2021-10

DEVELOPING DEFECT INSPECTION MODEL OF HARICOT BEANS USING COMPUTER VISION AND MACHINE LEARNING APPROACH

MELKAM, WALIE BALEW

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

MSc Thesis on:

DEVELOPING DEFECT INSPECTION MODEL OF HARICOT BEANS USING COMPUTER VISION AND MACHINE LEARNING APPROACH

BY:

MELKAM WALIE BALEW

October, 2021

Bahir Dar, Ethiopia

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
COMPUTING FACULTY

MSc Thesis on:

DEVELOPING DEFECT INSPECTION MODEL OF HARICOT BEANS USING COMPUTER VISION AND MACHINE LEARNING APPROACH

BY

Melkam Walie Balew

A thesis submitted to

The school of Graduate Studies of Bahir Dar Institute of Technology Bahir Dar University in partial fulfillment of the requirements for The Degree of Master of Science in **Information Technology**

Advisor: Abrham Debasu (Assistance Processor)

Bahir Dar, Ethiopia
October 2021

DECLARATION

This is to certify that the thesis entitled "**Developing Defect Inspection Model of Haricot Beans using Computer Vision and Machine Learning Approach"**, submitted in partial fulfillment of the requirements for the degree of Master of Science in **Information Technology** under **Computing Faculty**, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Melkam Walie		01/02/2014 E.C
Name of the candidate	Signature	Date

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING

Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student **Melkam walie Balew** Signature ________ Date 01/02/2014

As members of the board of examiners, we examined this thesis entitled "**DEVELOPING DEFECT INSPECTION MODEL OF HARICOT BEANS USING COMPUTER VISION AND MACHINE LEARNING APPROACH.**" by **Melkam walie.** We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of masters of Science in "Information Technology."

Board of Examiners

Name of Advisor Signature Date

Mr. Abrham Debasu (Asst. Prof) ________ 11/10/2021

Name of External examiner Signature Date

Dr.Adane Letta ________ ________

Name of Internal Examiner Signature Date

Dr. Abdulkerim Mohammed ________ 12/10/2021

Name of Chairperson Signature Date

Dr. Gebeyehu Belay ________ 11/10/2021

Name of Chair Holder Signature Date

Derejaw L ________ 11/10/2021

Name of Faculty Dean Signature Date

Asegahegn E. ________ Oct 11, 2021

በባሀር ዳር ዩኒቨርሲቲ ባሀር ዳር ቴክኖሎጂ ኢንስቲትዩት የኮምፒውቲንግ ፋካልቲ
Bahir Dar University Bahir Dar Institute Of Technology Faculty Of Computing

Faculty Stamp

Acknowledgment

I would like to express my deep gratitude to my master thesis advisor, Abrham Debasu. He spends vary much time instructing me how to write a paper, how to search literature and how to collect data. I am also grateful to Mrs. Fasika Hailu and friends for spending time read this thesis and providing useful suggestions.

ABSTRACT

Haricot bean is a very important legume growing worldwide with higher market demands. It is a very important foreign exchange earning crop in Ethiopia. It has been observed that the markets of this crop are gradually increasing. In Ethiopia, the defect inspections are performed by experts manually. It is labour-intensive, time-consuming, and suffers from the problem of inconsistency and inaccuracy. In this study, we develop a model to inspect defect detection of haricot beans using computer vision and machine learning approaches. The required images of haricot beans were captured from Bure Ethiopian Commodity exchange (ECX) centre in the Amhara region of Ethiopia. 1000 for each class defect and non-defect haricot bean were taken. The images were taken directly using smart phone by placing on the white paper. After image acquisition pre-processing had been used to get an enhanced image. For feature extraction the grey level co-occurrence matrix (GLCM) and the convolutional neural network (CNN) method had been considered. Besides, for classification three classifiers random forest (RF) , support vector machine (SVM) and end to end CNN were applied to classify to their predefined class. For developing a prototype and conducting experiment, Python programming language was used in this study.

In this study, three groups of experiment have been conducted (CNN features with SVM, RF; GLCM features with SVM, RF; finally and end to end CNN). From the experiment, the result revealed that the CNN method for feature extraction achieved an accuracy of 94% and 97% using SVM and RF classifiers, respectively. Further, using GLCM textural features methods were showed an accuracy of 88% and 97% for SVM, and RF classifiers, respectively. When using CNN as a classifier, an accuracy of 99% was achieved. It was concluded that, in all applied approaches, the model can identify defects and non-defect haricot bean with the highest accuracy. It is recommended that the developed approach should be implemented to other types of haricot beans, such as white beans and speckled beans.

Key words: *GLCM, CNN, SVM, RF, haricot* bean.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ANN: - Artificial Neural Network

CNN: - Convolutional Neural network

CSA: - Central statics Agency

COMESA: Common Market for Eastern and Southern Africa

DHB: - Defect haricot bean

DIP: - Digital Image Processing

ECX: - Ethiopian Commodity Exchange

EIAR:- Ethiopian Institute of Agricultural Research

GLCM: - Gray level Co-occurrence Matrix

KNN: - K-nearest Neighbor Network

NDHB- Non-defect haricot bean

RF: - Random Forest

SVM: - Support Vector Machine

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Background of the Study

The local name of haricot bean is ‘Boleqe’ also known as common bean, kidney bean and field bean is a very important legume crop growing worldwide. Haricot bean is the most economically important pulse crop grown in Ethiopia (Kebede et al., 2018). In Ethiopia haricot bean considered as the main cash crop and protein source of especially in lowlands and mid-altitude zones.

Ethiopia is one of the top 12 producers of total plus in the world market, third largest producer of haricot bean in COMESA member countries and the leading exporter in Africa (Kebede et al., 2018).

Ethiopia has been exporting haricot bean for more than 50 years and have been grown as food crop for longer period in the low and mid land altitude areas of the region (Epherem , 2016).The major haricot was producing regions in Ethiopia are Oromia, Southern Nations, Nationalities, and Peoples' Region (SNNPR), Amhara and Benshangul Gumuz which contribute more than 99% of total haricot bean output. In 2014/15 report showed that in agricultural production, Oromia took the lion share (51%) of haricot bean production in the country, followed by SNNPR (27%), Amhara (20%), Benishangul-Gumuz1.4% and the other regions contributing the remaining to the country total production (Epherem, 2016).

There are variety types of haricot bean grown in Ethiopia, including the mottled, red, white and black varieties. The pure red and pure white haricot bean are very important foreign exchange earnings in Ethiopia (Ferris & Kaganzi, 2008). On the other hand, lack of quality control and grading systems of the country did not get enough revenue from the product. This research focused on the two most economically important bean types, namely pure white and pure red beans.

Ethiopian commodity exchange (ECX) has established warehouses in the major coffee, sesame and white haricot bean marketing centers, including Awassa, Dilla, Soddo, Bonga, Jimma, Gimbi and Bedele for coffee, and Adama, Shashemene, Humera, Metema and Bure for sesame and haricot beans. These centers provide quality inspection, grading and warehouse services (Dawit & Gerdien, 2010).

The defect inspection of haricot beans is performed by human experts in the warehouse (WFP, 2011). This manual sorting of defect haricot bean by visual inspection is labor intensive, time-consuming and suffers from the problem of inconsistency and inaccuracy in judgment by a different human. Therefore, developing defect detection of haricot beans model is desired to push up the value of haricot bean products and to improve international competitiveness. The use of manual technique for food quality evaluation is labor intensive, time consuming and often subjective (Mauricio et al., 2019). The employment of different types of machine learning algorithms, for food quality evaluation have showed that promising results (Momin et al., 2017).

This research work presents a computer vision with machine learning algorithms which identifies the defected and non defected haricot beans. CNN and GLCM methods are used for feature extraction. SVM and RF classifiers are used to deal with Ethiopian haricot beans to inspect defected and non-defected types.

1.2. Statements of the Problem

Defect inspection of agricultural products helps to evaluate and determine their quality. This promotes their market. In the past years, manual inspections have had many problems in maintaining consistency and ensuring satisfactory detection efficiency (Tian et al., 2020). For example, ECX quality controls and grading of haricot bean is working in traditional way. On the other hand, haricot bean is a very important pulse in the export of Ethiopia as large amount is exported around the world market (Epherem, 2016).

The growth of computer vision technology, the automatic grading and defect inspection of grains has been achieved, and computer vision systems have been widely used in different fields of the agricultural and food production market segments, avoiding the high cost and low efficiency of traditional operations (Tian et al., 2020).

Despite increasing demand for quality haricot bean on the world export market, the existence of manual inspection of defect detection systems of haricot bean do not provide a good selection method of quality beans. Besides, this manual sorting of defect haricot bean by visual inspection is labor intensive, time-consuming and suffers from the problem of inconsistency and inaccuracy in judgment by a different human (Mauricio et al., 2019).

To overcome this, the use of modern technology is very important. Within this, this research work aims to develop defect inspection model of haricot beans using computer vision and machine learning algorithms.

1.2.1 Research Questions

To this end, this study attempts to explore and answer the following research questions

- ✓ Which feature extraction technique provides the highest performance?
- ✓ Which machine learning model provides the highest performance?

1.3. Objectives

1.3.1. General Objective

The general objective of this study is to develop defect inspection model of haricot beans using computer vision and machine learning approaches.

1.3.2. Specific Objectives

- ❖ To build a classification model for haricot bean defect and non-defect detection.
- ❖ To evaluate the performance of the proposed model using different evaluation metrics.
- ❖ To test the effect of feature extraction on defect detection model performance.
- ❖ To compare the results and performances of the machine learning techniques used.

1.4. Research Methodology

The computer vision and machine learning-based approach for defect detect model of haricot bean. To do this Undertaking experimental research involves dataset preparation, implementation and performance evaluation. The detail methods and tools that are used in this research work are described in the methodology chapter.

1.4.1. Literature Review

The literature review part gather and find relevant documents to indicate the solution of the problems based on the context of the previous literature. In this regard, documents related to computer vision and machine learning as well as related to defect detection of crops, pulse and beans image preprocessing and feature extraction are explored to organize relevant information.

1.4.2. Data Collection

Haricot bean is produced in different parts of the country. There are about two haricot bean delivery sites in Amhara region, namely Bure and Kombolcha. ECX has its own experts for sampling the haricot bean for defect inspection. The sample data for this research work was collected from Bure branch due to administrative cost and transportation problem.

1.4.3. Model Design and Experimentation

The computer vision and machine learning based model used in this research starts by capturing images of haricot beans, followed by preprocessing the image using different techniques and extracted features of the haricot bean images for the intended analyses. Both CNN and GLCM used for feature extraction. For this research work Python programming language used for image preprocessing and build the model of defect inspection. Finally we had experimentation undertaken for the evaluation of different measurement matrices.

1.5. Scope and Limitation of the Study

This research work is only restricted to inspect defect detection of haricot bean based on defect and non-defect type using texture characteristics.

1.6. Significance of the Study

To develop defects inspection of haricot beans model using machine learning, mainly, benefit haricot bean producer and exporters. This research work can show different machine learning algorithms for defect detection for haricot beans.

In general, the contributions of the research are:-

Scientific Contribution: - the aim of the research is to develop a defect inspection model using computer vision and machine learning approach, we had used a particular feature extraction and classification technique that identify haricot bean is defective or not. Besides, there is no other research work for Ethiopian haricot bean defect detection models that could yield the highest performance; the research is the startup reference for future researchers and scholars about the technique and methods for defect inspection in agricultural products, especially legumes.

Methodological Contribution: - The research plays a great role in understanding the steps and challenges of haricot bean defect detection starting from images preprocessing and feature extraction through CNN features and GLCM texture feature extraction with the classification of machine learning approaches.

Practical Contribution: - Determining defect detection for agricultural products has been an issue to compute the global market. Different researchers have attempted to automate it using various agricultural production using different techniques. The research has vital contribution in the process of defect inspection on haricot bean by maintaining consistency, ensuring satisfactory and efficiency when using technology rather than manual work in human labor.

1.7. Organization of the Document

The thesis is organized into five chapters including this one. Chapter two, deals with literature review. In Chapter three, methodology of the thesis is described. In Chapter four, the experiment results and discussions presented. In Chapter Five, conclusions and future work presented.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

In this chapter, detailed description of haricot bean, applying computer vision system for defect detection using image processing techniques, feature extraction algorithms, and classification models are discussed by exploring general literature reviews.

2. 2. General about Haricot Beans

Haricot bean (*Phaseolus Vulagris* L), locally known as 'Boleqe' also known as dry bean, common bean, kidney bean and field bean is a very important legume crop grown worldwide.
Haricot bean is the most economically important pulse crop grown in Ethiopia (Wondimu & Bogale, 2017). Haricot bean is considered as the main cash crop and protein source of many lowlands and mid-altitude zones of Ethiopia (Kabata et al., 2016)

Ethiopia exports haricot beans second to faba bean in the global market. Common beans are grown throughout the country and are an increasingly important commodity in the cropping systems of smallholder producers for food security and income (Ferris & Kaganzi, 2008). Farmers grow a wide range of bean types, in terms of color and size, but the most common types are the pure red and pure white haricot beans. The red beans are preferred by rural consumers. There are a wide range of reds, including red mottled varieties that are produced and sold in the rural markets. White haricot beans are sold almost exclusively for the export markets (Ferris & Kaganzi, 2008).

To support both the growth in domestic and export bean markets, the Ethiopian Institute of Agricultural Research (EIAR) has developed a range of high-yielding, multi-disease resistant bean varieties. The focus of this genetic improvement program has been on the pure red and white beans to support the commercial sector (Miklas et al., 2006).

2.3 Defect Inspection Parameters of Haricot Beans

The defect inspection of Ethiopian haricot beans is conducted manually following a set of standards and procedures of ECX. This can be briefly described as follows: a sample of 3kg per 10 tons of a truck, which is an average carrying capacity of a truck, and the 3kg haricot beans are divided into three equal sub-samples, which is intended for visual inspection and others. As requirement a quality haricot bean have a good natural color, free from objectionable odor, contain no damage and other insect, free from foreign matters (all matters other than haricot bean), free from toxic seeds and have a maximum of 13% moisture content. Damage beans that includes heat damage, germ damage, mold damage, split, cracked seed coat, sprouted insect immature, shriveled and broken kernel (WFP, 2011).

Foreign Matter ;- All mater other than haricot beans (organic and inorganic); leaves pods twigs, earth , sand, dust, stones , other crops seeds, detached seed coats and the likes.

Defects: - the sum of damage haricot beans, split and haricot bean without coat and cracked seed coat.

All the about terms categorized into four Parameters

a. Foreign Matters contain
b. DSW (Damage, shriveled and Weevil) contain
c. Insect board beans
d. Contrasting color (Fade Color)

2.4. Computer Vision Techniques

Computer vision uses a camera and computer instead of the human eye to identify, track and measure targets for further image processing. Computer vision systems used for classify and select food products. It extracts and analyze useful information from a specific object present in an image (Mauricio et al., 2019).

The defect inspection for agricultural products used to judge and determine the quality of the products to promote to the market (Gongal et al., 2015). The development of computer vision technology, the automatic grading and defect inspection of agricultural products were achieved, and computer vision systems have been broadly used in diverse fields of the agricultural and food production market segments, avoiding the high cost and low efficiency of manual operations (Tian et al., 2020).

Image Acquisition

Image Acquisition is the transferring of electronic signals from a sensor to a numerical representation by a device like a camera, scanner (Mauricio et al., 2019). The initial step in developing any machine vision system is the selection of appropriate image acquisition technique. Images are used for acquiring information, the aims of technology is to duplicate the role of human vision by electronically perceiving and understanding an image (Mahajan et al., 2015). Machine vision system equipped the computations by which useful information about an object can be automatically extracted and analyzed from an acquired image.

2.4.1. Image Preprocessing

Image preprocessing is performing to manipulate digital images for the purpose of improving their quality, reducing noise or correcting lighting problems. In addition to this image analysis refers to the process to differentiate regions of interest from other regions to extract information (Inácio & Rieder, 2018).

During preprocessing, systematic degradation cut off by brightness improvement is performed. Similar to brightness transform, the grayscale transformation is independent of the location of the pixel in the image. The histogram equalization allocates the brightness levels uniformly all over the brightness scale (Dighe, 2014). Besides, the geometric transform allows the exclusion of geometric distortion that arises when an image is taken probably from a different position. The geometric transform used to map the pixel to a new position. The intensity is usually estimated as an interpolation of the brightness of certain points in the neighborhood (Mohamed et al., 2018).

1. **Image Resizing**

Image resizing is preserving an important region of an image, minimizing distortions, and improving efficiency (Dighe, 2014). Image Resizing can be more effectively achieved with a better understanding of image contents. Image resizing has been a promising subject of image processing and computer vision (Dighe, 2014).

2. **Noise Removal**

Digital images are exposed to a variety of types of noise. Noise is the effect of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real picture (Mohamed et al., 2018). Filters are required for removing noises before processing.

Gaussian Filters: - to remove certain types of noise. Gaussian filters tend to blur sharp edges, destroy lines and other fine image details, each pixel to the average value, or a weighted average, of itself and its nearby neighbors; the Gaussian filter is just one possible set of weights .Gaussian filters useful for removing grain noise from a image. Gaussian filters are often used as the basis for nonlinear noise reduction filters (Hambal et al., 2017).

An Averaging Filter is helpful for removing grain noise from a picture. Hence each pixel put to the average of the pixels in its neighborhood (Mohamed et al., 2018).

Median Filtering is similar to an averaging filter, in that each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. On the other hand, median filtering, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is very less sensitive than the mean to extreme values of outlier (Mohamed et al., 2018). The effect of noise removal has a strong influence on the quality of the image processing techniques.

3. Contract Enhancement

Contrast enhancement is a process by which the pixel intensity of the image is changed to utilize the maximum possible bins .In general the “contrast” term refers to the separation of dark and bright areas present in an image (Bora, 2017). The advantage of contrast enhancement to removes the ambiguity that may arise between different regions in an image.

Contrast enhancement can be classified into two categories: Local contrast enhancement and Global contrast enhancements.

Global Contrast Enhancements: - it contrasts enhancement techniques, the global histogram information is considered for enhancement. As the whole image is considered at once, so, local information is ignored in this case. The main advantage of global contrast enhancement is that it is computationally simple and is suitable for overall enhancement of the image (Bora, 2017). The two examples of global contrast enhancement techniques are Histogram Equalization and Histogram Specification.

Histogram Equalization is a nonlinear technique for adjusting the contrast of an image using its histogram. Conventional contrast enhancement techniques frequently fail to produce satisfactory results for low-contrast images and cannot be automatically applied to different images because processing parameters must be specified manually to produce satisfactory results for a given image (Pradeep & Gnanapriya, 2016)

Local Contrast Enhancement: -it improves the local future brightness. So contrast ratio can be improved in every region of the image. Local contrast enhancement technique is computationally complex (Bora, 2017). It involves high cost due to consideration of over-lapped sub-blocks.
The two examples of local contrast enhancement techniques that are frequently adopted are AHE and CLAHE.

4. Color Space

A color space is a model for representing color in terms of intensity values. It is a one-to-four dimensional space. A color dimensional space one dimension per pixel represents the gray scale space (Mohamed et al., 2018).

Color Space Conversion: - converting the color space of the image in which it can be more precisely represented for extracting features of interest in an image. Some examples of color space are cieLAB, HSV, HSL (Pradeep & Gnanapriya, 2016)
Initially the images are converted from RGB to HSV color space where enhancement is achieved and reconverted to the RGB. Class Limited Adaptive Histogram Equalization is used to enhance the luminance component. Discrete Wavelet Transform is used for the Saturation components, and the decomposed approximation coefficients are modified by a mapping function derived from scaling triangle transform. Inverse Wavelet transform is applied to obtain the enhanced S component. The image is then converted back to the RGB color space (Pradeep & Gnanapriya, 2016).

5. Segmentation

Image segmentation can be defined as dividing an image into different groups or partitions based on some homogeneity criteria like color, intensity, or texture (Bora, 2017). There are different types of segmentation techniques.

Regional Growth Segmentation:- this method is a typical serial region segmentation algorithm, and its basic idea is to have similar properties of the pixels together to form a region (Yuheng, 2017) The advantage of regional growth segmentation is that it separates the connected regions with the same characteristics and provides good boundary information and segmentation results

(Yuheng, 2017). The initiative of regional growth is simple and requires only a few seed points to complete. And the growth criteria in the growing process can be freely specified. Finally, it can pick multiple criteria at the same time. The limitation is the computational cost large and also the noise and grayscale unevenness can lead to voids and over-division and the shadow effect on the image is often not very good.

Edge-Based Segmentation: - this technique detected edges in an image represents object boundaries, and are used to identify these objects. A gray edge between two adjacent regions with different gray values in the image, in this case the gray value is not continuous. This discontinuity can often be detected using derivative operations, and derivatives can be calculated using differential operators (Yuheng, 2017). Sobel and canny edge detection algorithms are some of the examples of edge-based segmentation techniques.

Thresholding Segmentation: - This is the simplest method of image segmentation where each pixel value is compared with the threshold value. If the pixel value is smaller than the threshold
It is a common segmentation algorithm that directly divides the image grayscale information processing based on the gray value of different targets. Threshold segmentation can be divided into local threshold methods and global threshold methods. The most commonly used threshold segmentation algorithm is the largest interclass variance method (Halwa et al., 2013), which selects a globally optimal threshold by maximizing the variance between classes. The advantage of the threshold method is that the calculation is simple and the operation speed is faster (Halwa et al., 2013). Especially when the target and the background have high contrast, the segmentation effect can be obtained. The drawback is that it is difficult to obtain accurate results for image segmentation problems where there is no significant grayscale difference or a large overlap of the grayscale values in the image (Halwa et al., 2013).

2.4.2. Feature Extraction

Feature extraction is the process of extracting certain characteristic features and generating a set of meaningful descriptors from an image. Feature extraction is a form of dimensionality reduction and efficiently represents the major attributes that are useful for the effective classification of each class (Öztürk & Akdemir, 2018). Transforming the input data haricot bean image into a set of features is called feature extraction. The purpose of the feature extraction stage is to extract various features from a given haricot bean images which best characterizes a given haricot bean. Features are the information or list of numbers that are extracted from an image. These are real-valued numbers (integers, float or binary). There are a wider range of feature extraction algorithms in Computer Vision. When deciding about the features that could quantify haricot bean quality, could possibly think of Color, Texture and Shape as the primary ones. The feature extraction is an important part of classifier because it affects working of classifier (Yogesh et al., 2020).

1. Gray level Co-occurrence Matrix (GLCM)

GLCM is a popular texture-based feature extraction method. The GLCM determines the textural relationship between pixels by performing an operation according to the second-order statistics in the images (Öztürk & Akdemir, 2018). The GLCM determines the frequency of combinations of these pixel brightness values determined. That is, it represents the frequency formation of the pixel pairs. The GLCM properties of an image are expressed as a matrix with the same number of rows and columns as the gray values in the image. The elements of this matrix depend on the frequency of the two specified pixels. Both pixel pairs can vary depending on their neighborhood. These matrix elements contain the second-order statistical probability values depending on the gray value of the rows and columns. If the intensity values are wide, the transient matrix is quite large. This creates a time-consuming process load.

GLCM points to Gray level Co-occurrence matrix is of 2nd order statistics, so information with regards to pixels of pairs are collected by GLCM.GLCM exhibits how the pixel brightness in an image occurs. A matrix is built up at a distance of d=1 and at angles in degrees 0^0, 45^0, 90^0, 1350. Haralick also offered different measures i.e. entropy, energy, contrast, correlation, dissimilarity, homogeneity etc. These dimensions calculate at different angles(Sharma et al., 2015).

Second order measures define the relationship between groups of two (usually neighbouring) pixels in the original image (Sharma et al., 2015). GLCM texture works by picks up the relation between two pixels at a time; these are the reference and the neighbour pixel. GLCM is prepared from gray scale values. It is taken into account how often a pixel with gray level (gray scale intensity or gray tone) values come either horizontally, vertically and diagonally to leveled the pixels with the value j. GLCM directions are: Horizontal(0) ,Vertical(90) ,Diagonal ,bottom left to top right(-45) ,top left to bottom right (-135) .They are announced as P0, P45, P90 and P135 respectively (Sharma et al., 2015).

2.5. Machine Learning Approach

Machine learning technology have been generally applied in quality determination of agricultural and food products. Different machine learning techniques like Artificial Neural Network ,decision trees, Naïve Bayes, k-means clustering, support vector machines, random forest, k-Nearest Neighbor and soon have been used extensively in agriculture related fields (Saha & Manickavasagan, 2021). Machine learning algorithms had possessed the ability to learn from data without relying on explicit programming.

2.5.1. Lazy Learning

Lazy learning methods are the decision of how to generalize a new query is encountered. When the query instance is received, a set of similar related patterns is retrieved from the available training patterns set and it is used to categorize the new occurrence. For the selection similar patterns, a distance measure is used having nearby point's higher significance (Galván et al., 2009). In general lazy learner work by selecting the k nearest input patterns to the query points, in the base of the Euclidean distance.

1. **K-Nearest Neighbor**

K-Nearest Neighbor used three Distance metrics namely Manhattan distance, Euclidean distance, P norm distance while calculating the distance between the points. The k-NN algorithm searches the training dataset for the k samples that are nearest to the point to be classified. The new data point is assigned to a class label of the new data point through majority vote among its k nearest neighbors. The optimal value of k is used for finding a good balance between under fitting and over fitting. If the value of k is too small, it will be more prone to noise points and if the k value is too large, the neighborhood may comprise of points from other classes. The main advantages of k-NN are that the cost of the learning process is nil. No optimization is required and it's simple to program with high accuracy (Saha & Manickavasagan, 2021). The limitation of k-NN is very prone to overfitting due to the curse of dimensionality. The curse of dimensionality describes in which the feature space tends to become increasingly scattered for a higher number of dimensions for a training dataset of fixed size. In other words, the closest neighbors being relatively far away in a high-dimensional space can give a very good estimate (Saha & Manickavasagan, 2021).

2. Naïve Bayes

Naïve Bayes is a powerful simple generative machine learning classifier that utilizes the concept of conditional probability (Bayes' theorem) to describe the outcome probabilities of related events. The naïve highlights the assumption that each feature is independent and identically distributed than the others indicating that a feature value has no relationship with the value of another feature (Saha & Manickavasagan, 2021). It has been used in many domains with high accuracy. The major limitation of this algorithm is that it is in capable of learning the interaction between two predictor variables or features due to the assumption of conditional independence (Saha & Manickavasagan, 2021).

3. Random Forest

Random forest is a machine learning algorithm which is comprised of n collections of de-correlated decision trees (Kaitlin et al., 2018). It is assembled the idea of bootstrap aggregation, which is a method for resampling with replacement in order to reduce variance. Random Forest uses multiple trees to average (regression) or compute majority votes for classification in the terminal leaf nodes when making a prediction. To build the idea of decision trees, random forest models have resulted in significant improvements in prediction accuracy as compared to a single tree by growing 'n' number of trees. Each tree in the training set is sampled randomly without replacement (Kaitlin et al., 2018). Decision trees have a tree-like structure in the top the node is considered the root of the tree that is recursively split at a series of decision nodes from the root until the terminal node or decision node is reached.

Random forests are considered as a highly accurate and robust method because of the number of decision trees participating in the process. It does not suffer from the overfitting problem (Navlani, 2018). The main reason is that it takes the average of all the predictions, which cancels out the biases. The algorithm can be used in both classification and regression problems.

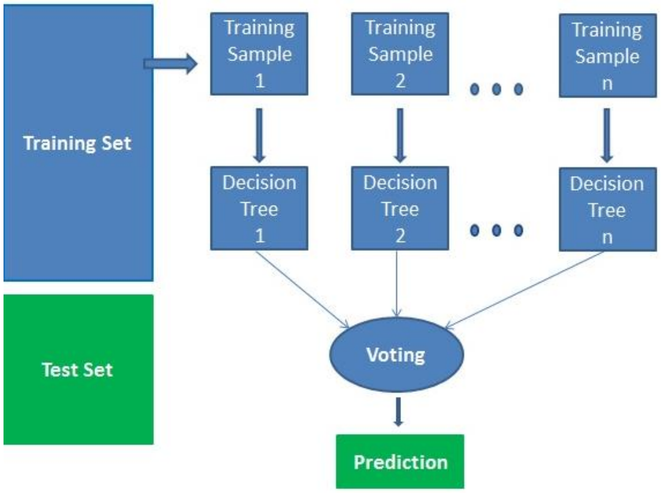


Figure 2.1 Random Forest Classifier (Navlani, 2018)

4. Decision Trees

Decision trees represents like a tree having internal nodes representing a test on a feature, each branch representing the result of a test, and each leaf node representing the class label followed by the execution of decision after considering all the features. Decision tree have three types of nodes Root nodes, internal nodes and leaf nodes. It works different data types such as numeric, ratings, categorical and is also capable of handling missing data in response as well as independent variables. Decision tree is works based on a series of Boolean tests. The working way of a decision tree starts with the greedy algorithm, in which the tree is structured in a top-down iterative divide-and-rule approach. In the initial stage, the root node comprises of the training data set. The input data is divided iteratively based on selected features. The test features at each node are spitted based on decision tree functions like Gini index and entropy. One major advantage of a decision tree is the non-requirement of creation of dummy variables. The limitation of decision tree is the large growth of the tree resulting in one leaf per observation. In addition, it is impossible to reconsider a decision once the training data set have been divided for answering a problem (Saha & Manickavasagan, 2021).

2.5.2. Active Learning

Active learning works on based on the model's capability to distinguish between instances of different classes, the improvement of active learning for one performance measure often came at the expense of another measure. By determining which instances supposed to be annotated, active learning algorithms can reduce the time, effort, and resources needed to train an accurate predictive mode (Ramirez-Loaiza et al., 2017).

1. Feed Forward Neural Network

Feedforward Neural Network has the connections in only one direction and can be represented as a directed acyclic graph. The node receives inputs from the upstream nodes and to delivers the output to the downstream nodes without cycles. Such kind of network represents a function and has no internal state except the link weights. In other hand, a recurring neural network feeds the input nodes with their outputs. In this way, the network forms a dynamic system that may or may not reach a steady-state (Inácio & Rieder, 2018).

2. Back Propagation

The backpropagation algorithm is supervised learning applied with a known dataset of input-target to output samples and has been used in detection and classification applications (Hameed et al., 2016). BP has an input layer, one or more hidden layers, and an output layer. Layers are connected in sequence starting from the input layer through the hidden layers to the output layer. The connections between layers have weights and each layer includes one or more neurons (Tian et al., 2020).

The basic concept of BP is to minimize the overall output error gradually during the learning process. The training sets are estimated iteratively through the input layer to predict the correct output. The BP process is two stages, forward and backward process. In the forward process, the BP is the inputs to the neural network are the weights of interconnections between inputs and hidden layers. The hidden layer is defined by the hidden layer that will pass through the activation function (F) (Hossain & Alam , 2019).

After calculating the overall output by multiplying the output of the hidden layer neurons with the hidden layer weights the results, pass through an activation function called the threshold.

3. Support Vector Machines

Support vector machine is a supervised learning approach. SVM plots the training data into another space higher than the original space and divides the instances belonging to different categories by separating these instances non-linearly and linearly (Hameed et al., 2016). Support vector machine tries to keep the separation boundary between two different categories (classes) as wide as possible (Saha & Manickavasagan, 2021). The perpendicular bisector of the shortest line connecting the two classes is called a hyperplane. The training instances closest to the hyperplane are called support vector machines.

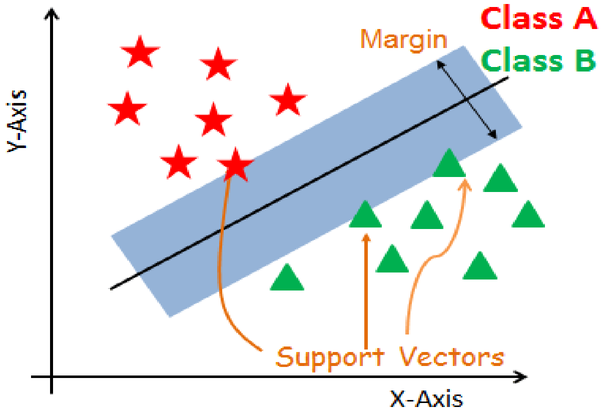


Figure 2.2 SVM for binary classification (**taha , 2021**)

Support Vector Machines aims to obtain the optimal separating points of one class from the rest through a selection of one's passing through the largest gaps possible between points of different classes. The new points are classified to a certain class depending on the side of the surfaces they fall on. There are different kernel functions used in SVM are linear kernel function, radial basis function, polynomial kernel function, and sigmoid kernel functions. The advantage of SVM is very effective while working with high dimensional spaces, which requires learning from several features in the problem (Hasan et al., 2019). SVM has effective when the data is relatively small i.e., a high dimensional space with few points (Saha & Manickavasagan, 2021). In addition, they require less memory storage as a subset of points is used only to represent the boundary surfaces. The limitation SVM models take intensive calculations while the model is trained. Furthermore, they do not quantify the confidence percentage of a prediction which otherwise can be done through k-fold cross-validation with an increased computation cost (Saha & Manickavasagan, 2021)

2.5.3. Deep Learning

Deep Learning Networks have more nodes and more complex means of layer interconnection, they require strong computational power for their training and they have automatic extraction of the parameters (Inácio & Rieder, 2018).

Convolutional neural network is the first type the deep learning algorithms. CNN also the best learning algorithms for understanding image content and have shown very good performance in image segmentation, classification, detection, and retrieval related tasks (Khan et al., 2020).

It is typically composed of a stack of convolutional modules that perform feature extraction (Agarap, 2017). The basic CNN structure for classification problems consists of different layers namely convolutional layers, pooling layers, and fully connected layers. The convolution layers consist of filters (kernels) with a specified stride and are responsible for the extraction of useful features such as edges, from the input data image. Stride explains as the number of shifts of pixels on the input data matrix. The pooling layer is design for reducing the spatial size of the input data thereby restricting the number of parameters and computation in the network. It independently operates on each feature map, whereby max-pooling being the most common approach used in the pooling process. In the fully connected layer, every node in the first layer is associated with every node in the second layer of the deep network system (Saha & Manickavasagan, 2021).

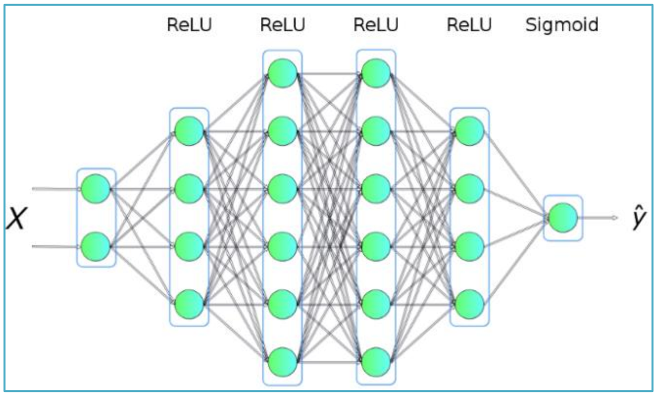


Figure 2.1: Dense neural network architecture (**Skalski**, 2018)

The convolutional layer is a set of convolutional kernels each neuron acts as a kernel. The sub-division of an image into small blocks helps in extracting feature motifs. The Kernel involves the images using a specific set of weights by multiplying its elements with the corresponding elements of the receptive field.

Pooling layers help to extract a combination of features, which are invariant to translational shifts and small (Khan et al., 2020). Reducing the size of the feature map to an invariant feature set not only regulates the complexity of the network but also helps to increase the generalization by reducing overfitting. There are different types of pooling formulations such as max, average; L2, overlapping, spatial pyramid pooling are used in CNN.

The Activation Function serves as a decision function and helps in learning of complex patterns. The use an appropriate activation function can accelerate the learning process. There are various activation functions such ReLU, sigmoid, and tanh, ReLU. The activation function MISH, which has shown great performance than ReLU in most of the recently proposed deep networks on benchmark datasets (Khan et al., 2020).

Dropout is the way regularization within the network, which ultimately improves generalization by randomly skipping some units or connections with a certain probability. In multiple connections learn a non-linear relation are sometimes co-adapted, which causes overfitting (Khan et al., 2020). The random dropping of some connections or units produces several thinned network architectures, and finally, one representative network is selected with small weights.

Fully Connected layer is used at the end of the network for classification. Unlike pooling and convolution, it is a global operation. It takes input from feature extraction and globally analyses the output of all the preceding layers (Khan et al., 2020).

2.7. Related Work

In this sub-section, related works of earlier studies are reviewed and presented. Those literatures in the area computer vision and machine learning algorithms in agricultural products are reviewed chronologically.

Laurent et al. (2010) investigated the effect of storage on hard-to-cook haricot beans using computer vision systems. Histogram features were used for their analysis. They confirmed that beans undergo color changes during storage, which is related to the hard-to-cook phenomenon. The findings demonstrated the ability of color histogram-based image processing of beans for assessing such phenomena in terms of color image attributes.

Nasirahmadi & Behroozi-Khazaei (2013) proposed a computer vision system to identify ten varieties of beans grown in Iran, which employs a multilayer perception artificial neural network (MLP-ANN) to classify the grains from color features. However, different from most of the works addressing bean classification, the results obtained in this work were expressed only in terms of sensibility (96%) and specificity (97.1%).

Araújo et al. (2015) also proposed a method for fast quality control of red beans, which first maps the pixels belonging to the grains, extracting them from the blue background, and then classifies those pixels using an algorithm based on the fuzzy C-means clustering (FCM), to determine the mixture of other types of commercial beans as well as low discolored red beans, in the analyzed sample. In their experiments, they reported accuracies varying from 10 to 96% depending on the mixture of colors and textures of the grains contained in the analyzed sample.

Kavitha & Suruliandi. (2016) proposed that a feature extraction model using texture and color was proposed for the identification of melanoma. The image is preprocessed to increase the resolution and it is segmented using Simple Adaptive Thresholding Algorithm. Then the filtered image is subjected to feature extraction. These texture features are used to evaluate skin lesion discrimination using GLCM matrix. Histogram analysis technique is used for color feature extraction. The experimental result shows that when the texture feature is combined with RGB color space have a results of 93% and texture feature provides in the case of sensitivity and specificity achieved the result of 88.2 % and 85.5% respectively . In this work they didn’t use CNN as feature extraction and classifier and no done any other classifier algorithm for comparison.

Momin et al. (2017) performed a machine vision-based soybean quality evaluation. The authors used image processing algorithms to detect various forms of Materials Other than Grain (MOG), also known as dockage fractions, such as split beans, contaminated beans, defect beans and stem/pods. The HSI (hue, saturation, and intensity) color model was used to segment the image background and subsequently, dockage fractions were detected using median blurring, morphological operators, watershed transformation, and component labeling based on projected area and circularity. The algorithms successfully identified the dockage fractions with an accuracy of 96% for split beans, 75% for contaminated beans, and 98% for both defect beans and stem/pods.

Pinto et al.(2017) proposed classification of green coffee bean images based on defect types using convolutional neural network, The result of CNN showed that the set of some classes had good classification accuracy which is 90% and some other classes had lower classification accuracy have 72% for the color images, They concluded that CNN has some advantages for feature detection on the shapes of the image of spacial filters however in the classification coffee bean images using the CNN model, the color characteristics have a strong influence. They have not used other feature extraction method like GlCM or any.

Mengistu (2018) proposed the effects of segmentation techniques in the digital image-based identification of Ethiopian coffee varieties. He used Otsu, Fuzzy-C-Means (FCM), and K-means segmentation techniques. To classify different varieties of Ethiopian coffee beans based on growing region back propagation neural network (BPNN) was used and obtained 94.54% accuracy.

Also, Garcia et al. (2018) proposed a computer vision system for automatic bean classification. They evaluated the performance of SVM and RF algorithms in classifying local and global features extracted from the segmented grains, which are combined using a bag-of feature (BoF) method. They achieved 98.5% accuracy using RF and global features. Although the authors mention that the beans are segmented using the WT algorithm, they do not present results obtained in the segmentation step.

Tian et al. (2019) proposed the detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense. They have used two types of images, healthy apple images and diseased apple images and the datasheets were 500 healthy apple images and 140 diseased apple images. All images were resized to 512x512 pixels. They used based on image data augmentation, densely connected neural network (DenseNet) utilized to optimize feature layers of the YOLO-V3 model which have lower resolution. The report showed that DenseNet greatly improved the utilization of features in the neural network and enhanced the detection result of the YOLO-V3 model.

Wallelign et al. (2019) proposed to design a robust CNN model that classifies raw coffee beans into their 12 quality grades using small datasets which have high data variability. They used different camera resolutions one using Samsung s7 edge camera resolution 2268x4032 pixels and the second iPhone 7 with a resolution of 3024x4032 pixels were captured and they tried to fit all the beans on the A4 white paper. All images were resized to 224x224 pixels, intensity values were scaled between 0 and 1. The report showed that how the images captured, camera differences, illumination conditions, and scale introduced task-irrelevant features to datasets. They also used preprocessing techniques applied to the input in order to reduce task-irrelevant features. But their report showed that adding the preprocessing techniques did not improve the performance of the CNN model. Finally, they reported that accuracy of 89.01% on the test dataset for classifies the beans into their quality grades.

Yogesh et al. (2020) proposed computer vision-based analysis and detection of defects in fruits using pixels containing the defected regions is segmented and their features are extracted. They captured the images by a 13 MP CMOS camera with a resolution of 1920 * 1080 pixels, and the image is resized to 120 * 120 pixels in JPEG format. All the samples are taken at a distance of 60 cm. RGB images of fruits are taken for quality inspection based on non-invasive methods. The proposed system includes segmentation, classification, and recognition of defects in fruits. The database of fruit has been created by a human expert that includes two categories i.e. healthy fruit and defective fruit. The output of defect is observed with KNN, GoogleNet, and SVM with various fruits like apple, pears, pomegranate, and litchi. The result showed that the SVM approach is better as a fruit classifier in terms of defect detection. But this research has a limitation of the unavailability of a pre-trained network of various fruit defects. In the absence of a pre-trained network, one needs to create a database of defected fruit from scratch. It will consume lots of time.

In the work, Dos Santos et al. (2020) employed both computer vision and machine learning for classifying coffee beans defects according to their shape and color features. They used DNN, RF, and SVM algorithms for the classification of coffee defect types. The results had shown that all the classifier models presented similar performance. However, the color descriptors pointed to the importance of classifying coffee bean defects. The data reported in this study provides evidence that computer vision along with machine learning algorithms can be used to identify and classify coffee beans with an accuracy of 88%.

In the work, Adhitya et al. (2020) employed a methodology for textural feature extraction using GLCM on digital images of cocoa beans. The co-occurrence matrix features of the gray level co-occurrence matrix were compared with the convolutional neural network (CNN) method for the feature extraction method. They applied classifiers for conclusive assessment and classification to obtain an accuracy performance analysis. They conclude that using the GLCM texture feature extraction more reliable results than using CNN feature extraction from the final classification.

Summary

In summary, the above studies use different segmentation, feature extraction, and classification techniques depending on the characteristics of the agricultural products. In the reviewed literature, there are several works proposing the use of computer vision systems and machine learning techniques for the analysis and classification of seeds and grains. However, there are very few works dealing with an automatic visual inspection of grains, especially in haricot beans. Even, there are differences in size, color and shape among haricot beans.

On the other hand, defect inspection of agricultural products is the most important factor affecting world market competence and customer satisfaction. The existing manual inspections have had many problems in maintaining consistency and ensuring satisfactory detection efficiency. For the evaluation of haricot bean defect detection take considerable time, and sometimes scarce of qualifies expert in the area. In addition, subjective errors from manual inspections have their own impact. To overcome this, the use of modern technology is very important. As a result, this research work aims to develop defect inspection of haricot beans using computer vision and machine learning algorithms approach.

The present research work considered and tested different segmentation and feature extraction technique for haricot bean defect detection. As the color, size, and morphology of haricot beans are different from the above-reviewed studies.

As final step, GLCM and CNN for feature extraction and SVM and RF for classification were used. The algorithms were chosen based on their efficiency in earlier studies for other feature extractions and classifications.

CHAPTER 3: METHODOLOGY

3.1. Introduction

In this chapter, a detailed description of the haricot bean classification model based on defect and non-defect haricot bean is discussed. In section 3.2, overview of the Model is presented. The methods of haricot image collection are discussed in section 3.3. Then description of image preprocessing techniques, feature extraction, and classification are presented in section 3.4, section 3.5 and section 3.6 respectively.

3.2. System Architecture

The proposed system includes image preprocessing, feature extraction, and image classification. In image preprocessing, resizing image, noise removal, enhancing the image quality contrast and brightness, image adjustment or reconstruction, image transformation segmentation, and determination of the part of the image to be evaluated. The preprocessing of haricot bean images is performed by feature extraction, measurement, and filtering. The third component of the method is feature extraction. It contains CNN and GLCM feature extraction. CNN was used for deep feature extraction because it is powerful. GLCM algorithm was used for texture picks up based on the relation between two pixels at a time, i.e., the reference and the neighbor pixel. GLCM is prepared from grayscale values and considers how often a pixel is with gray level (Sharma et al., 2015). Finally, SVM and RF were utilized for the classification of haricot bean defect and non-defect. The proposed model is presented in figure 3.1 below.

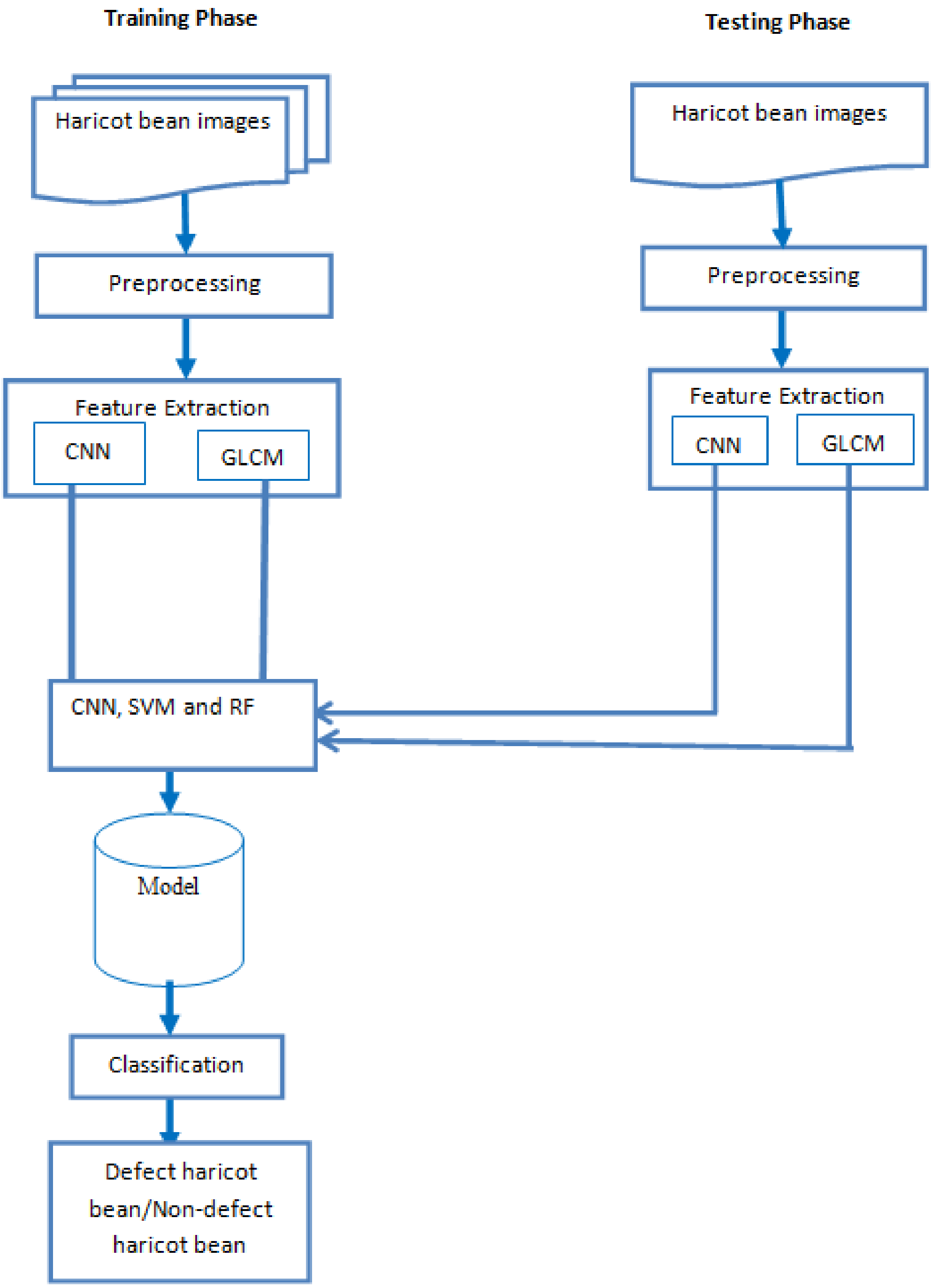


Figure 3.1: Proposed model for haricot bean quality and defect detection

3.3. Image Acquisition

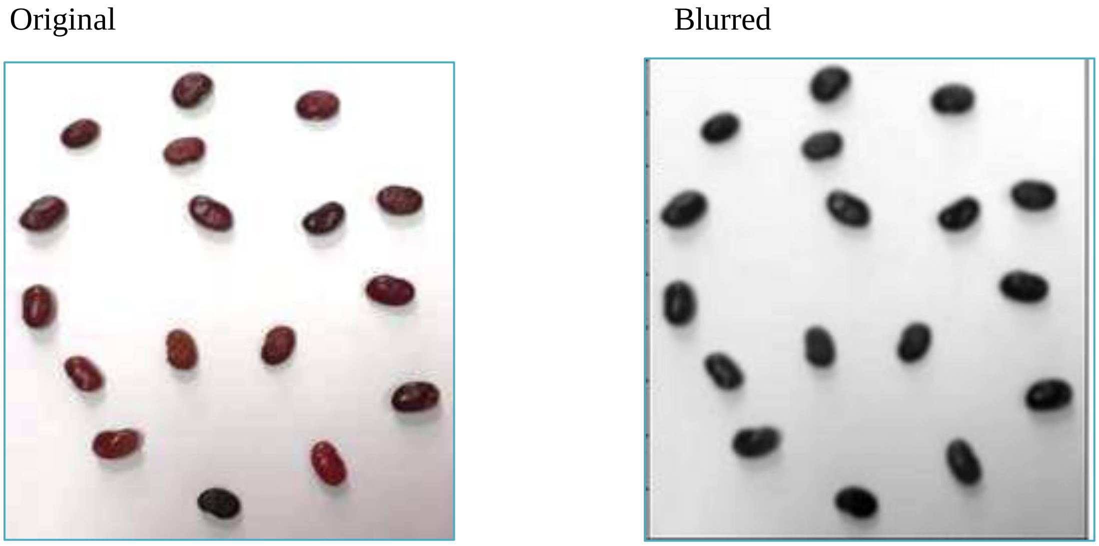
In order to collect the haricot bean images, the haricot bean were categorized into defect, and non-defects types. The image was captured using Samsung A10 with 13 MP CMOS camera having a resolution of 4128x1956 pixels. In the work of (Wallelign et al., 2019), coffee bean sample images were captured using Samsung G7 edge camera with resolution of 2268x4032 pixels on a white paper A4 size.

3.4.1. Image Resizing

The captured images contain large image sizes and, thus, resized into (224 x224) pixels. The image size was adapted from (Wallelign et al., 2019), which was used for coffee beans grading using CNN.

3.4.2. Noise Removal

Gaussian blur filter was used by calculating a weighted average of the color values of the pixels. In a Gaussian blur, the pixels nearest the center of the kernel are given more weight than those far away from the center. This averaging is done on a channel-by-channel basis, and the average channel values become the new value for the filtered pixel. Larger kernels have more values factored into the average, and this implies that a larger kernel will blur the image more than a smaller kernel.



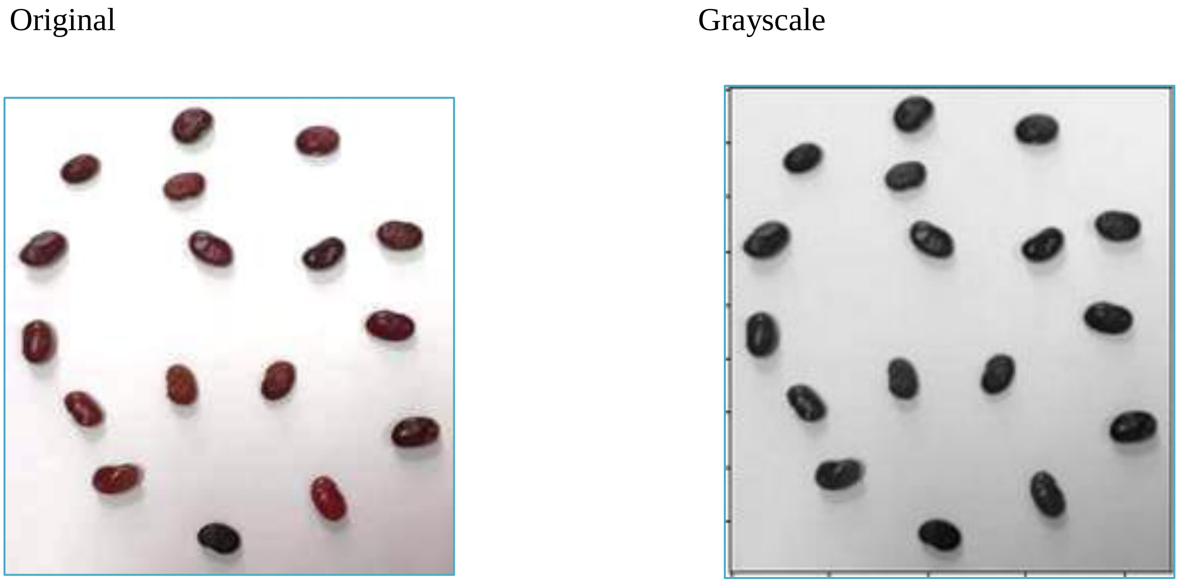
Original

Blurred

Figure 3.3: Gaussian blur

3.4.3. Grayscale Conversion

All the input images are presented in an RGB format. These have to be first converted from RGB format to a grayscale format. A grayscale (or gray level) image is simply one in which the only colors are shades of gray. Grayscale conversion is the process of converting the true color image (RGB) to the grayscale image which is usually performed by matching the luminance of the color image. Converting the RGB image into a gray scale images are preferred over colored ones to simplify mathematics. It is relatively easier to deal with (in terms of mathematics) a single color channel (shades of white/black) than multiple color channels. The grayscale conversion was performed using OpenCV cvtColor module. An example image is shown in Figure 3.4 created using the OpenCV cvtColor module.



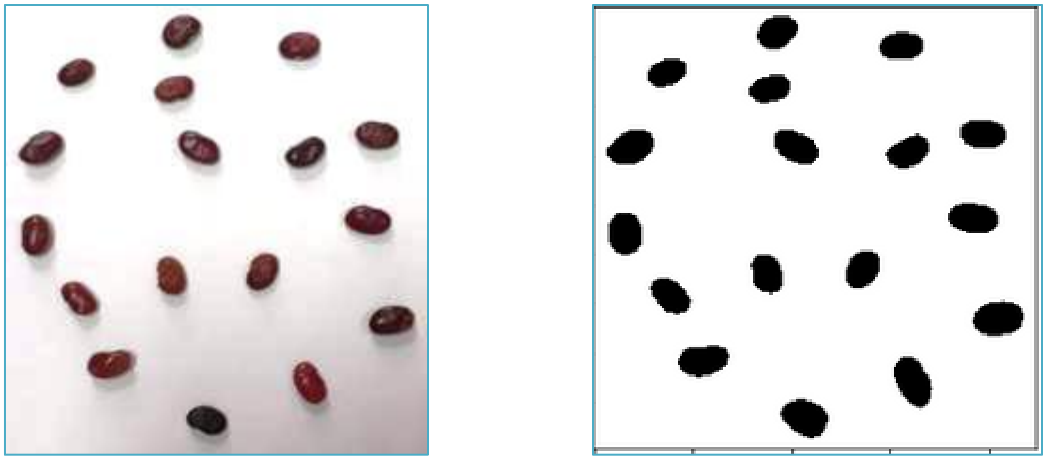
Original

Grayscale

Figure 3.4: Grayscale conversions

3.4.4. Segmentations

Otsu threshold segmentation for this work was used as it is a technique based on automatic selection region. The method is a type of global thresholding in which it depends only on gray value of the image. It requires computing a gray level histogram before running. For two dimensional, this method is usually proposed which works on both gray-level threshold of each pixel. Otsu’s method is expected in finding the optimal value for the global threshold and based on the interclass variance maximization (Patil & Shaikh, 2016).



(a) (b)

Figure 3.5: *(a) Original Image (b) Result of OTSU Threshold Segmentation*

3.5. Feature Extraction

In this thesis GLCM texture features and CNN feature methods were used for feature extraction. For the evolution of machine learning, CNN was used for the feature extraction process.

3.5.1. Feature Extraction Using CNN

After segmentation with Otsu thresholding, the segmented of each haricot bean contain pixels as input of CNN model.

The input to the first convolution layer is segmented 224 x 224 x 3 images. Thirty two convolution layers were used. This number is picked by using the trial and error method. If the network is unable to extract features, the layers were increased but after 16 layers the result is similar, so 16 convolution layers were considered in this study.

Table 3.1: CNN Model for Deep Feature Extraction of Haricot bean

Layer	Filter	padding	Output map size	Activation
Con2d	3x3	1	224x224x3	relu
Con2d	3x3		224x224x32	
Dropout	3x3		112x112x32	
Con2d	3x3	1	112x112x112	relu
Maxpooling	2x2		55x55x16	
Flatten			48400	
Dense1			128	relu
Dense2			2	sigmoid

A typical CNN architecture consists of several nested convolutional and pooling layers followed by fully connected layers at the end. A simplified presentation of this kind of network can be (Input - Conv - ReLU - Pool – FC) (Wang et al., 2018).

Input: The input of the CNN tends to be the 3-channel color image or 1-channel gray image matrices containing the intensity values at each position.

Conv: In the first convolutional layer, the (224 x 224 x3) preprocessed 38 images was entered. When move to the next convolutional layer, it extracts the features of the haricot bean defect and defects based on color and shape.

Activation Functions:-we tried to test different types of activation functions like sigmoid, Relu, and were tested and the sigmoid activation function show a better result. Sigmoid is the most widely used activation function by adding max-pooling 2X2 transformations to the output response of the convolutional network.

Maxpooling: - is to reduce the amount of the network parameters and the computation cost. The pooling layer is common to be placed between two successive convolutional layers. Max-pooling 2X2 that outputs the max value from the neighborhood of the input feature map were used. After all, these layers were stacked together to form a complete CNN. The input is fed forward into the network for decision making. And the hyper parameters are updated by the backpropagation algorithm.

Dropout is used to prevent overfitting and most popular regularization technique for deep neural networks. We use in the input and the hidden layer nodes.

3.5.2. Texture Feature Extraction (GLCM)

In statistical texture analysis, from the distribution of intensities, the texture features are obtained at specified positions relative to one another in an image. The statistics of texture are categorized into first-order, second-order, and higher-order statistics. The technique of extracting second-order statistical texture features is done using Gray Level Co-occurence Matrix (GLCM). The first-order texture measure is not related to pixel neighbor relationships and it is calculated from the original image. GLCM done by the relation between two pixels at the same time, called a reference pixel and a neighbor pixel. A Gray Level Co-occurence Matrix is defined by a matrix in which the number of rows and columns are equal to the number of gray levels G in an image. The matrix element P (i, j | Δx, Δy) is the relative frequency where i and j represent the intensity and both are separated by a pixel distance Δx, Δy. The different textural features such as energy, entropy, contrast, homogeneity, correlation, and dissimilarity can be computed using the GLCM matrix. Gray level Co-occurrence Matrix shows how the pixel brightness in an image occurs. GLCM also provides different measures i.e. entropy, energy, contrast, and correlation (Wang et al., 2018). In the case of this study, when using CNN as feature extraction, if there is changes in the shape of the haricot bean image and different from the trained shape, CNN may considers as defect. So to avoid such problem GLCM is prepared from grayscale values. It is taken into account how often a pixel with the gray level that is the grayscale intensity or gray tone values comes either horizontally, vertically, and diagonally to leveled the pixels with the values. GLCM extracted features in terms of the following.

Contrast: is the Sum of Square Variance'. It defers the calculation of the intensity contrast linking pixel and its neighbor over the whole image. At constant image contrast value is 0.In contrast measure, weight increases exponentially (0, 1, 4, 9) as persists from the diagonal. Range=[0,size(GLCM,1)-1)^2]

$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \text{ -- } (3.1)$$

Since (i-j) increases contrast continues to increase exponentially. When i and j are equal i.e. i-j=0.no contrast is there. When i and j are differ by 1, small contrast is there is 1. When i and j differ by 2, the contrast is expanding and weight is 4 (Sharma et al., 2015)

Correlation: It passes the calculation of the correlation of a pixel and its neighbor over the whole image means it figures out the linear dependency of gray levels on those of neighbouring pixels. On behalf a perfectly positively or negative correlated image, the correlation value is 1 and -1.On behalf of constant image its value is NaN...Range= [-1, 1] and the formula is

$$\sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right] \text{-- } (3.2)$$

Energy: Since energy is used for doing work, Thus orderliness. It makes use for the texture that calculates orders in an image. It gives the sum of square elements in GLCM. It is fully different from entropy. When the window is proficient orderly, energy value is high .The square root of ASM (Angular Second Moment) texture character is used as Energy. Its range is[0 1].Since constant image its value is 1.

The equation of energy is

$$\sum_{i,j=0}^{N-1} P_{i,j}^2 \text{ -- } (3.3)$$

Homogeneity: In short term it is going by the name of HOM. It passes the value that calculates the tightness of distribution of the elements in the GLCM to the GLCM diagonal. For diagonal GLCM its value is 1 and its range is [0, 1].Opposite of contrast weight is homogeneity weight values, with weight decreases exponentially loose from the diagonal. The weight employed in contrast is (i-j)^2 and in homogeneity ,it is 1/1+(i-j) ^{2}.

The equation is $\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$ --- 3.4)

Dissimilarity is a measure of distance between pairs of objects pixels.

The equation is $\sum_{i,j=0}^{N-1} P_{i,j}|i-j|$ --- (3.5)

3.6. Classification

In this study for haricot bean image classification we have used three machine learning algorithms. These are CNN, RF and SVM. In the following topics we will see each of these classifier algorithms.

3.6.1 Conventional Neural Network Classifier

The layer parameters are consists of a number of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward propagation, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As the result, the network learns about the filters. The filter activates specific type of feature at some spatial position in the input. The activation function is to down sampling layer, and like convolutions, this method is applied one patch at a time. Finally CNN has also fully connected layer that classifies the images the desired classes.

After feature extracted from both CNN and GLCM we have used SVM and RF algorithms for the classification of images defect haricot bean or non- defect Haricot beans.

3.6.2 Support Vector Machine Classifier

Our proposed method goes as the input (feature vectors) was based on the results of the convolutional neural network and GLCM. CNN was used to extract features and those features were used to train using SVM model and classifies, the same scenario in GLCM, features extracted from GLCM and train SVM model then classify. Once the SVM classifier was trained using the feature vectors, the class of images were predicted.

3.6.3 Random Forest (RF) Classifier

In our proposed method takes the input (feature vectors) was based on the results of the convolutional neural network and GLCM. CNN was used to extract features and those features were used to train using RF model and classify, the same scenario in GLCM, features extracted from GLCM and train RF model then classify. Once the RF classifier was trained using the feature vectors, the class of images were predicted.

3.7 Evaluation Metrics

For the evaluation and comparison of our model performances, different statistical performance metrics was applied. For binary classifications, the metrics can be calculated based on the entries of a confusion matrix. The comparison of the predicted class with the actual class allows distinguishing between correctly positive or negative classified for example true positive, true negative and incorrectly classified examples (false positive, false negative). This distinction in turn enables the calculation of various statistical quality measures (Schmitt et al., 2020).

		Predicted value	
		Defect	Non-defect
Actual value	Defect	True positive (TP)	False Negative (FN)
	Non-defect	False positive (FP)	True Negative (TN)

Table 3.2 Confusion matrix for the evaluation the performance of the model

Terms associated with Confusion matrix:

Predicted value: Outcome of the model on the validation set

Actual value: Values seen in the training set

True Positives (TP): are the cases when the actual class of the data point was true and the predicted is also true.

True Negatives (TN): are the cases when the actual class of the data point was false and the predicted is also false.

False Positives (FP): are the cases when the actual class of the data point was False and the predicted is true. False is because the model has predicted incorrectly and positive because the class predicted was a positive one.

False Negatives (FN): are the cases when the actual class of the data point was true and the predicted is false. False is because the model has predicted incorrectly and negative because the class predicted was a negative one.

Different performance metrics are used to evaluate different Machine Learning Algorithms. For this research work we used for image classification of defect haricot bean and non-defect haricot bean for we used evaluation metric of machine learning algorithms such as Accuracy, precision, recall, and FL.

Accuracy is in classification problems is the number of correct predictions made by the model over all kinds predictions made. It is one metric which gives the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy = Number of correct predictions / Total number of predictions.

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FP+FN} \text{---------------------------3.1}$$

Precision is calculated as the number of correct positive predictions (TP), divided by the total number of positive predictions (TP + FP).Precision gives the fraction of correctly identified as positive out of all predicted as positives.

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{---------------------------3.2}$$

Recall is used to measure the fraction of positive patterns that are correctly classified

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{------------------------------3.3}$$

F-score is a measure of the accuracy of the test. It is calculated, based on precision and reminders, by the formula:

$$\text{F-score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{---------------------------3.4}$$

In this research case, these representations can be interpreted as:

TP: The number of defect haricot bean classified as defect haricot bean.

TN: The number of non-defect haricot bean classified as non-defect haricot bean

FP: The number of non-defect haricot bean as defect haricot beans.

FN: The number of defect haricot beans classified as non-defect haricot bean.

CHAPTER 4: EXPERIMENTATION AND DISCUSSIONS

In this study an attempts is made to construct a model for haricot bean quality and defect inspection mode .Detail implementation procedure, dataset preparation and experimental results are presented below.

4.1. Dataset Preparation

The images of haricot bean grains were collected from ECX in bure branch Amhara region. The haricot bean grain legumes are categorized by the domain experts in the ECX quality inspection laboratory. The sample haricot bean grain legumes used in our experimentation are summarized in Table 4.1 below. A total of 2000 haricot bean for each class are prepared to train and test to the proposed model. These 2000 haricot bean sample constituents are separated into their corresponding two classes defect and Non-defect haricot beans. Therefore, we finally have 1 output to predict defect or non-defect haricot beans. Image acquisition is done using Samsung mobile Model A10 with specification of 13 mega pixels. During image acquisition, the camera is mounted on a stand which provides easy vertical movement. We used A4 size white for capturing images to remove background noise. The samples of haricot bean are placed directly under the camera for image acquisition.

There is no fixed rule for separation training and testing datasets. Most of the researchers were used 70:30 ratio for separation datasets. It is also depends on data characters, data size etc. We can used 70:30; 80:20; 65:35; 60:40 etc. anything which suits of your data characters (Pankaj, 2020).we have checked different ratio for training and testing split but the best performance among them was training 75% of the data is used and 25% of the data is used for testing the model.

(a)**Training Dataset**

Training data set is the general term for the samples used to create the model

No	Sample type	Data size	Image format
1	Non-defect Haricot beans	750	JPEG
2	Defect haricot beans	750	JPEG

(b) **Test Dataset**

Test data set is used to qualify performance

No	Sample type	Data size	Image format
1	Non-defect Haricot beans	250	JPEG
2	Defect haricot beans	250	JPEG

Table 4.1: Train and test data image of haricot beans

4.2. Experimental Setup

The implementation of this research work is done under a machine that has the following Specification details. Experiments and related analysis processes are done:

- Computer with Intel Core ™ i5-7200 CPU
- 2.7GHZ speed processor
- 8.00 GB RAM
- 1TB hard disk space
- Windows 10 (Pro) installed

4.3 Experimental Tools and Techniques

As a programming language Python 3.8 is used which an open-source, with variety is of free libraries, rich documentation, including contributor support. The supportive libraries and Software tools are listed next.

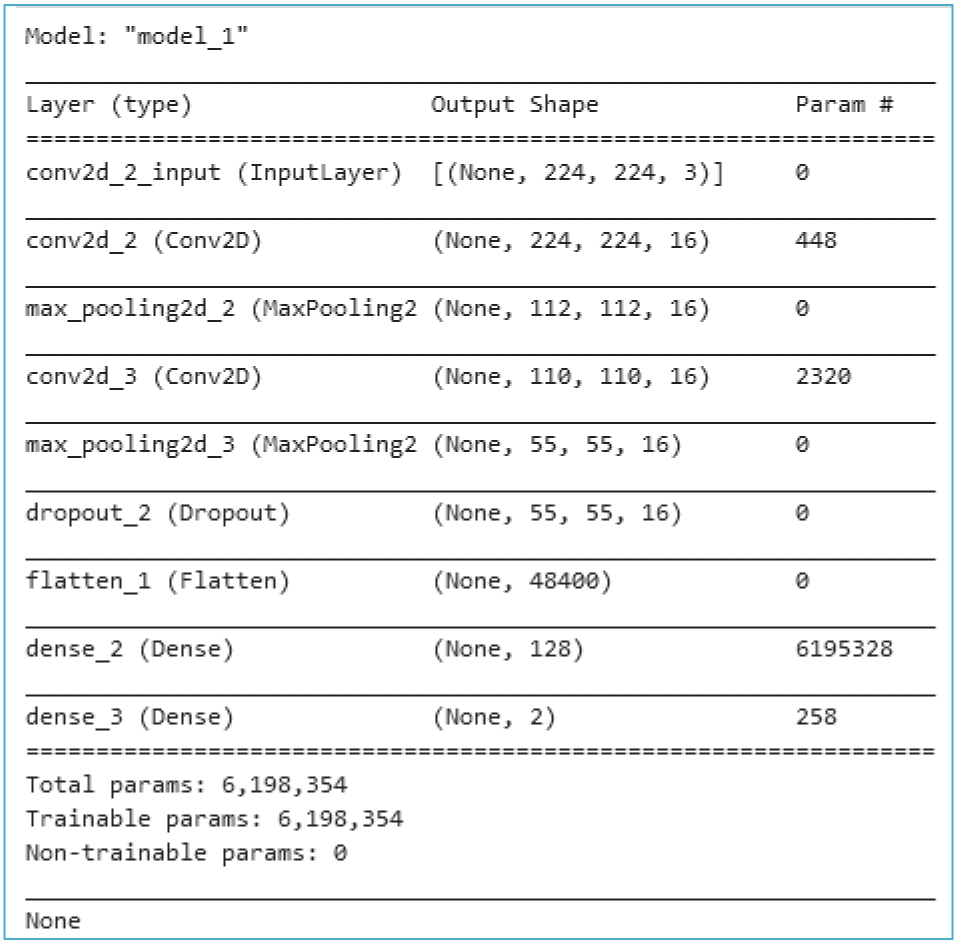
- Numpy : library for mathematical functionalities
- Matplotlib : plotting library
- OpenCV : image processing library and computer vision library
- Scikit-learn : machine learning library
- Mahotas : additional computer vision and image processing library

Jupiter notebook development IDE is used and the program is done with Python 3.8 language with OpenCV and Keras. OpenCV is an open-source library of image processing functions, whose goal is real-time computer vision. Keras is the system modular library written in Python capable of running on top of Tensor Flow. The TF (Tensor Flow) was selected as a backend that both TF and Keras were optimized to perform tasks. Both systems are implemented in Python which allows the user to work with them in a compact way without having to use multiple files as a programming language.

The preprocessed images further labeled, encoded and split the image train and test using 75% for training and 25% for testing.

4.4. Experiment on Feature Extraction using Convolutional Neural Network (CNN)

We used haricot bean images, the first step; the haricot bean images were labeled defect and non-defect haricot bean, and loaded into an array based on each class. This full dataset array was split into 75% for training and 25% for testing, and then a CNN was built using the architecture layer in Figure 4.1



```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2_input (InputLayer)  [(None, 224, 224, 3)]     0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 224, 224, 16)      448
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 112, 112, 16)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 110, 110, 16)      2320
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 55, 55, 16)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 55, 55, 16)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 48400)             0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               6195328
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 258
=================================================================
Total params: 6,198,354
Trainable params: 6,198,354
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.1: Summary of our proposed model

The following diagrams show the training accuracy, testing accuracy, training loss, and testing loss of our end to end CNN model.

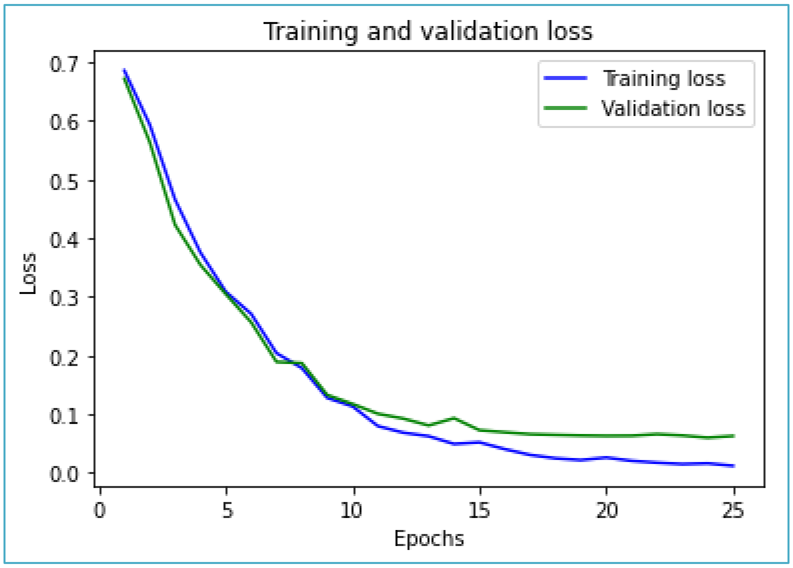


Figure 4.2a: The Training Loss and validation loss of our end to end CNN model

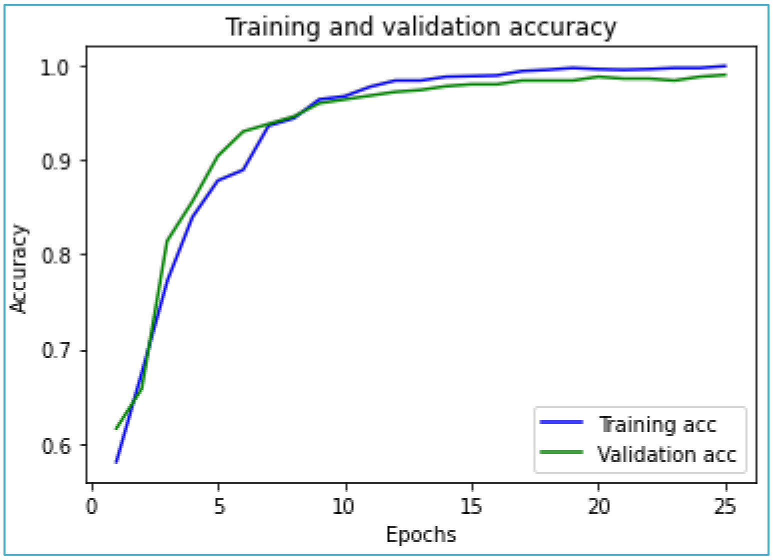
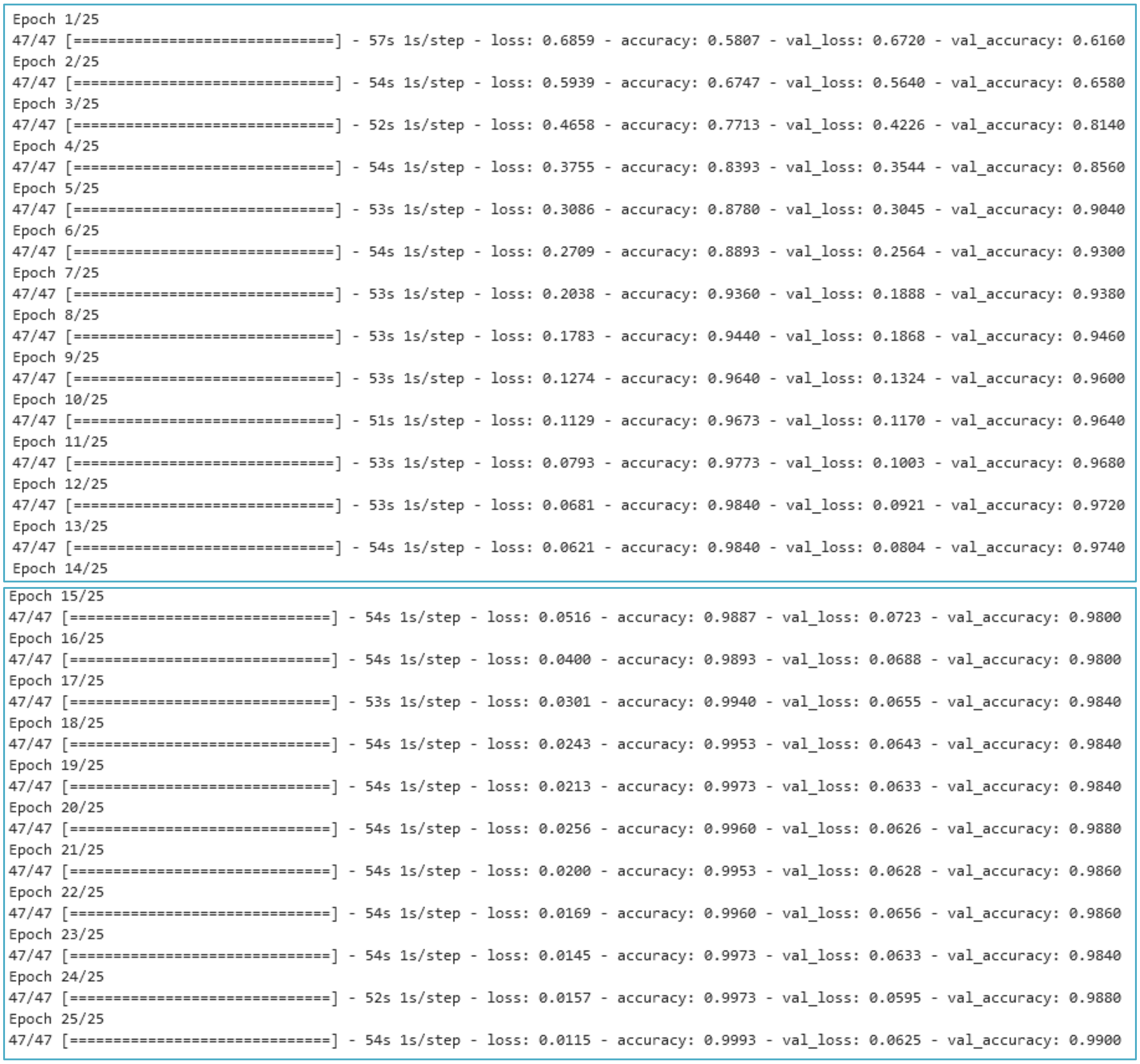


Figure 4.2b: The Training accuracy and validation accuracy of our end to end CNN model.

Figures 4.2a and 4.2b shows the accuracy and loss for the feature extraction model at epoch 25 applied to the haricot bean dataset. Training loss is the summation of an error made on the training dataset, and it also implies the model performance behavior of each iteration. And also validation loss is a result error after running the validation dataset through the previously trained network.



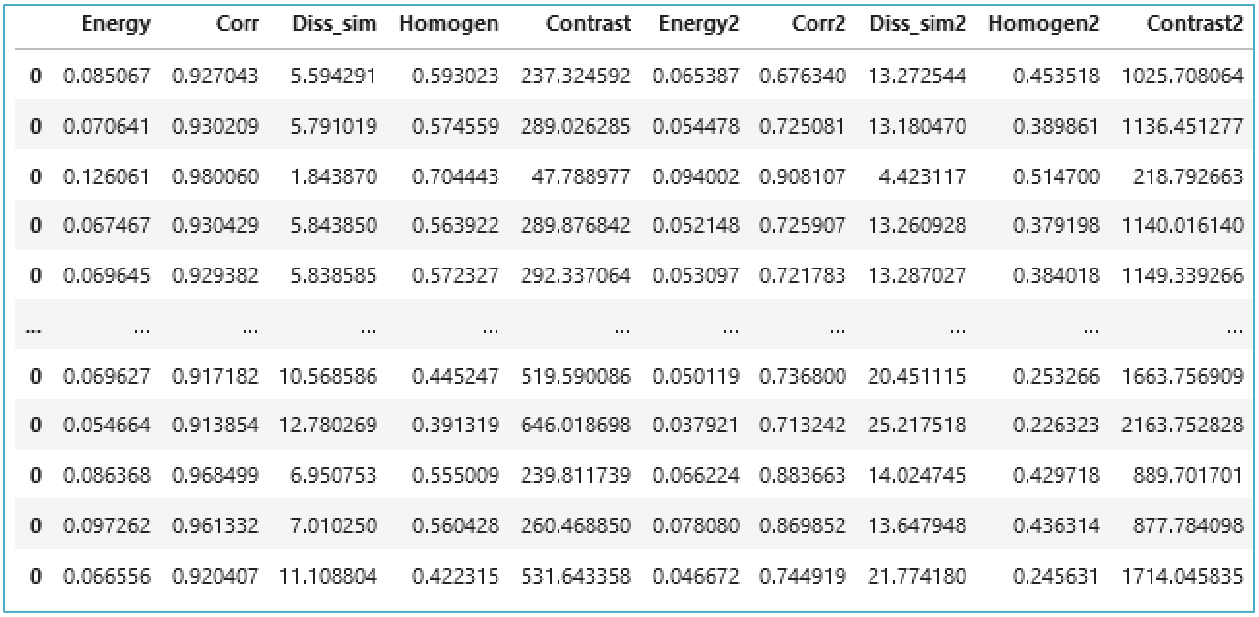
```
Epoch 1/25
47/47 [==============================] - 57s 1s/step - loss: 0.6859 - accuracy: 0.5807 - val_loss: 0.6720 - val_accuracy: 0.6160
Epoch 2/25
47/47 [==============================] - 54s 1s/step - loss: 0.5939 - accuracy: 0.6747 - val_loss: 0.5640 - val_accuracy: 0.6580
Epoch 3/25
47/47 [==============================] - 52s 1s/step - loss: 0.4658 - accuracy: 0.7713 - val_loss: 0.4226 - val_accuracy: 0.8140
Epoch 4/25
47/47 [==============================] - 54s 1s/step - loss: 0.3755 - accuracy: 0.8393 - val_loss: 0.3544 - val_accuracy: 0.8560
Epoch 5/25
47/47 [==============================] - 53s 1s/step - loss: 0.3086 - accuracy: 0.8780 - val_loss: 0.3045 - val_accuracy: 0.9040
Epoch 6/25
47/47 [==============================] - 54s 1s/step - loss: 0.2709 - accuracy: 0.8893 - val_loss: 0.2564 - val_accuracy: 0.9300
Epoch 7/25
47/47 [==============================] - 53s 1s/step - loss: 0.2038 - accuracy: 0.9360 - val_loss: 0.1888 - val_accuracy: 0.9380
Epoch 8/25
47/47 [==============================] - 53s 1s/step - loss: 0.1783 - accuracy: 0.9440 - val_loss: 0.1868 - val_accuracy: 0.9460
Epoch 9/25
47/47 [==============================] - 53s 1s/step - loss: 0.1274 - accuracy: 0.9640 - val_loss: 0.1324 - val_accuracy: 0.9600
Epoch 10/25
47/47 [==============================] - 51s 1s/step - loss: 0.1129 - accuracy: 0.9673 - val_loss: 0.1170 - val_accuracy: 0.9640
Epoch 11/25
47/47 [==============================] - 53s 1s/step - loss: 0.0793 - accuracy: 0.9773 - val_loss: 0.1003 - val_accuracy: 0.9680
Epoch 12/25
47/47 [==============================] - 53s 1s/step - loss: 0.0681 - accuracy: 0.9840 - val_loss: 0.0921 - val_accuracy: 0.9720
Epoch 13/25
47/47 [==============================] - 54s 1s/step - loss: 0.0621 - accuracy: 0.9840 - val_loss: 0.0804 - val_accuracy: 0.9740
Epoch 14/25
Epoch 15/25
47/47 [==============================] - 54s 1s/step - loss: 0.0516 - accuracy: 0.9887 - val_loss: 0.0723 - val_accuracy: 0.9800
Epoch 16/25
47/47 [==============================] - 54s 1s/step - loss: 0.0400 - accuracy: 0.9893 - val_loss: 0.0688 - val_accuracy: 0.9800
Epoch 17/25
47/47 [==============================] - 53s 1s/step - loss: 0.0301 - accuracy: 0.9940 - val_loss: 0.0655 - val_accuracy: 0.9840
Epoch 18/25
47/47 [==============================] - 54s 1s/step - loss: 0.0243 - accuracy: 0.9953 - val_loss: 0.0643 - val_accuracy: 0.9840
Epoch 19/25
47/47 [==============================] - 54s 1s/step - loss: 0.0213 - accuracy: 0.9973 - val_loss: 0.0633 - val_accuracy: 0.9840
Epoch 20/25
47/47 [==============================] - 54s 1s/step - loss: 0.0256 - accuracy: 0.9960 - val_loss: 0.0626 - val_accuracy: 0.9880
Epoch 21/25
47/47 [==============================] - 54s 1s/step - loss: 0.0200 - accuracy: 0.9953 - val_loss: 0.0628 - val_accuracy: 0.9860
Epoch 22/25
47/47 [==============================] - 54s 1s/step - loss: 0.0169 - accuracy: 0.9960 - val_loss: 0.0656 - val_accuracy: 0.9860
Epoch 23/25
47/47 [==============================] - 54s 1s/step - loss: 0.0145 - accuracy: 0.9973 - val_loss: 0.0633 - val_accuracy: 0.9840
Epoch 24/25
47/47 [==============================] - 52s 1s/step - loss: 0.0157 - accuracy: 0.9973 - val_loss: 0.0595 - val_accuracy: 0.9880
Epoch 25/25
47/47 [==============================] - 54s 1s/step - loss: 0.0115 - accuracy: 0.9993 - val_loss: 0.0625 - val_accuracy: 0.9900
```

Table 4.2 Times elapsed, Training loss, Accuracy, and Validation loss and Validation Accuracy for each Epoch

We made an experiment for epoch of 5, 10, 15, 20, and 25 on our feature extraction experimental parameters for the CNN feature extraction. From the experiment we have seen that the number of epoch was increase, the accuracy also increases, while the loss value decreases

4.5. Feature Extraction: Gray Level Co-Occurrence Matrix (GLCM) Textural Features

We applied GLCM texture extraction to the dataset. For the computation of the co-occurrence matrix implemented in this study, the distance = 0 and the angles were 45^0 and 90^0 and we also had degree=0^0 and distance 1, 3 and 5. Table 4.2 provides an example of the extracted features with distance =1 and horizontal angle 0^0. From these five features, i.e., energy, correlation, dissimilarity, homogeneity, and Contrast were extracted, as depicted in table 4.3 25 features were extracted. The two classes of haricot bean defect and non-defect haricot beans features were extracted from the haricot bean image datasets. As we see from the figure 4.4 GLCM extracted 25 features.



	Energy	Corr	Diss_sim	Homogen	Contrast	Energy2	Corr2	Diss_sim2	Homogen2	Contrast2
0	0.085067	0.927043	5.594291	0.593023	237.324592	0.065387	0.676340	13.272544	0.453518	1025.708064
0	0.070641	0.930209	5.791019	0.574559	289.026285	0.054478	0.725081	13.180470	0.389861	1136.451277
0	0.126061	0.980060	1.843870	0.704443	47.788977	0.094002	0.908107	4.423117	0.514700	218.792663
0	0.067467	0.930429	5.843850	0.563922	289.876842	0.052148	0.725907	13.260928	0.379198	1140.016140
0	0.069645	0.929382	5.838585	0.572327	292.337064	0.053097	0.721783	13.287027	0.384018	1149.339266
...	...	...	...	...	...	...	...	...	...	...
0	0.069627	0.917182	10.568586	0.445247	519.590086	0.050119	0.736800	20.451115	0.253266	1663.756909
0	0.054664	0.913854	12.780269	0.391319	646.018698	0.037921	0.713242	25.217518	0.226323	2163.752828
0	0.086368	0.968499	6.950753	0.555009	239.811739	0.066224	0.883663	14.024745	0.429718	889.701701
0	0.097262	0.961332	7.010250	0.560428	260.468850	0.078080	0.869852	13.647948	0.436314	877.784098
0	0.066556	0.920407	11.108804	0.422315	531.643358	0.046672	0.744919	21.774180	0.245631	1714.045835

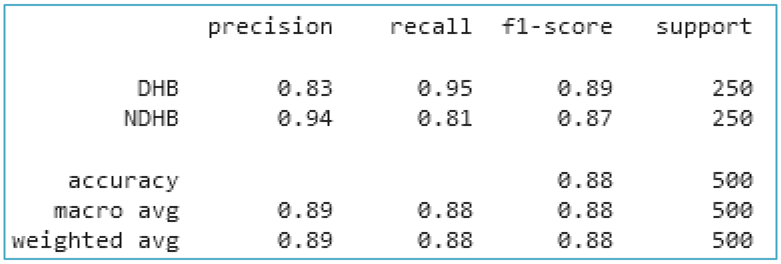
Table 4.3: Features extracted using GLCM texture

4.6. Classification

Classification was carried out based on the extracted features in CNN and GLCM feature extracted. We implemented the classification process using our dataset. As previously mentioned, we split our dataset into 75% for training and 25% for testing. We used the testing (25%) data and then implemented the classification process to a model that had been previously trained. To analyze the performance of the two feature extraction methods, we applied three types of classifiers CNN,SVM and Random forest (RF) classifier and compared the results in which feature extraction way is better performance for the classification of haricot bean defect and non-defect ones.

4.6.1. Experiment on GLCM Feature Extraction and SVM Classifier

In this Experiment features extracted from the GLCM features used to classify using SVM algorithm and achieve the accuracy of 88%



```
              precision    recall  f1-score   support

         DHB       0.83      0.95      0.89       250
        NDHB       0.94      0.81      0.87       250

    accuracy                           0.88       500
   macro avg       0.89      0.88      0.88       500
weighted avg       0.89      0.88      0.88       500
```

Figure 4.3: Precision, recall, and F1-score of SVM with the application of GLCM Features

Figure 4.3 shows that there is 17% false positive when the model predicted Defect Haricot bean, which means the image is not Defect Haricot bean but predicted as Defect. There are 19% false negative when the model predicted Non-defect, which means the image was Non-defect but the model predicts as Defect. The model also has a 5% false negative rate when it predicts the Defect Haricot bean.

Actual value	Defect	Non-defect
Defect	237	13
Non-defect	48	202

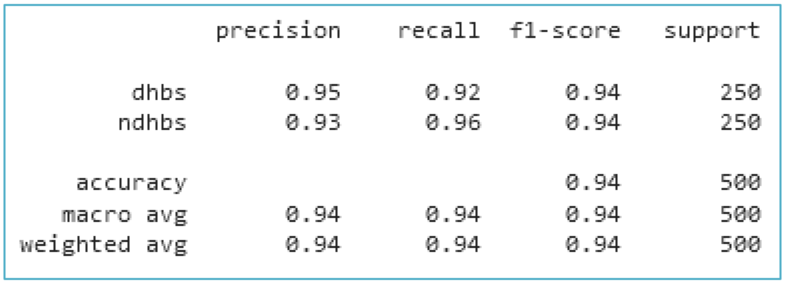
Prediction value

Table 4.4: Performance evaluation matrix of SVM using GLCM features.

As shown in Table 4.4, Non-defect samples are correctly classified as 202 and 48 is incorrectly classified out of 250 samples. Among 250 samples of Defect haricot beans, 237 are correctly classified but there are 13 Defect beans which are wrongly classified as Non-defect.

4.6.2. Experiment on CNN feature Extraction and SVM Classifier

In this experiment, we used CNN for feature extraction and we used SVM for classification, 1500 training samples used as input for SVM classifier. The remaining 500 samples are used for testing purposes. The average accuracy recorded in this experiment was 94%.



```
              precision    recall  f1-score   support

        dhbs       0.95      0.92      0.94       250
       ndhbs       0.93      0.96      0.94       250

    accuracy                           0.94       500
   macro avg       0.94      0.94      0.94       500
weighted avg       0.94      0.94      0.94       500
```

Figure 4.4: Precision, recall, and F1-score of the above experiment.

Figure 4.4 shows that there is 5% false positive when the model predicted Defect Haricot bean, and there is 4% false negative when the model predicted Non-defect. The model also has a 8% false negative rate when it predicts the Defect Haricot bean and a 4% false positive rate when the model predicted Non-defect Haricot bean.

		Prediction value	
		Defect	Non-defect
Actual value	Defect	231	19
	Non-defect	11	239

Table 4.5: Performance evaluation matrix of SVM using CNN features.

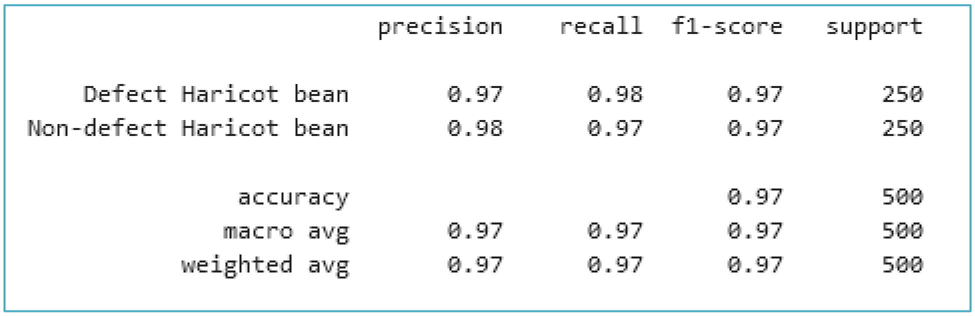
As shown in Table 4.5, Non-defect samples are correctly classified as 239 with 93% precision and 11 is incorrectly classified. Among 250 samples of Defect haricot beans, 231 are correctly classified with 94% accuracy but there are 11 Defect beans which are wrongly classified.

4.6.3. Experiment on GLCM Feature Extraction and RF Classifier

In this experiment, we used GLCM for feature extraction and we used RF for classification, features extracted by GLCM that have 25 features and 1500 training samples used as input for RF classifier. The remaining 500 samples are used for testing purposes. The average accuracy recorded in this experiment was 97%.

parameter	**values**
max_depth	10
max_features	0.3
min_samples_leaf':	10
min_samples_split	0.01
n_estimators	300
max_depth	10

Table 4.6: Optimal parameters of RF based on experiment



```
                         precision    recall  f1-score   support

    Defect Haricot bean       0.97      0.98      0.97       250
Non-defect Haricot bean       0.98      0.97      0.97       250

               accuracy                           0.97       500
              macro avg       0.97      0.97      0.97       500
           weighted avg       0.97      0.97      0.97       500
```

Figure 4.5: Precision, recall, and F1-score of RF with the application of GLCM Features

Figure 4.5 shows that there is a 3% false positive when the model predicted Defect and Non-defect Haricot beans. Similarly, the model also has a 2% false negative rate when it predicts the Defect and Non-defect Haricot beans.

		Defect	Non-defect
Actual value	Defect	244	6
	Non-defect	7	243

Prediction value

Table 4.7: Performance evaluation matrix of RF using GLCM features.

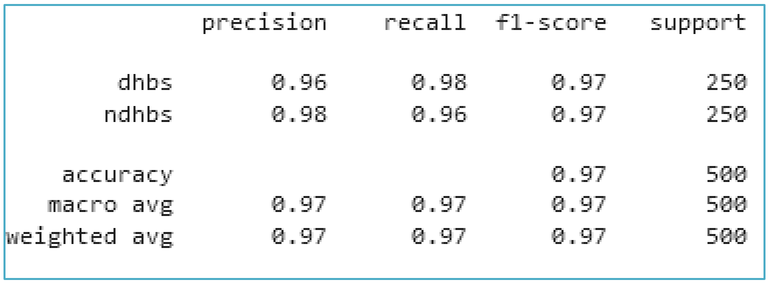
As shown in the confusion matrix, 243 Non-defect Haricot bean samples are correctly classified as Non-defect. Similarly, among 250 samples of Defect Haricot beans, 244 are correctly classified as Defect, but there are 6 Defect Haricot beans which are wrongly classified as Non-defect.

4.6.4. Experiment on CNN Feature Extraction and RF Classifier

In this experiment, we used CNN for feature extraction and we used RF for classification, 1500 training samples used as input for RF classifier. The remaining 500 samples are used for testing purposes. The average accuracy recorded in this experiment was 97%.

Parameter	**Values**
max_depth	10
max_features	0.3
min_samples_leaf':	10
min_samples_split	0.01
n_estimators	300
max_depth	10

Table 4.8: Optimal parameters of RF



```
              precision    recall  f1-score   support

        dhbs       0.96      0.98      0.97       250
       ndhbs       0.98      0.96      0.97       250

    accuracy                           0.97       500
   macro avg       0.97      0.97      0.97       500
weighted avg       0.97      0.97      0.97       500
```

Figure 4.6: Precision, recall, and F1-score of the above experiment.

Figure 4.6 shows that there is a 4% false positive when the model predicted Defect Haricot bean, and there is 4% false negative when the model predicted Non-defect Haricot bean. The model also has a 2% false negative rate when it predicts the Defect Haricot bean, and a 2% false positive rate when the model predicted Non-defect Haricot bean.

		Defect	Non-defect
Actual value	Defect	245	5
	Non-defect	10	240

Prediction value

Table 4.9: Performance evaluation matrix of RF using CNN features.

As shown in Table 4.9, Non-defect samples are correctly classified as 240 with 98% precision and 10 is incorrectly classified. Among 250 samples of Defect haricot beans, 245 are correctly classified with 98% accuracy but there are 10 Defect beans which are wrongly classified as Non-defect.

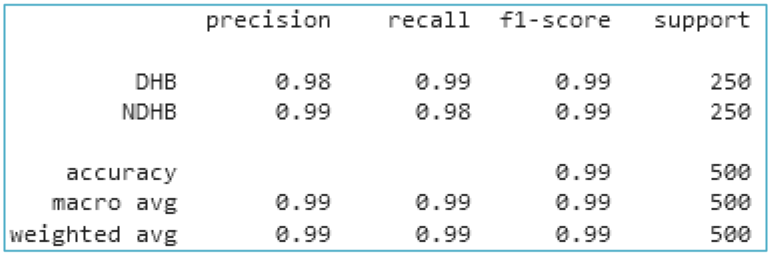
4.6.5 Experiment on CNN as Classifier

In this experiment, we have used a-CNN end - end for classification, 1500 training samples used as input for CNN classifier. The remaining 500 samples are used for testing purposes. The average accuracy recorded in this experiment was 99%.

Actual value	Defect	248	2
	Non-defect	4	246
		Defect	Non-defect
		Prediction value	

Table 4.10: Confusion matrix of the proposed end to end CNN model

As shown in Table 4.10, Non-defect samples are correctly classified as 246 with 98% precision and 4 is incorrectly classified. Among 250 samples of Defect haricot beans, 248 are correctly classified with 99% accuracy but there are 2 Defect beans which are wrongly classified as Non-defect.

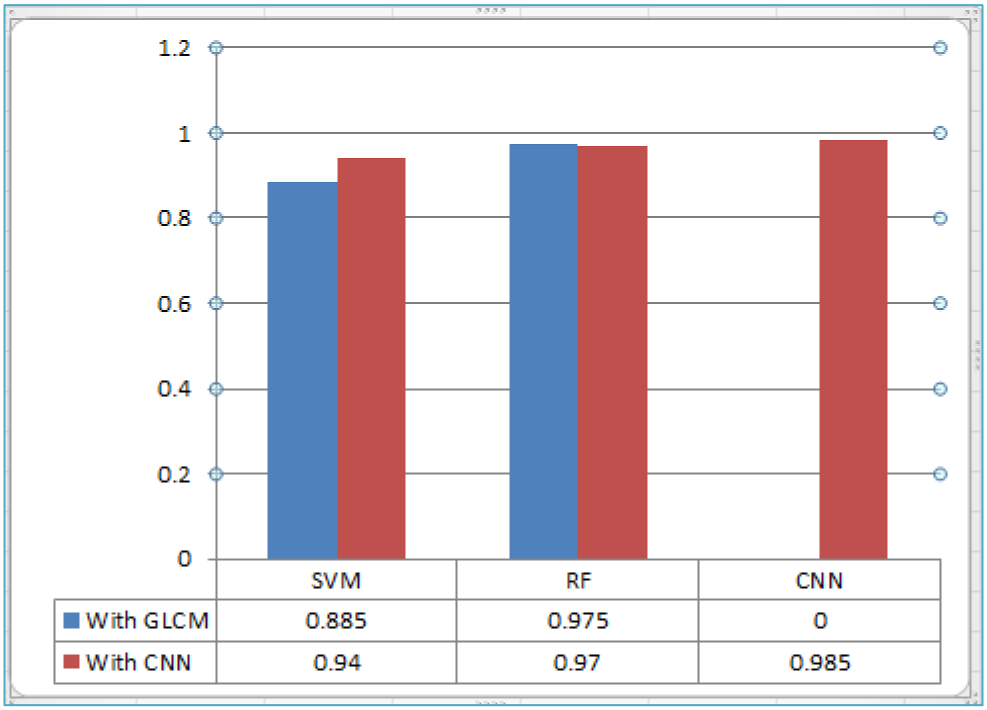


```
              precision    recall  f1-score   support

         DHB       0.98      0.99      0.99       250
        NDHB       0.99      0.98      0.99       250

    accuracy                           0.99       500
   macro avg       0.99      0.99      0.99       500
weighted avg       0.99      0.99      0.99       500
```

Figure 4.7: Precision, recall, and F1-score of CNN

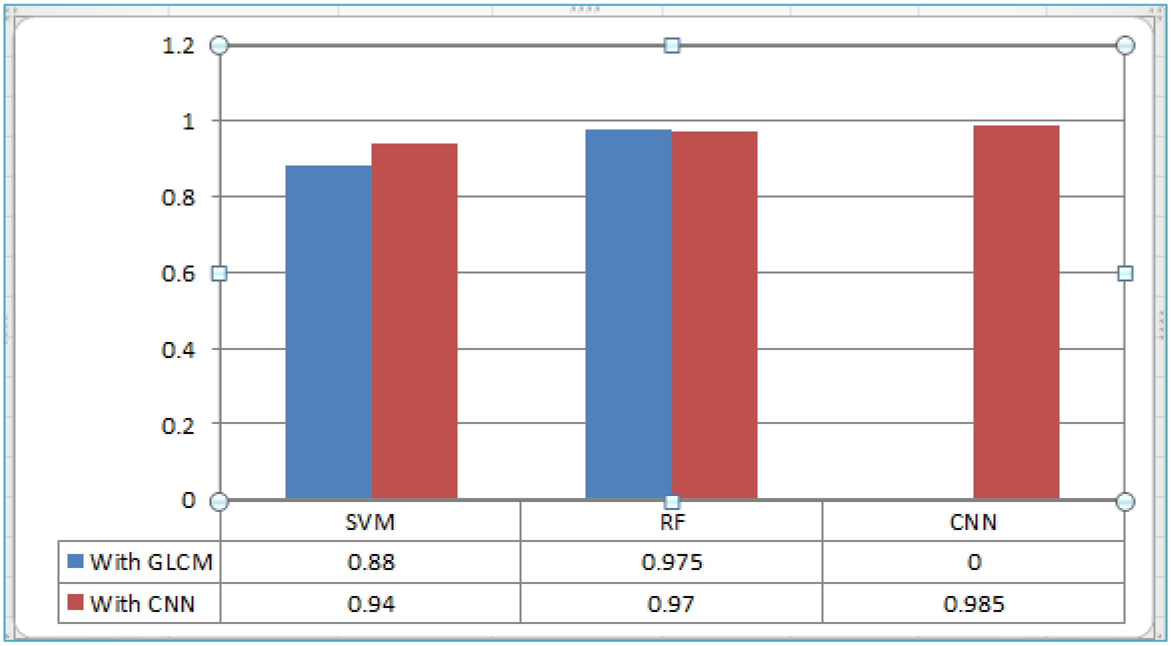
Figure 4.7 shows that there is a 2% false positive when the model predicted Defect Haricot bean, and there is 2% false negative when the model predicted Non-defect Haricot bean. The model also has a 1% false negative rate when it predicts the Defect Haricot bean and a 1% false positive rate when the model predicted Non-defect Haricot bean.

4.6.6. Comparison of CNN and GLCM Feature Extraction for SVM, RF and CNN



	SVM	RF	CNN
🟦 With GLCM	0.885	0.975	0
🟥 With CNN	0.94	0.97	0.985

Figure 4.8: Comparison of SVM, RF, and CNN by the average precision of Defect and Non-defect Haricot beans



	SVM	RF	CNN
🟦 With GLCM	0.88	0.975	0
🟥 With CNN	0.94	0.97	0.985

Figure 4.9: Comparison of SVM, RF, and CNN by the average recall of Defect and Non-defect Haricot beans

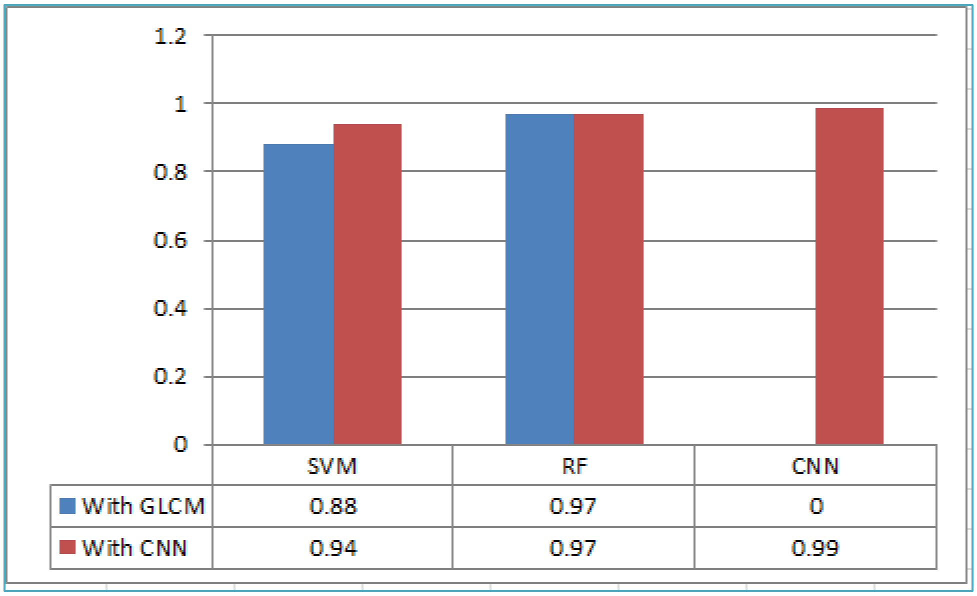


Figure 4.10: Comparison of SVM, RF, and CNN by the average f1-score of Defect and Non-defect Haricot beans

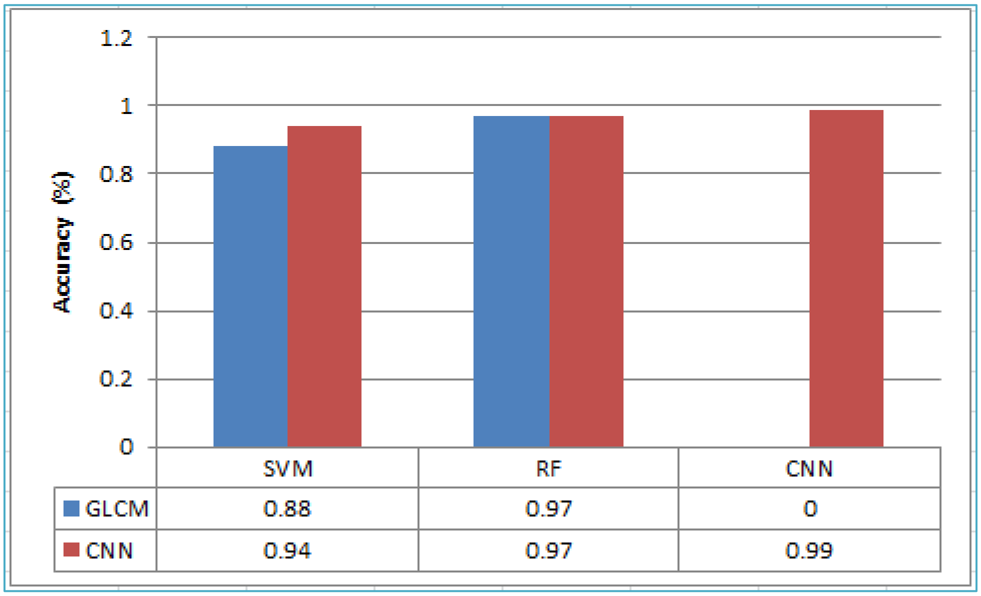


Figure 4.11: Comparison of accuracy of SVM, RF, and CNN

Classifier	Accuracy gained from	
	CNN as Feature Extractor	GLCM texture feature Extractor
SVM	94%	88%
RF	97%	97%
CNN	99%	-

Table 4.11: Evaluation of results accuracy of feature Extracted from CNN, GLCM and end to end CNN as classifier

4.7 Result and Discussion

We carried out experiments to evaluate the proposed model and compare its performance with the performance of existing methods for haricot bean classification.

To analyze the performance of the two feature extraction methods, we applied three types of Classifiers—SVM, Random Forest and CNN compared to the results that from Table 4.9 we can summarize our research results as follows: using features extracted from the CNN method can achieve an accuracy of 94% with the SVM classifier and 97% with the RF classifier. But CNN as a classifier achieved the highest accuracy score which is 99%.

Based on GLCM textural features an accuracy of 88% with the SVM classifier and 97% with the RF classifier achieved.

As can be seen in Table 4.10, the performance of haricot bean defect detection model of the current study shows promising result when used CNN as feature Extraction and RF classier.

The objective of the study was answering the following research questions;

Which feature extraction technique provides the highest performance?

Which machine learning model provides the highest performance?

After conducting different experiments and result analysis the research questions are answered as follows;

- The highest performance was achieved by implementing CNN feature extraction technique.
- CNN is the best performing model it has achieved 99% accuracy.

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1. Conclusion

Haricot bean play important economic, and food and nutrition security roles in Ethiopia. Recently, the production and supply of pulses, increased due to increased demand in both local and international markets, On the other hand, quality inspection of agricultural products is the most important factor affecting world market competence and customer satisfaction. The existing manual inspections have had many problems in maintaining consistency and ensuring satisfactory detection efficiency. For the evaluation of haricot bean defect detection take considerable time, and sometimes the scarce of qualified expert in the area. In addition, subjective errors from manual inspections have their own impact. The development of haricot bean quality inspection and defect detection can minimize all above mentioned problems.

We demonstrated our method by implementing a textural feature extraction of haricot bean and classifications, consisting of two classes of haricot beans (defect and non-defect haricot bean). By utilizing the co-occurrence matrix feature of GLCM, we extracted 25 features. In our method, we also use as CNN feature Extraction and classification.

To this end, the study follows experimental research, which involves data set preparation for training and evaluating haricot bean model. Image of haricot bean (Defect and Non-defect haricot bean) are have been collected from ECX in Bure branch Amhara region. The images are preprocessed and followed by feature extraction. Features extracted from the CNN method can achieve an accuracy of 94% and 97% for SVM and RF classifier respectively. Based on GLCM textural features, method can achieve an accuracy of 88% and 97% for SVM and RF classification accuracy respectively. We had also experiments using CNN used as a classifier method, and which achieved an accuracy of 99%.From our result we can conclude that our method utilizing the CNN features extraction methods as one part of the feature extraction process from haricot bean images, can achieve promising results compared to GLCM texture feature extraction method.

5.2. Recommendation

As haricot bean is one of the pulse product that Ethiopia exports to different countries of the world, such research work can facilitate haricot beans to be easily competitive to the market and increase the export earnings. This study is therefore beneficial for haricot bean exporters and farmers. In this study, we have proposed a model for quality inspection and defect detection for red haricot bean, and we classified using two classes: defect and Non-defect haricot beans. Therefore, our recommendation is to use moisture content as additional quality inspection parameters .We also recommends working on the other types of haricot beans such as white beans and speckled beans.

References

Adhitya, Y., Prakosa, S. W., Köppen, M., & Leu, J. S. (2020). Feature extraction for cocoa bean digital image classification prediction for smart farming application. *Agronomy*, *10*(11). https://doi.org/10.3390/agronomy10111642

Araújo, S. A. De, Pessota, J. H., & Kim, H. Y. (2015). Beans quality inspection using correlation-based granulometry. *Engineering Applications of Artificial Intelligence*, *40*, 84–94. https://doi.org/10.1016/j.engappai.2015.01.004

Dawit & Gerdien. (2010). The Ethiopian Commodity Exchange (ECX): An overview. *EPOSPEA & Wageningen UR*, *June*, 34.

Epherem Terefe. (2016). Review of Haricot bean Value Chain in Ethiopia. *Journal of Asian and African Studies* , *24*, 65–72. www.iiste.org

Ferris, S., & Kaganzi, E. (2008). Evaluating marketing opportunities for haricot beans in Ethiopia. *International Center for Tropical Agriculture*, 68.

Galván, I. M., Valls, J. M., Lecomte, N., & Isasi, P. (2009). A lazy approach for machine learning algorithms. *IFIP International Federation for Information Processing*, *296*, 517–522. https://doi.org/10.1007/978-1-4419-0221-4_60

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture*, *116*, 8–19. https://doi.org/10.1016/j.compag.2015.05.021

Hambal, A. M., Pei, Z., & Ishabailu, F. L. (2017). *Image Noise Reduction and Filtering Techniques*. *6*(3), 2033–2038. https://doi.org/10.21275/25031706

Hameed, A. A., Karlik, B., & Salman, M. S. (2016). Back-propagation algorithm with variable adaptive momentum. *Knowledge-Based Systems*, *114*, 79–87. https://doi.org/10.1016/j.knosys.2016.10.001

Hasan, H., Shafri, H. Z. M., & Habshi, M. (2019). A Comparison between Support Vector Machine (SVM) and Convolutional Neural Network (CNN) Models for Hyperspectral Image Classification. *IOP Conference Series: Earth and Environmental Science*, *357*(1). https://doi.org/10.1088/1755-1315/357/1/012035

Hossain, M. A., & Alam Sajib, M. S. (2019). Classification of Image using Convolutional Neural Network (CNN). *Global Journal of Computer Science and Technology*, *19*(2), 13–18. https://doi.org/10.34257/gjcstdvol19is2pg13

Inácio, D., & Rieder, R. (2018). *Computer vision and arti fi cial intelligence in precision agriculture for grain crops : A systematic review*. *153*(June), 69–81. https://doi.org/10.1016/j.compag.2018.08.001

Jyoti Bora, D. (2017). Importance of Image Enhancement Techniques in Color Image Segmentation: a Comprehensive and Comparative Study. *ArXiv*.

Kabata, A., Henry, C., Moges, D., Kebebu, A., Whiting, S., Regassa, N., & Tyler, R. (2016). Determinants and Constraints of Pulse Production and Consumption among Farming Households of Ethiopia. *Journal of Food Research*, *6*(1), 41. https://doi.org/10.5539/jfr.v6n1p41

Kaitlin, ;, Smith, T. ;, & Sadler, B. (2018). Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. *Recommended Citation Kirasich*, *1*(3), 9. https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol1/iss3/9

Kavitha, J. C., & Suruliandi, A. (2016). Texture and color feature extraction for classification of melanoma using SVM. *2016 International Conference on Computing Technologies and Intelligent Data Engineering, ICCTIDE 2016*. https://doi.org/10.1109/ICCTIDE.2016.7725347

Kebede, B., Korji, D., & Amare, G. (2018). Participatory Evaluation and Selection of Improved Haricot bean Varieties at Liben District, Lowland Agro Ecology of Guji Zone, Oromia Regional State, Ethiopia. *Advances in Crop Science and Technology*, *06*(01), 1–4. https://doi.org/10.4172/2329-8863.1000332

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6

Mahajan, S., Das, A., & Sardana, H. K. (2015). Image acquisition techniques for assessment of legume quality. *Trends in Food Science and Technology*, *42*(2), 116–133.

https://doi.org/10.1016/j.tifs.2015.01.001

Mauricio, Candelo-Becerra, J. E., & Hoyos, F. E. (2019). Quality and defect inspection of green coffee beans using a computer vision system. *Applied Sciences (Switzerland)*, *9*(19). https://doi.org/10.3390/app9194195

Miklas, P. N., Kelly, J. D., Beebe, S. E., & Blair, M. W. (2006). *Common bean breeding for resistance against biotic and abiotic stresses : From classical to MAS breeding Common bean breeding for resistance against biotic and abiotic stresses : From classical to MAS breeding*. *January*. https://doi.org/10.1007/s10681-006-4600-5

Mohamed, A., Karrar, A., & Sun, J. (2018). Some Of The Techniques for Images Processing Using Matlab. *American Journal of Engineering Research*, *7*(9), 173–184.

Momin, M. A., Yamamoto, K., Miyamoto, M., Kondo, N., & Grift, T. (2017). Machine vision based soybean quality evaluation. *Computers and Electronics in Agriculture*, *140*, 452–460. https://doi.org/10.1016/j.compag.2017.06.023

Öztürk, Ş., & Akdemir, B. (2018). Application of Feature Extraction and Classification Methods for Histopathological Image using GLCM, LBP, LBGLCM, GLRLM and SFTA. *Procedia Computer Science*, *132*(Iccids), 40–46. https://doi.org/10.1016/j.procs.2018.05.057

Patil, A. B., & Shaikh, J. (2016). OTSU thresholding method for flower image segmentation. *International Journal of Computational Engineering Research (IJCER)*, *6*(5), 1–6. http://paper.researchbib.com/view/paper/92912

Pinto, C., Furukawa, J., Fukai, H., & Tamura, S. (2017). Classification of Green coffee bean images basec on defect types using convolutional neural network (CNN). *Proceedings - 2017 International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA 2017*. https://doi.org/10.1109/ICAICTA.2017.8090980

Pradeep, A. G., & Gnanapriya, M. (2016). *Using HSV Color Space*. *4*, 5072–5075.

Priyanka C. Dighe, S. K. G. (2014). Survey on Image Resizing Techniques. *International Journal of Science and Research (IJSR)*, *3*(12), 1444–1448. https://www.ijsr.net/archive/v3i12/U1VCMTQ3MjI=.pdf

Ramirez-Loaiza, M. E., Sharma, M., Kumar, G., & Bilgic, M. (2017). Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, *31*(2), 287–313. https://doi.org/10.1007/s10618-016-0469-7

S.Yuheng. (2017). Image Segmentation Algorithms. *Architectures and Algorithms for Digital Image Processing II*, *0534*, 172. https://doi.org/10.1117/12.946577

Saha, D., & Manickavasagan, A. (2021). Machine learning techniques for analysis of hyperspectral images to determine quality of food products: A review. *Current Research in Food Science*, *4*(December 2020), 28–44. https://doi.org/10.1016/j.crfs.2021.01.002

Schmitt, J., Bönig, J., Borggräfe, T., Beitinger, G., & Deuse, J. (2020). Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing. *Advanced Engineering Informatics*, *45*(May 2019). https://doi.org/10.1016/j.aei.2020.101101

Sharma, E. K., Priyanka, E., Kalsh, E. A., & Saini, E. K. (2015). GLCM and its Features. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, *4*(8), 2180–2182.

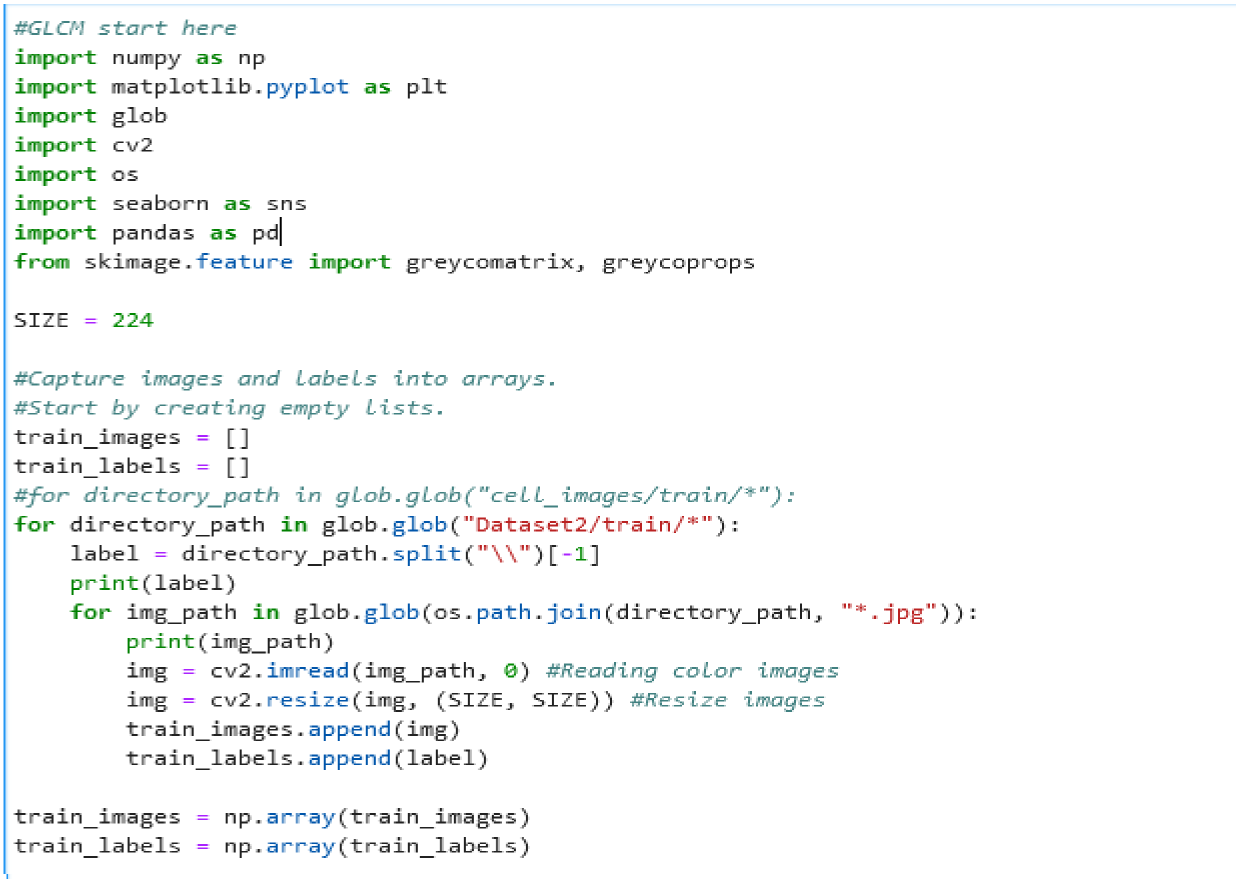
Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation —A review. *Information Processing in Agriculture*, *7*(1), 1–19. https://doi.org/10.1016/j.inpa.2019.09.006

WFP. (2011). *Ethiopia haricot beans*. 3–5.

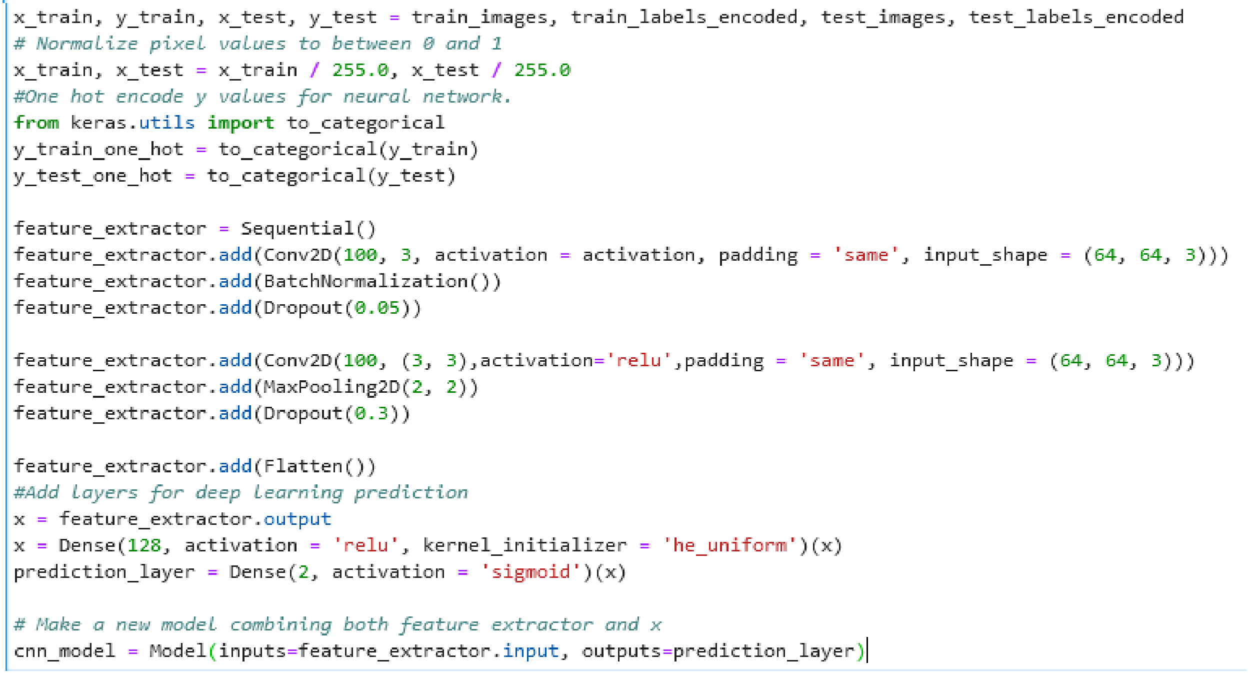
Wondimu, W., & Bogale, A. (2017). Genetic variability, heritability and genetic advance of some haricot bean (Phaseolus vulgaris L.) varieties at Bench-Maji Zone, Southwest Ethiopia. *Asian Journal of Crop Science*, *9*(4), 133–140. https://doi.org/10.3923/ajcs.2017.133.140

Yogesh, Dubey, A. K., Ratan, R., & Rocha, A. (2020). Computer vision based analysis and detection of defects in fruits causes due to nutrients deficiency. *Cluster Computing*, *23*(3), 1817–1826. https://doi.org/10.1007/s10586-019-03029-6

Appendix A: CNN code for feature Extraction

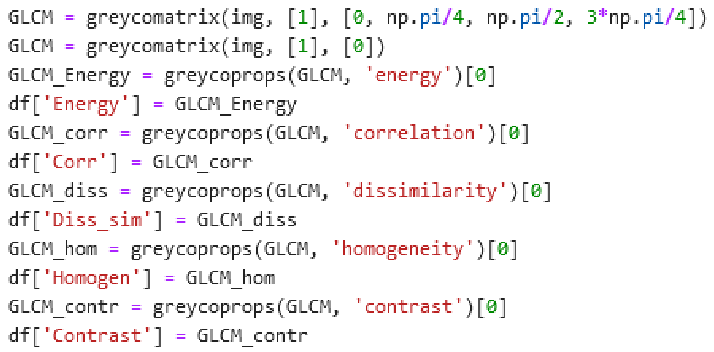


```
#GLCM start here
import numpy as np
import matplotlib.pyplot as plt
import glob
import cv2
import os
import seaborn as sns
import pandas as pd
from skimage.feature import greycomatrix, greycoprops

SIZE = 224

#Capture images and labels into arrays.
#Start by creating empty lists.
train_images = []
train_labels = []
#for directory_path in glob.glob("cell_images/train/*"):
for directory_path in glob.glob("Dataset2/train/*"):
    label = directory_path.split("\\")[-1]
    print(label)
    for img_path in glob.glob(os.path.join(directory_path, "*.jpg")):
        print(img_path)
        img = cv2.imread(img_path, 0) #Reading color images
        img = cv2.resize(img, (SIZE, SIZE)) #Resize images
        train_images.append(img)
        train_labels.append(label)

train_images = np.array(train_images)
train_labels = np.array(train_labels)
```

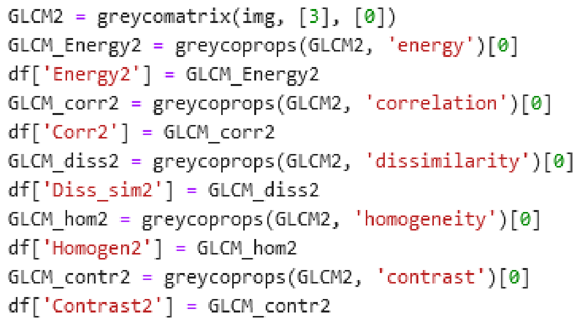


```
x_train, y_train, x_test, y_test = train_images, train_labels_encoded, test_images, test_labels_encoded
# Normalize pixel values to between 0 and 1
x_train, x_test = x_train / 255.0, x_test / 255.0
#One hot encode y values for neural network.
from keras.utils import to_categorical
y_train_one_hot = to_categorical(y_train)
y_test_one_hot = to_categorical(y_test)

feature_extractor = Sequential()
feature_extractor.add(Conv2D(100, 3, activation = activation, padding = 'same', input_shape = (64, 64, 3)))
feature_extractor.add(BatchNormalization())
feature_extractor.add(Dropout(0.05))

feature_extractor.add(Conv2D(100, (3, 3),activation='relu',padding = 'same', input_shape = (64, 64, 3)))
feature_extractor.add(MaxPooling2D(2, 2))
feature_extractor.add(Dropout(0.3))

feature_extractor.add(Flatten())
#Add layers for deep learning prediction
x = feature_extractor.output
x = Dense(128, activation = 'relu', kernel_initializer = 'he_uniform')(x)
prediction_layer = Dense(2, activation = 'sigmoid')(x)

# Make a new model combining both feature extractor and x
cnn_model = Model(inputs=feature_extractor.input, outputs=prediction_layer)
```

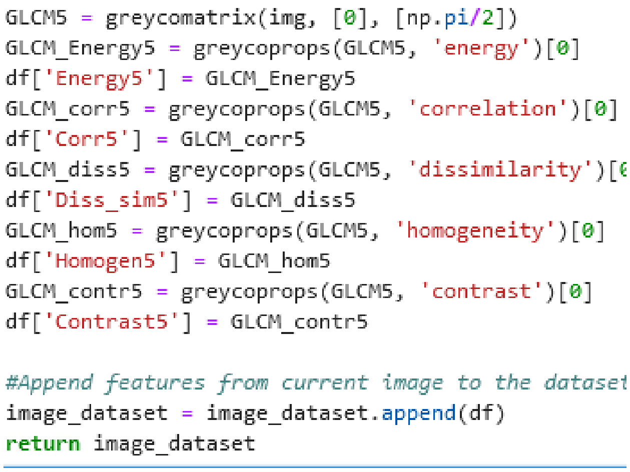
Appendix B: Code for GLCM Texture Feature Extraction



```
GLCM = greycomatrix(img, [1], [0, np.pi/4, np.pi/2, 3*np.pi/4])
GLCM = greycomatrix(img, [1], [0])
GLCM_Energy = greycoprops(GLCM, 'energy')[0]
df['Energy'] = GLCM_Energy
GLCM_corr = greycoprops(GLCM, 'correlation')[0]
df['Corr'] = GLCM_corr
GLCM_diss = greycoprops(GLCM, 'dissimilarity')[0]
df['Diss_sim'] = GLCM_diss
GLCM_hom = greycoprops(GLCM, 'homogeneity')[0]
df['Homogen'] = GLCM_hom
GLCM_contr = greycoprops(GLCM, 'contrast')[0]
df['Contrast'] = GLCM_contr
```

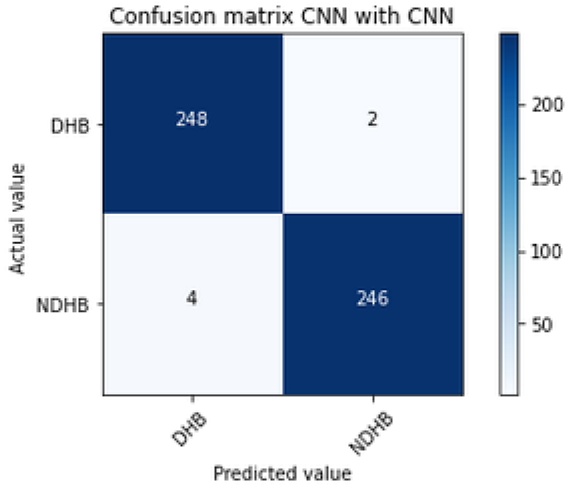


```
GLCM2 = greycomatrix(img, [3], [0])
GLCM_Energy2 = greycoprops(GLCM2, 'energy')[0]
df['Energy2'] = GLCM_Energy2
GLCM_corr2 = greycoprops(GLCM2, 'correlation')[0]
df['Corr2'] = GLCM_corr2
GLCM_diss2 = greycoprops(GLCM2, 'dissimilarity')[0]
df['Diss_sim2'] = GLCM_diss2
GLCM_hom2 = greycoprops(GLCM2, 'homogeneity')[0]
df['Homogen2'] = GLCM_hom2
GLCM_contr2 = greycoprops(GLCM2, 'contrast')[0]
df['Contrast2'] = GLCM_contr2
```

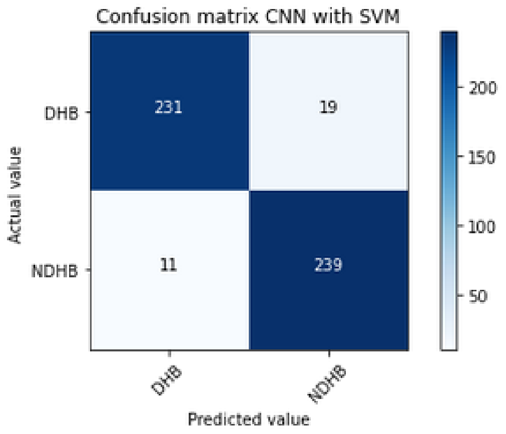


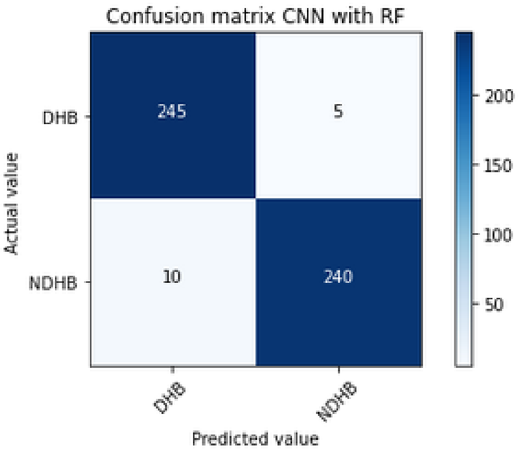
```
GLCM5 = greycomatrix(img, [0], [np.pi/2])
GLCM_Energy5 = greycoprops(GLCM5, 'energy')[0]
df['Energy5'] = GLCM_Energy5
GLCM_corr5 = greycoprops(GLCM5, 'correlation')[0]
df['Corr5'] = GLCM_corr5
GLCM_diss5 = greycoprops(GLCM5, 'dissimilarity')[0
df['Diss_sim5'] = GLCM_diss5
GLCM_hom5 = greycoprops(GLCM5, 'homogeneity')[0]
df['Homogen5'] = GLCM_hom5
GLCM_contr5 = greycoprops(GLCM5, 'contrast')[0]
df['Contrast5'] = GLCM_contr5

#Append features from current image to the dataset
image_dataset = image_dataset.append(df)
return image_dataset
```

Appendix C: Confusion matrix of CNN end to end classifier

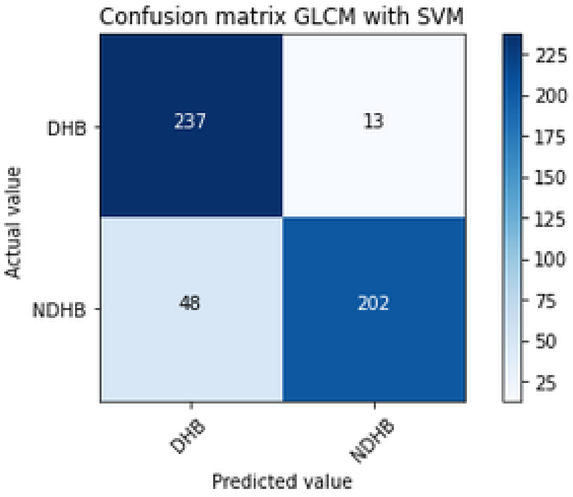


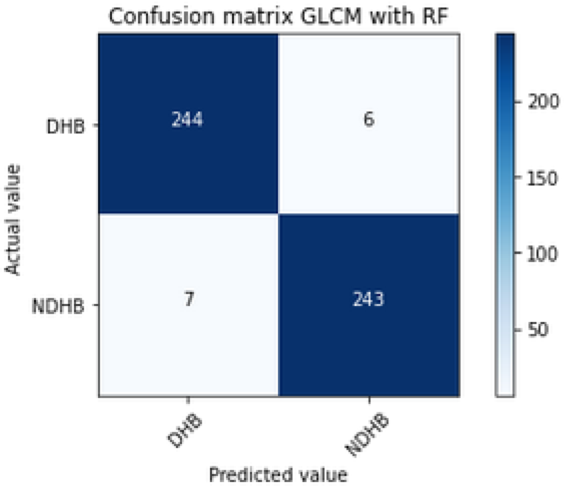
Appendix D: Confusion matrix of CNN as feature extraction SVM and RF as classifier





Appendix E: Confusion matrix of GLCM as feature extraction SVM and RF as classifier


Confusion matrix GLCM with SVM
DHB
NDHB
Actual value
DHB
NDHB
Predicted value


Confusion matrix GLCM with RF
DHB
NDHB
Actual value
DHB
NDHB
Predicted value