rotation. It was widely used in image mosaic, recognition, retrieval and etc. Bay and Tuytelaars [26] proposed speeded up robust features and used integral images for image convolutions and Fast-Hessian detector. Their experiments turned out that it was faster and it works well. Table 4.1 shows detailed comparisons of SIFT and SURF based on the scale space theory, keypoint detection, orientation, descriptor, size of descriptor.

Table 4.1: Comparisons of SIFT and SURF

	SIFT ALGORITHM	**SURF ALGORITHM**
SCALE SPACE	The Difference of Gaussian (DoG) function is convolved with images of varying sizes using the same size of filter.	Laplacian of Gaussian (LoG) is used to convolve the integral image using different sizes of box filters.
KEYPOINT DETECTION	Using local extrema detection, Non maxima suppression, and Hessian matrix to eliminate edge response	Using the Hessian matrix and Non Maxima suppression, locate the key points.
ORIENTATION	The scale of the key point is utilized to calculate the amount of Gaussian blur for the image, and the amplitude and orientation of the image gradient are sampled around the key point location. For the same, the histogram's orientation is used.	The prevalent orientation of the Gaussian weighted Haar Wavelet responses at each sample point within a circular neighbourhood surrounding the interest points is detected via a sliding Orientation window of size $\pi/3$.
DESCRIPTOR	The keypoint descriptor generates orientation histograms spanning 4 × 4 sample sections, allowing for a substantial change in gradient locations. For each orientation histogram, with the length of each arrow corresponding to the magnitude of the histogram entry.	The interest point is covered by an orientation quadratic grid with 4x4 square subregions. The wavelet responses are calculated from 5x5 samples for each square. Descriptor of SURF is $V = (\sum dx , \sum dy , \sum \lvert dx \rvert , \sum \lvert dx \rvert)$
SIZE OF DESCRIPTOR	128 bits	64 bits

Chapter 5

EXPERIMENTS AND RESULTS

5.1 Experimental Setup

The first set of experiments assesses the individual classification performance of the SIFT and SURF descriptors on the 300 images from Kaggle Vegetable Dataset.

The experiment is carried out using 15 different varieties of popular vegetables found all over the world. The vegetables include tomato, bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, and radish. The developed methodology is consisting of three steps. In the first step from A total of 21000 images of size 224×224 and in *.jpg format in Kaggle Vegetable Image Dataset, 300 vegetable photos are carefully selected, there are two phases: training and testing. Each Vegetable class is given 20 images, 10 of which are utilized for training and the rest for testing. The system is trained with 150 photos, and it learns about each one. The system is trained to recognize vegetables based on color throughout the training phase. Using previous information about each vegetable image, the algorithm identified the image of the unknown vegetable. The system analyzes the color properties of the unknown image and compares them to those recorded in the database before classifying the unknown image to the desired known vegetable image using the information gained during training. The second stage involves extracting features from the Vegetable pictures. In this study, we employed color feature extraction after capturing and resizing the RGB photos of various vegetables, we converted them into multiple color space channels like RGB, XYZ, and

HSV using the MATLAB command to improve classification accuracy. In the final step, SIFT and SURF algorithms are utilized to further classify the vegetables based on knowledge gained during the training phase. For each dataset, the breakdown is illustrated in Tables 4.1 through 4.9.

5.2 Datasets Used

Table 5.1 shows detailed features of dataset a total of 21000 images of size 224×224 and in *.jpg format in Kaggle Vegetable Image Dataset, 300 vegetable photos are carefully selected from 15 Vegetable classes namely; tomato, bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, and radish. Figure 5.1 shows some of the vegetable pictures taken from the Kaggle Vegetable Image Dataset.

Table 5.1: Detailed information of Kaggle Vegetable Dataset

Detailed information	**Values**
Number of classes	15
Class names	bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, and radish, tomato
Total number of images	300
Number of images in each class	20
Image type	JPG
Images of size	224x224

Fıgure 5.1: Images taken from Kaggle Vegetable Image Dataset [27]

Table 5.2 shows the vegetable classes arranged in ascending order. Classes are denoted by numbers from 1 to 15 for example Beans is vegetable class (1) Bitter Gourd vegetable class (2), Bottle Gourd vegetable class (3), Bringal vegetable class (4), Broccoli vegetable class (5), Cabbagge vegetable class 6), Capsicum Assorted (7), Capsicum Mixed (8), Carrot (9), Cucumber (10), Papaya (11), Potato (12), Pumpkin (13), Radish (14) and Tomatoes vegetable class (15). For each vegetable class 20 images are selected and the system divides the images into 50%, 10 images are used for training and the remaining 10 are used to test the system. This is done for all the Vegetable Classes.

Table 5.2: Number of Images used in Experiment from Kaggle Dataset

VEGETABLE CLASSES	**NUMBER OF TRAIN IMAGES**	**NUMBER OF TEST IMAGES**
Beans (1)	10	10
Bitter Gourd (2)	10	10
Bottle Gourd (3)	10	10
Bringal (4)	10	10
Broccoli (5)	10	10
Cabbagge (6)	10	10
Capsicum Assorted (7)	10	10
Capsicum Mixed (8)	10	10
Carrot (9)	10	10
Cucumber (10)	10	10
Papaya (11)	10	10
Potato (12)	10	10
Pumpkin (13)	10	10
Radish (14)	10	10
Tomatoes (15)	10	10
TOTAL IMAGES (300)	**150**	**150**

5.3 Experimental Results

In the first experiment, SIFT algorithm is used to extract texture features from individual image vegetable classes and classify them. Table 5.3 shows that the highest results obtained while using the SIFT Classifier are classes; Bottle Gourd (100%), Bringal, (100%), Cucumber (100%). The total Average of correctly recognized individual Vegetable Classes is 80% and 19.3% for incorrectly recognized individual Vegetable Classes.

In the next experiment, SURF algorithm is used to extract texture features from individual image vegetable classes and classify them. Table 5.4 shows that the highest results obtained while using the SURF Classifier are classes 3,5,10,13; Bottle Gourd(100%), Broccoli (100%), Cucumber(100%) and Pumpkin(100%). The total Average of correctly recognized individual Vegetable Classes using SURF is 84.7% and 15.3% for incorrectly recognized individual Vegetable Classes.

Table 5.3: Evaluation of SIFT Descriptor and Classifier on Individual Image Vegetable Classes

VEGETABLE CLASSES		**SIFT ALGORITHM ACCURACY (%)**
Training classes (Number of images)	Testing classes (Class number)	Correct Recognition
15 classes (150)	Beans (1)	90
15 classes (150)	Bitter_Gourd (2)	90
15 classes (150)	Bottle_Gourd (3)	100
15 clas3es (150)	Bringal (4)	100
15 classes (150)	Broccoli (5)	30
15 classes (150)	Cabbagge (6)	70
15 classes (150)	Capsicum Assorted (7)	90
15 classes (150)	Capsicum Mixed (8)	80
15 classes (150)	Carrot (9)	90
15 classes (150)	Cucumber (10)	100
15 classes (150)	Papaya (11)	90
15 classes (150)	Potato (12)	60
15 classes (150)	Pumpkin (13)	90
15 classes (150)	Radish (14)	70
15 classes (150)	Tomatoes (15)	50
TOTAL AVERAGE (%)		**80%**

Table 5.4: Evaluation of SURF Descriptor and Classifier on Individual Image Vegetable Classes

VEGETABLE CLASSES		SURF ALGORITHM ACCURACY (%)
Training classes (Number of images)	Testing classes (Class number)	Correct Recognition
15 classes (150)	Beans (1)	90
15 classes (150)	Bitter_Gourd (2)	90
15 classes (150)	Bottle_Gourd (3)	100
15 classes (150)	Bringal (4)	90
15 classes (150)	Broccoli (5)	100
15 classes (150)	Cabbagge (6)	50
15 classes (150)	Capsicum Assorted (7)	90
15 classes (150)	Capsicum Mixed (8)	60
15 classes (150)	Carrot (9)	60
15 classes (150)	Cucumber (10)	100
15 classes (150)	Papaya (11)	90
15 classes (150)	Potato (12)	80
15 classes (150)	Pumpkin (13)	100
15 classes (150)	Radish (14)	80
15 classes (150)	Tomatoes (15)	90
TOTAL AVERAGE (%)		**84.7%**

Table 5.5 is a comparison table that shows performance of system according to the time spent training the Vegetable Dataset using feature descriptors SIFT and SURF as classifiers.

Table 5.5: Comparison of Results of Classification of Vegetables Using SIFT and SURF

FEATURE DESCRIPTORS	ACCURACY (%)	TIME SPENT(s) For Training
SIFT	80%	1293s
SURF	84.7%	50s

Table 5.6 is a comparison table that shows the time taken for feature matching varying vegetable classes using SIFT and SURF. According to the results obtained SURF had the least amount of time spent for feature matching; 50 seconds for testing one class and 2 minutes and 52 seconds for testing 15 classes were achieved. For SIFT, 9 minutes and 35 seconds were spent to test one class and 21 minutes and 33 seconds for testing 15 classes.

Table 5.7 and 5.8 shows correct detection rate for each color space using three color channels; RGB, XYZ and HSV to extract color features separately from the individual vegetables classes. SIFT and SURF algorithms are applied afterwards as feature extractors.

Table 5.6 Comparison of Time taken for Feature Matching Vegetables Using SIFT and SURF

FEATURE DESCRIPTORS	**VEGETABLE CLASSES**		**TIME(s)**
	Training Images (10 Samples For Each Class)	Testing Images (10 Samples For Each Class	Time Taken For Feature Matching
SIFT	15 classes (150)	1 class (10)	**575s**
	15 classes (150)	15 classes (150)	**1293s**
SURF	15 classes (150)	1 class (10)	**50s**
	15 classes (150)	15 classes (150)	**172s**

Table 5.7: Correct Detection Rate for each Color Space using three Channels of the Space separately and Apply SIFT Algorithm

VEGETABLE CLASSES		**COLOR SPACES**								
Training Images	Testing Images	**R**	**G**	**B**	**X**	**Y**	**Z**	**H**	**S**	**V**
15 classes	Beans	90	80	80	80	80	80	40	100	90
15 classes	Bitter Gourd	90	90	100	100	90	90	50	100	90
15 classes	Bottle Gourd	100	100	100	100	100	100	50	100	100
15 classes	Bringal	100	100	100	100	100	100	50	90	100
15 classes	Broccoli	50	30	30	40	50	30	40	60	40
15 classes	Cabbagge	50	70	50	50	60	50	20	50	40
15 classes	Capsicum Assorted	80	80	80	90	90	80	60	70	80
15 classes	Capsicum Mixed	50	80	70	70	90	60	30	50	50
15 classes	Carrot	70	100	60	70	90	60	0	60	80
15 classes	Cucumber	100	100	100	100	100	100	30	100	100
15 classes	Papaya	100	90	70	80	80	90	30	70	80
15 classes	Potato	70	70	70	70	70	80	20	50	70
15 classes	Pumpkin	90	90	80	100	90	80	30	70	90
15 classes	Radish	80	70	70	80	70	70	40	70	70
15 classes	Tomatoes	50	50	60	40	40	40	40	60	60
Total Average (%)		**78**	**80**	**74.7**	**78**	**80**	**74**	**35.3**	**73.3**	**76**

Table 5.8: Correct Detection Rate for each Color Space using three Channels of the Space separately and Apply SURF Algorithm

VEGETABLE CLASSES		**COLOR SPACES**								
Training Images	Testing Images	**R**	**G**	**B**	**X**	**Y**	**Z**	**H**	**S**	**V**
15 classes	Beans	90	100	90	90	90	80	60	100	90
15 classes	Bitter Gourd	90	100	90	90	100	100	60	90	90
15 classes	Bottle Gourd	100	100	100	100	100	100	90	100	100
15 classes	Bringal	90	90	90	90	90	90	80	90	90
15 classes	Broccoli	100	100	90	80	90	70	40	100	100
15 classes	Cabbagge	40	50	50	40	50	60	40	30	50
15 classes	Capsicum Assorted	90	70	70	90	90	90	80	70	90
15 classes	Capsicum Mixed	60	50	50	60	60	60	70	90	50
15 classes	Carrot	70	60	60	60	60	60	60	70	60
15 classes	Cucumber	90	100	100	90	100	90	60	100	100
15 classes	Papaya	90	80	80	90	80	90	70	80	80
15 classes	Potato	80	90	80	80	80	60	70	80	90
15 classes	Pumpkin	100	100	100	100	100	100	100	70	100
15 classes	Radish	70	80	80	80	80	80	90	70	70
15 classes	Tomatoes	80	80	80	70	80	80	70	80	90
Total Average (%)		**82.7**	**83.3**	**80.7**	**80.7**	**83.3**	**80.7**	**69.3**	**81.3**	**83.3**

Table 5.9 and Table 5.10 show the evaluation of SURF and SIFT results upon varying number of testing images. The experimental results revealed that the categorization accuracy is dependent on the degree of differentiation between vegetable types.

Table 5.9: Evaluation of SURF Results upon Varying Number of Testing images.

VEGETABLE CLASSES		**ACCURACY (%)**
Training Images	Testing Images	Correct Recognition
15 classes (150)	2,4,3- (30 images)	93.3%
15 classes (150)	13 – (10 images)	100%
15 classes (150)	12,15,14,13–(40 images)	87.5%
15 classes (150)	11,10 –(20 images)	95%
15 classes (150)	3,4,5,6,7,8,9,10-(80 images)	81.3%
15 classes (150)	15,14,13,12,11,9,10,7,6,8,4,3,2,5-(140 images)	84.3%
15 classes (150)	3,6,5,4,2,1- (60 images)	86.7%
TOTAL AVERAGE (%)		**89.7%**

Table 5.10: Evaluation of SIFT Results upon Varying Number of Testing images.

VEGETABLE CLASSES		**ACCURACY (%)**
Training Images	Testing Images	Correct Recognition
15 classes (150)	2,4,3- (30 images)	96.7%
15 classes (150)	13 – (10 images)	90%
15 classes (150)	12,15,14,13–(40 images)	67.5%
15 classes (150)	11,10 –(20 images)	95%
15 classes (150)	3,4,5,6,7,8,9,10- (80 images)	82.5%
15 classes (150)	15,14,13,12,11,9,10,7,6,8,4,3, 2,5-(140 images)	79.3%
15 classes (150)	3,6,5,4,2,1- (60 images)	80%
TOTAL AVERAGE (%)		**84.4%**

5.4 Comparisons with the State-of-the-art

This study is conducted to find out the highest accuracy for vegetable image classification using two approaches, the accuracy of the model was tested by using both SIFT and SURF algorithms. The results obtained from classes Bottle Gourd, Broccoli, Cucumber, Pumpkin and Bringal, was 100% proving the performance of the model. As in Table 5.11, the method produced better results than other previous studies. The initial experiment is done with 15 types of common vegetables that are found throughout the world. The vegetables that are chosen for the experimentation are- bean, bitter gourd, bottle gourd, brinjal, broccoli, cabbage, capsicum, carrot, cauliflower, cucumber, papaya, potato, pumpkin, radish and tomato A total of 300 images from 15 classes are used where each class contains 20 images of size 224×224 and in *.jpg format.

With SIFT and SURF approaches used for training different number of vegetable images data set, the experimental results showed that the classification accuracy increased as the number of data set decreases. The experimental verification indicated that the accuracy rate of SURF approach in the test data set reached as high as 100%, which was greatly improved compared with Custom design CNN (97.5%), Caffe's AlexNet model (92.1%), BP neural network (78%) and SVM (80.5%), VGG-M (95.8%), VGG-M-BN (96.5%) and 21000(99.69%), 3000(99.40%), 3000(99.33%) Deep feature extraction (SqueezeNet), ReliefF, LDA methods.

Table 5.11: Comparisons with the state-of-the-art for vegetable classification

<table>
<tr><th>Author(s)</th><th>Year</th><th>Method</th><th>Type</th><th>Dataset Information</th><th>Accuracy</th></tr>
<tr><td rowspan="3">Zhu L et al [9]</td><td rowspan="3">2018</td><td>Caffe's AlexNet model</td><td rowspan="2">Vegetable</td><td rowspan="2">24000 images (5 classes)</td><td>92.1%</td></tr>
<tr><td>BP neural network and</td><td>78%</td></tr>
<tr><td>SVM</td><td></td><td></td><td>80.5%</td></tr>
<tr><td rowspan="2">Li Z et al [15]</td><td rowspan="2">2020</td><td>VGG</td><td rowspan="2">Vegetable</td><td rowspan="2">10 classes and 12000 images</td><td>96.5%'</td></tr>
<tr><td>VGG-M-BN</td><td>95.8%</td></tr>
<tr><td>M.I .Ahmed et al. [22]</td><td>2021</td><td>Custom design CNN</td><td>Vegetable</td><td>15 class and 21000 images</td><td>97.5%</td></tr>
<tr><td rowspan="3">Bayğın M [21]</td><td rowspan="3">2022</td><td rowspan="3">Deep feature extraction(Squeeze Net), ReliefF, LDA</td><td rowspan="3">Vegetable</td><td>21000 images</td><td>99.69%</td></tr>
<tr><td>3000 images</td><td>99.40%</td></tr>
<tr><td>3000 images and 15 classes</td><td>99.33%</td></tr>
<tr><td rowspan="4">This study</td><td rowspan="4">2023</td><td rowspan="2">SIFT</td><td rowspan="2">Vegetable</td><td>Bottle Gourd, Bringal, Cucumber</td><td>100%</td></tr>
<tr><td>300 images and 15 classes</td><td>80%</td></tr>
<tr><td rowspan="2">SURF</td><td rowspan="2">Vegetable</td><td>Bottle Gourd, Broccoli, Cucumber, Pumpkin</td><td>100%</td></tr>
<tr><td>300 images and 15 classes</td><td>84.7%</td></tr>
</table>

5.5 Discussion on Experimental Results

The experimental results indicate that SURF algorithm method achieved the highest accuracies in classifying vegetable classes and had the least amount of time spent for feature matching.

From the results presented above, SIFT and SURF algorithms performed relatively better on different vegetable classes selected from the Kaggle Vegetable Image Dataset. The physical features of the vegetables, such as color and texture, have influence on the vegetable recognition. This effect is seen clearly in Table 5.6 and 5.7 where both color and texture features were extracted using color space channels, SIFT and SURF algorithms respectively. Higher recognition rates are obtained on vegetable classes Bottle Gourd, Cucumber, Bringal, Broccoli and Pumpkin as 100% recognition rate is achieved. The algorithms' performance also varies within the same dataset for varying number of testing images. For a more robust system, less testing images is used to improve the whole systems' performance as the system correctly recognises the vegetable classes. This improved the reliability of the system significantly as shown in the above tables.

Chapter 6

CONCLUSION

This study provided techniques for automatically identifying and categorizing vegetable images. The methodology employs two descriptors, namely SIFT algorithm and SURF algorithm, for vegetable classification. Furthermore, color information obtained from the extracted red-green-blue (RGB) color channels of the vegetable images during training process is used for accurately detecting the color of the vegetable images. Color features are extracted from the vegetable images using color space channels. In this case, the images are read and immediately converted into XYZ and HSV color space and for each X, Y, Z and H, S, V color channels SIFT and SURF methods can be utilized once the color vegetable features for each image have been generated. The experiments have been evaluated using 300 images of 15 different classes from Kaggle Vegetable Image Dataset. The experimental results revealed that the categorization accuracy is dependent on the degree of differentiation between vegetable types. In SURF, the maximum level of accurately identified vegetables is attained, with Bottle Gourd (100%), Cucumber (100%) Broccoli (100%), Pumpkin (100%) correctly recognised with a total average accuracy of 84.7% for the entire system and with SIFT, 80% accuracy was achieved from training the entire system. The results also revealed that HSV color space channel is relatively weaker than XYZ in recognition. RGB color space outperformed the other color space channels with (G) color channel having the highest recognition rate of 83.3% in SURF and 80% in SIFT respectively. In this thesis, two feature extractors are evaluated, and the experimental findings indicate that the SURF algorithm performs better than SIFT in terms of properly identifying vegetable images and computation time. SIFT is expected to achieve better accuracy, however since the number of images used are small, SURF achieves better results and faster computation. Alternative feature extraction such as shape and size can augment the feature set to improve classification results However, incorporation of these features may slow down the recognition time, new feature extraction methods might be utilized in future studies to enhance the effectiveness of vegetable classification.

REFERENCES

[1] Bolle, Ruud M., et al. "Veggievision: A produce recognition system." *Proceedings Third IEEE Workshop on Applications of Computer Vision.* WACV'96. IEEE, 1996.

[2] Seng, Woo Chaw, and Seyed Hadi Mirisaee. "A new method for fruits recognition system." *2009 International conference on electrical engineering and informatics.* Vol. 1. IEEE, 2009.

[3] Rocha, Anderson, et al. "Automatic fruit and vegetable classification from images." *Computers and Electronics in Agriculture* 70.1 (2010): 96-104.

[4] Aibinu, Abiodun Musa, et al. "Automatic fruits identification system using hybrid technique." *2011 Sixth IEEE International Symposium on Electronic Design, Test and Application.* IEEE, 2011.

[5] Rocha, Anderson, et al. "Automatic produce classification from images using color, texture and appearance cues." *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing.* IEEE, 2008.

[6] Heidemann, Gunther. "Unsupervised image categorization." *Image and Vision computing* 23.10 (2005): 861-876.

[7] Zawbaa, Hossam M., et al. "Automatic fruit image recognition system based on shape and color features." *Advanced Machine Learning Technologies and*

Applications: Second International Conference, AMLTA 2014, Cairo, Egypt, November 28-30, 2014. Proceedings 2. Springer International Publishing, 2014.

[8] Arivazhagan, Shebiah, et al. "Fruit recognition using color and texture features." *Journal of Emerging Trends in Computing and Information Sciences* 1.2 (2010): 90-94.

[9] Zhu, Ling, et al. "High performance vegetable classification from images based on alexnet deep learning model." *International Journal of Agricultural and Biological Engineering* 11.4 (2018): 217-223.

[10] Jana, Susovan, Saikat Basak, and Ranjan Parekh. "Automatic fruit recognition from natural images using color and texture features." *2017 Devices for Integrated Circuit (DevIC)*. IEEE, 2017.

[11] Sakai, Yuki, et al. "A vegetable category recognition system using deep neural network." *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS).* IEEE, 2016.

[12] Zeng, Guoxiang. "Fruit and vegetables classification system using image saliency and convolutional neural network." *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC).* IEEE, 2017.

[13] Femling, Frida, Adam Olsson, and Fernando Alonso-Fernandez. "Fruit and vegetable identification using machine learning for retail applications." *2018*

14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE, 2018.

[14] Kuang, Hulin, et al. "Multi-class fruit detection based on image region selection and improved object proposals." *Neurocomputing* 283 (2018): 241-255.

[15] Li, Zhenbo, et al. "Vegetable recognition and classification based on improved VGG deep learning network model." *International Journal of Computational Intelligence Systems* 13.1 (2020): 559-564.

[16] Joseph, Jinu Lilly, Veena A. Kumar, and Santhosh P. Mathew. "Fruit classification using deep learning." *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2021.* Springer Singapore, 2021.

[17] Yuhui, Zhan, et al. "An automatic recognition method of fruits and vegetables based on depthwise separable convolution neural network." *Journal of Physics: Conference Series.* Vol. 1871. No. 1. IOP Publishing, 2021.

[18] Bhavya, J. K., et al. "The literature survey on intra class fruits and vegetable recognition system using deep learning.", *International Research Journal of Modernization in Engineering Technology and Science,* Vol. 3, Issue 7, 2021.

[19] Latha, R. S., et al. "Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network." *2021 International Conference on Computer Communication and Informatics (ICCCI).* IEEE, 2021.

[20] Yuesheng, Fu, et al. "Circular fruit and vegetable classification based on optimized GoogLeNet." *IEEE Access* 9 (2021): 113599-113611.

[21] Bayğin, Mehmet. "Vegetable and Fruit Image Classification with SqueezeNet based Deep Feature Generator." *Turkish Journal of Science and Technology* 17.1 (2022): 121-134.

[22] Ahmed, M. Israk, Shahriyar Mahmud Mamun, and Asif Uz Zaman Asif. "DCNN-based vegetable image classification using transfer learning: A comparative study." *2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP).* IEEE, 2021.

[23] Om, Patil, and Gaikwad Vijay. "Classification of vegetables using TensorFlow. " *International Journal for Research in Applied Science and Engineering Technology* 6.4 (2018): 2926-2934.

[24] Garg, Meenakshi, and Gaurav Dhiman. "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants." *Neural Computing and Applications* 33 (2021): 1311-1328.

[25] Lowe, David G. "Distinctive image features from scale-invariant keypoints. " *International journal of computer vision* 60 (2004): 91-110.

[26] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." *Lecture notes in computer science* 3951 (2006): 404-417.

[27] https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset