



**DEBRE BERHAN UNIVERSITY**  
**COLLEGE OF COMPUTING**  
**DEPARTMENT OF INFORMATION SYSTEMS**  
**IMAGE BASED COFFEE BEAN CLASSIFICATION**  
**USING DEEP LEARNING TECHNIQUE.**

**By**  
**GEBREYES GEBEYEHU GELETA**

**DEBRE BERHAN, ETHIOPIA**

**JUNE 2021**

**DEBRE BERHAN UNIVERSITY**  
**COLLEGE OF COMPUTING**  
**DEPARTMENT OF INFORMATION SYSTEMS**

**IMAGE BASED COFFEE BEAN CLASSIFICATION**  
**USING DEEP LEARNING TECHNIQUE.**

A THESIS SUBMITTED TO THE DEPARTMENT OF INFORMATION SYSTEMS OF  
DEBRE BERHAN UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENT  
FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SYSTEMS

By  
GEBREYES GEBEYEHU GELETA

DEBRE BERHAN, ETHIOPIA  
JUNE 2021

**DEBRE BERHAN UNIVERSITY**  
**COLLEGE OF COMPUTING**  
**DEPARTMENT OF INFORMATION SYSTEMS**

**IMAGE BASED COFFEE BEAN CLASSIFICATION  
USING DEEP LEARNING TECHNIQUE.**

By

GEBREYES GEBEYEHU GELETA

Name and signature of members of the examining board

<b>Title</b>	<b>Name</b>	<b>Signature</b>	<b>Date</b>
Advisor	Michael Melese (PhD)	-----	-----
Chair Person	-----	-----	-----
External Examiner	-----	-----	-----
Internal Examiner	-----	-----	-----

DEBRE BERHAN, ETHIOPIA

JUNE 2021

**DEBRE BERHAN UNIVERSITY**  
**COLLEGE OF COMPUTING**  
**DEPARTMENT OF INFORMATION SYSTEMS**

This is to declare that the thesis prepared by Gebreyes Gebeyehu, titled: **Image Based Coffee Bean Classification Using Deep Learning Technique** and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Systems fulfills with the regulations of the University and meets the accepted standards with respect to originality and quality.

---

GEBREYES GEBEYEHU GELETA

JUNE 2021

This thesis has been submitted for examination with our approval as university advisor.

---

Advisor: Michael Melese (PhD)

JUNE 2021

## **ACKNOWLEDGEMENT**

As usual, I'm not sure what I did to deserve your blessing. I don't believe we should spend our lives praying for things since God already knows what we deserve. But I believe I owe it to the almighty GOD to thank Him for what He has bestowed upon me. I'd like to express my gratitude to you, Lord, for life and everything that comes with it. Thank you for the day, the hour, and the minute that you have given us. I'd would like express my deepest gratitude to my advisor, Dr. Michael Melese, for his boundless support, pleasant working atmosphere, and helpful counsel, particularly for his sociability. Whenever I had a question, he was always available. He continually made it possible for me to be productive and guided me in the proper direction. Thank you for everything and wish you happy life. I must express my deepest gratitude for the experts who were involved in this research Mr. Silesh and other Buna board employees. Finally, I want to express my gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and my postgraduate lovely classmates.

---

GEBREYES GEBEYEHU GELETA

JUNE 2021

## ABSTRACT

Coffee is one of the most famous gentle drink within the world. Most peoples take a cup of or double cup of coffee every morning to stimulate them even though they drink after the launch and totally, over 2.25 billion cups are fed on every day. Monetarily, coffee is the second most sent out commodity after oil, and utilizes more than 100 million individuals around the world. Coffee Arabica has developed for thousands of years in Ethiopia, in the southwestern highlands forests.

The classification and grading of coffee in Ethiopian coffee quality inspection and certification center or Coffee board is manual. This leads to so many problems like pruning of error, inefficient, require a lot of labor and is not cost effective. This research was conducted with the objective of developing an appropriate computer routine algorithm that can characterize different varieties of coffee based on their growing region.

To address this problem, we proposed deep learning approach for classification of coffee bean based on their growing regions. The proposed system has two main components namely, training model and testing the trained model or developing web application using flask. In this study, we applied different image preprocessing techniques such as: - removing noise, normalizing images and resizing images. We proposed two novel segmentation algorithms which are used to extract region of interest (ROI) and both are achieved excellent accuracy in this study. Two frameworks are proposed: one is to train a deep neural network model from scratch, and the other is to transfer learning a pre-trained network model. The model with best performance were obtained by testing different network layers, optimizers, learning rates, loss functions, number of epoch, batch sizes, and steps. Based on the optimized model.

The developed classification model trained on 3120 dataset's collected from ECQIAC. To increase the dataset, we applied different augmentation techniques. We split the dataset into different test options such as: - 90:10, 80:20 and 70:30 for training and testing respectively. The model classified the input coffee bean image into Gujji, Jimma, Kaffa, Nekempti, Sidamo and Yirgacheffe using softmax classifier function with 97.8 % accuracy. The entire system has been evaluated by employee from ECQIAC, the analysis shows that the system is effective to classify green coffee bean based on their growing region.

**Keywords:** Ethiopian coffee, Coffee Bean Classification, Deep Learning, Convolutional Neural Network (CNN).

# Table of Contents

<b>ACKNOWLEDGEMENT</b> .....	i
<b>ABSTRACT</b> .....	ii
<b>LIST OF FIGURES</b> .....	vi
<b>LIST OF TABLES</b> .....	viii
<b>LIST OF ABBREVIATIONS</b> .....	ix
<b>CHAPTER ONE</b> .....	1
<b>INTRODUCTION</b> .....	1
1.1. Background .....	1
1.2. Motivation of Research.....	3
1.3. Problem Statement .....	4
1.4. Objectives of the study.....	6
1.4.1. General objective .....	6
1.4.2. Specific Objectives .....	6
1.5. Significance of the Study.....	6
1.6. Scope and Limitation of the study .....	7
1.7. Organization of the Thesis .....	7
<b>CHAPTER TWO</b> .....	9
<b>LITERATURE REVIEW</b> .....	9
2.1. Overview .....	9
2.2. Coffee in Ethiopia .....	9
2.2.1. Classifications and Grading of Ethiopian Coffee .....	9
2.2.2. Coffee Varieties in Ethiopia.....	12
2.2.3. Ethiopian Coffee Processing.....	17
2.3. Image Representation.....	19
2.4. Image Processing .....	22
2.4.1. Image Acquisition.....	22
2.4.2. Image Pre-processing.....	23
2.4.3. Image segmentation .....	27
2.4.4. Feature Extraction .....	31
2.5. Machine Learning .....	33
2.5.1. Classification approaches.....	33

2.5.2. Artificial Neural Network .....	34
2.6. Deep Learning Approach .....	35
2.6.1. Recurrent Neural Network (RNN).....	36
2.6.2. Long Short-Term Memory (LSTM) Neural Network .....	37
2.6.3. Convolutional Neural Network.....	38
2.6.4. Examples of CNN Architectures .....	48
2.7. Related Works.....	51
2.7.1. Related to Coffee and Crop Product .....	51
2.7.2. Related to convolutional neural network approach .....	55
<b>CHAPTER THREE .....</b>	<b>59</b>
<b>SYSTEM DESIGN AND ARCHITECTURE .....</b>	<b>59</b>
3.1. Overview .....	59
3.2. Design Science Research Methodology.....	59
3.3. The Proposed System Architecture.....	62
3.3.1. Image Acquisition .....	64
3.3.2. Image Preprocessing .....	64
3.3.3. Image Segmentation.....	67
3.3.4. Data augmentation .....	69
3.3.5. Normalization .....	69
3.3.6. Convolutional Neural Network Model Architecture .....	70
3.3.7. Hyperparameters .....	73
3.3.8. Classification.....	74
3.4. Model Evaluation Metrics.....	74
<b>CHAPTER FOUR.....</b>	<b>76</b>
<b>EXPERIMENT AND EVALUATION.....</b>	<b>76</b>
4.1. Introduction.....	76
4.2. Description and Preparation of Dataset .....	76
4.3. Experimental Design.....	77
4.4. Simulation Environment .....	77
4.5. Experimental Results .....	78
4.5.1. Experiments on Training from the scratch.....	78
4.5.2. Training and testing state-of-the-art CNN architecture .....	84



4.6. Summary of performance comparison.....	87
4.7. Comparison of Related Works with this Study.....	88
4.8. User Acceptance Testing .....	89
4.9. Discussion of the Research Questions .....	91
4.10. Web application design and Implementation.....	92
<b>CHAPTER FIVE .....</b>	<b>97</b>
<b>CONCLUSION AND RECOMMENDATION .....</b>	<b>97</b>
5.1. Conclusion .....	97
5.2. Contribution of this study .....	98
5.3. Recommendation .....	98
<b>REFERENCES.....</b>	<b>100</b>
<b>APPENDIX A: .....</b>	<b>106</b>
<b>APPENDIX B: .....</b>	<b>107</b>

## List of Figures

Figure 2. 1 Moisture Apparatus .....	10
Figure 2. 2 Screen Size Apparatus .....	11
Figure 2. 3 Sidamo coffee bean image.....	13
Figure 2. 4 Yirgacheffe Coffee Bean Image .....	14
Figure 2. 5 Jimma Coffee Bean Image .....	14
Figure 2. 6 Kaffa Coffee Bean Image .....	15
Figure 2. 7 Nekempti Coffee Bean Image .....	16
Figure 2. 8 Gujji Coffee Bean Image.....	16
Figure 2. 9 Digitization of a continuous image.....	20
Figure 2. 10 Fundamentals steps of digital image processing .....	22
Figure 2. 11 White Color Pixel Value Corrupted with Salt & Pepper Noise .....	25
Figure 2. 12 classification of image segmentation techniques .....	28
Figure 2. 13 Watershed segmentation process.....	29
Figure 2. 14 Otsu-Thresholding.....	30
Figure 2. 15 Classification of Feature Extraction Method.....	32
Figure 2. 16 Schematic of a typical Artificial Neural Network (ANN) architecture.....	35
Figure 2. 17 Venn diagram which describes deep learning .....	36
Figure 2. 18 RNN structure.....	37
Figure 2. 19 Long Short-Term Memory neural network .....	38
Figure 2. 20 Architecture of convolutional neural network.....	39
Figure 2. 21 convolution operation using same padding and one stride.....	40
Figure 2. 22 Max Pooling .....	41
Figure 2. 23 ADAM algorithms.....	47
Figure 2. 24 AlexNet Architecture.....	49
Figure 2. 25 VGGNet architecture.....	50
Figure 2. 26 Inception module, naïve version.....	51
Figure 2. 27 Inception module with dimensionality reduction .....	51
Figure 3. 1 Design Science Research Process Model Adopted From [63].....	60
Figure 3. 2 Design Science Research Framework .....	62
Figure 3. 3 Proposed System Process .....	63

Figure 3. 4 Image resizing Algorithm.....	65
Figure 3. 5 Resized Image .....	65
Figure 3. 6 Algorithm for median filtering .....	66
Figure 3. 7 Enhanced Imaged Through Median Filtering.....	66
Figure 3. 8 Image after watershed segmentation was applied .....	68
Figure 3. 9 Coffee bean Image after Otsu applied .....	69
Figure 3. 10 Image normalization algorithm .....	70
Figure 3. 11 Proposed CNN classification model.....	71
Figure 3. 12 Customized pretrained model.....	73
Figure 4. 1 Graph of training and Test accuracy for experiment one .....	78
Figure 4. 2 Training and Test Loss for experiment one.....	79
Figure 4. 3 Graph of training and Test accuracy for experiment two.....	81
Figure 4. 4 Graph of training and Test Loss for experiment two .....	81
Figure 4. 5 Trained Model Sample Screenshot of experiment three .....	83
Figure 4. 6 Graph of training and Test accuracy for experiment three.....	84
Figure 4. 7 Graph of training and Test loss for experiment three.....	84
Figure 4. 10 sample Output of VGG19.....	86
Figure 4. 11 Screenshot image of VGG16 classification report .....	87
Figure 4. 13 Command prompt window to launch Flask server and to get IP address .....	93
Figure 4. 14 Command prompt window to launch Flask server and to get IP address .....	93
Figure 4. 15 Home page of proposed classification system.....	94
Figure 4. 16 Uploading coffee bean image from local disk.....	94
Figure 4. 17 Home page after the coffee bean image uploaded.....	95
Figure 4. 18 Sample Result of classification system .....	95
Figure 4. 19 uploading sample Gujji image.....	96
Figure 4. 20 uploaded Gujji image interface .....	96
Figure 4. 21 Classification Result.....	96

## List of Tables

Table 2. 1 Coffee Bean Characteristics.....	17
Table 2. 2 Class index.....	26
Table 2. 3 One-Hot encoding.....	27
Table 2. 4 Descriptions of segmentation Techniques .....	30
Table 2. 5 Summary of related work .....	57
Table 3. 1 Total number of Image dataset .....	64
Table 3. 2 Label Encoding .....	67
Table 3. 3 One-Hot Encoding .....	67
Table 3. 4 Hyperparameters Description .....	74
Table 3. 5 Confusion Matrix.....	75
Table 4. 1 Confusion matrix of experiment one .....	79
Table 4. 2 Performance result of experiment one .....	79
Table 4. 3 Confusion matrix of experiment two.....	80
Table 4. 4 Performance results of experiment two .....	80
Table 4. 5 Confusion matrix of experiment three .....	82
Table 4. 6 Confusion matrix for binary thresholding test size 0.2.....	83
Table 4. 7 confusion matrix of VGG19 .....	85
Table 4. 8 classification report of VGG19.....	86
Table 4. 9 Comparison Table of All Experiment.....	88
Table 4. 10 Comparison of Related Works with this Study .....	89
Table 4. 11 User Acceptance Evaluation Criteria and Their Results .....	90

## List of Abbreviations

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
CCD	Charge Coupled Device
CNN	Convolutional Neural Network
CSS	Cascading Style Sheet
DIP	Digital Image Processing
DNs	Digital Numbers
DS	Design Science
DSRM	Design Science Research Methodology
ECQIAC	Ethiopian Coffee Quality Inspection and Auction Center
ECX	Ethiopian Commodity Exchange
FIFO	First in First Out
GDP	Gross Domestic Product
GPU	Graphics Processing Unit
HIS	Hue, Saturation and Intensity
HOG	Histogram Oriented Gradient
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ILSVRC	ImageNet LargeScale Visual Recognition Challenge
JPEG	Joint Photographer Expert Group
LBP	Local Binary Pattern
ML	Machine Learning
PDE	Partial Differential Equations
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SURF	Speeded Up Robust Features
URL	Uniform Resource Locator
VGG	Visual Geometry Group
WSGI	Web Server Gateway Interface

# CHAPTER ONE

## INTRODUCTION

### 1.1. Background

The agricultural sector in Ethiopia represents 45 % of the gross domestic product and about 85 % of the population gains their livelihood directly or indirectly from agricultural production including livestock. The importance of agricultural research and its impact on development in Ethiopia can hardly be over emphasized [1]. Relative to other African countries, agricultural research in Ethiopia is quite young. Organized agricultural research activities and actual relations between agricultural research and development started with the inception of the Institute of Agricultural Research in 1966.

Coffee is one of the most famous gentle drink within the world. Most peoples take a cup of or double cup of coffee every morning to stimulate them even though they drink after the launch and totally, over 2.25 billion cups are fed on every day [2]. Monetarily, coffee is the second most sent out commodity after oil, and utilizes more than 100 million individuals around the world [3].

Many nations ' economies depend on espresso production for stabilization and development. Coffee is now cultivated in over 60 tropical nations around the world, and accounts for a significant portion of many's foreign exchange earnings. Ethiopia is Coffee Arabica's Motherland. It has a wide selection of coffee and its different roots. Ethiopian coffee is rich with original taste and aroma because of the country's geographic (altitude, soil, temperature, rainfall, topography, ecology), genotypes, and cultural range. Coffee has developed for thousands of years in Ethiopia, in the southwestern highlands forests. Kaffa coffee drives word, name of a spot in the highlands of south west Ethiopia where coffee is first found. It is also considered to be the first African exporter of Coffee Arabica, and is currently the world's fifth largest espresso maker. There are exceptional varieties of coffee Arabica which grows within Ethiopia. The criteria used for coffee class include bean size, shape, colour, acidity of the aroma, taste and body [4].

Ethiopia produces premium quality coffee according to the international coffee organization's survey, the county of origin for crop. After Brazil, Vietnam, Colombia and Indonesia, it is the leading producer in Africa and the 5<sup>th</sup> in the world. When we look at Arabica alone, after Brazil and Colombia, Ethiopia is the third-largest producer. Ethiopia also has the largest highland region suitable for production in coffee Arabica and therefore has the potential to be a leading producer in terms of both quality and quantity. Nearly, all coffees produced in Ethiopia are color grown, with 40-60% canopy cover, besides few home garden systems in Eastern Ethiopia. The coffee vegetation is also specially either neighborhood varieties/ land races or of wild origin. The chemical inputs for manufacturing are very low and even non-existence in most cases, while processing involves both the wet and dry methods.

The dominant technique but is the dry (natural) approach, with low environmental impact. Despite its importance for millions of people worldwide, coffee production is currently limited by abiotic (humidity, dew and rainfall) and biotic (fungi, bacteria and nematodes) factors in the center of origin and other major producing countries [5].

Digital image processing is the use of computer algorithms to do digital image processing. The image processing is an examination and manipulation of a digital image, in particular to improve the image processing quality [6]. The technique of digital image processing can be used in various fields, such as crop products and medical industry [7].

The researcher intended to apply image processing on classification and grading of Ethiopian coffee beans in this study. An application using image processing techniques for computer vision includes five basic steps such as image acquisition, preprocessing, segmentation, extraction and classification of features [3]. Artificial Neural Networks (ANN) is an attempt to mimic the neurons of the brain. The models used, however, have many simplifications and thus do not reflect the true behavior of the brain. Development of ANN had its first peak in the 1940s, and development has since gone up and down. The weighted sum of signals from other connected nodes is computed and a model of a neuron is created. The nodes are connected in two main patterns. No loops occur in feed-forward networks and loops occur in recurring networks. There is also a multi-layer feed-forward network with an initial input stage, hidden layers, and output layer [8].

The main objective of machine learning (ML) is to develop systems that can change their behavior autonomously based on experience. ML methods use training data to induce general models that can detect the presence or absence of new (test) data patterns. In the case of images, training data may take the form of a set of pixels, regions or images, which can be labeled or not. Patterns can correspond to low - level attributes, such as a label for a group of pixels in a segmentation task, or high - level concepts [9].

Deep learning is a technology inspired by the functioning of human brain. In deep getting to know, networks of synthetic neurons analyze large dataset to automatically discover underlying styles, without human intervention, deep learning identify patterns in unstructured statistics such as, Images, sound, video and text. Convolutional neural networks (CNN) become very famous for photo classification in deep getting to know; CNN's perform better than human subjects on most of the image type datasets [10]

## **1.2. Motivation of Research**

Why we were inspired to study in this area is Ethiopia's economy is based on agriculture. The agricultural sector suffers from poor cultivation practices and frequent drought. Although recent joint efforts by the Government of Ethiopia and donors have strengthened Ethiopia's agricultural resilience. In addition to this, Coffee is a major export crop in the Ethiopia as well as it produced excellent foreign currency for the country. So, coffee product has a great influence for the development of the Ethiopian economy. Nowadays, the main headache in the business world is competitive force. Gaining market share as well as market penetration is directly related with quality of the product in which the company will produce. Classification and sorting of coffee in coffee quality inspection and certification center organization which is found in Addis Ababa is manual. According to international coffee organization report, the county of origin for crop, Ethiopia produces premium quality coffee. It is the leading producer in Africa, and the 5th in the world, following Brazil, Vietnam, Colombia and Indonesia. If we consider Arabica alone, Ethiopia is the 3rd largest producer after Brazil and Colombia. Ethiopia also has the largest highland area suitable for Arabica production and, hence has the potential to be a leading producer in both quality and quantity. Therefore, the coffee quality inspection and certification center should keep the quality in order to be competitive and to get more foreign currency.



### 1.3. Problem Statement

Ethiopia is the first country from Africa by exporting high quality coffee Arabica to the foreign country. To keep this rank and to become the leading country by exporting coffee Arabica in the world it is necessary to keep the quality coffee. Ethiopian coffee quality inspection and certification center or Coffee board is responsible for classifying and grading coffee bean before exporting. To be competitive in the market the organization should care about the quality of the product specifically.

The classification and grading of coffee in Ethiopian coffee quality inspection and certification center or Coffee board is manual. This leads to so many problems like pruning of error, inefficient, require a lot of labor and is not cost effective. Moreover, lack of well-educated personnel, use of traditional grading technique and lack of advanced measuring equipment.

This method employs visual and manual methods of inspection of the major entities used, including appearance, texture, shape, size and color of coffee beans, exposing the quality assessment to inconsistent results and subjectivism. In addition, the tedious and time-taking human operator inspection activity for grading this high value product is very expensive, less efficient and less effective generating less descriptive and biased data information for quality control and other innovative improvement activities. Lack of a specialized specific field of study and qualification at country level for sorting, grading and classification of this item represents an important drawback which affects the reliability, efficiency and consistency of the practice. The cost incurred to fulfill this gap at various scales of trainings to generate capable experts is also significant. This will be a serious problem when observed from the perspective of extending and decentralizing the classification and grading activities to many other regions of the country. For example, they measure the size and shape of bean using the apparatus paul kaack & co Hamburg 20. They add 300g coffee bean into this apparatus and shake the apparatus until it stops screening the beans which are the screen size under 1/14 inch. They told may this takes more than five minutes for 300g coffee bean, therefore think how much time it consumes and it is very tedious. Previously, different findings [11] [12] were conducted in this area. However, those study didn't apply different deep learning technique mainly convolutional neural network. Now day's CNN performing promising accuracy in image classification and helps the researcher to use the model which are trained for different situation through using their weights. According to [11], the

researcher only applied manual binary thresholding to get region of interest. This type of segmentation is simply and ease to segment image, but it is not that much enough method for segmenting image which are overlapped together. Which mean that, it is very difficult to know the intensity value of overlapped area of bean. The classification performance of machine learning algorithm is not promising because, lack of many deep hidden layer [12]. Human intervention during feature extraction is also one of the challenging task in machine learning algorithm. The extent in which machine learning algorithm classification accuracy in previous studies are 77.% [11] and 82.8% [12]. As a result, replacement of the human operator systems with the consistent, nondestructive, superior speed, precise and cost effective automated system of coffee quality classification and determination is necessary for such commercial products that generate a huge amount of income to the country. The automated computer vision classification and grading system enables eliminating possible and potential human error and bias in the process. The application of this mainly three objective inspection techniques such as morphological, texture and color has expanded into many and diverse industries for food and agriculture to assist the inspection and grading of various fruits and vegetables in a non-destructive method. Its speed and accuracy satisfy the ever-increasing production and quality requirements, thereby promoting the development and expansion of totally automated processes. In addition, it has been successfully applied in the analysis of grain characteristics and evaluation of food crops [11]. Therefore, due to the above mentioned problems we were tried to employ relevant and applicable image analysis, processing and classification techniques and models, with the aim of introducing and achieving the mentioned benefits of technological approaches for coffee bean raw quality value classification and determination. The new technologies of image analysis and machine vision have not been fully explored in the development of automated machine in agricultural and food industries. This calls for launching explore researches for applying, evaluating and developing these emerging technologies to assist the betterment of quality control and productivity issues of the sectors.

In general, manual sorting and classification, which is based on traditional visual quality inspection performed by human operators, are tedious, time-consuming, slow and inconsistent [3].

At the end of this study, this research will answer the following research questions:

- ⇒ Which method is best from convolutional neural network algorithm for classification of Ethiopian coffee bean?

- ⇒ Which segmentation algorithm is best for classification of Ethiopian coffee bean?
- ⇒ To what extent the deep learning algorithm classify the coffee bean images?

## **1.4. Objectives of the study**

### **1.4.1. General objective**

The general object of the study is to design and develop a system to classify coffee bean using deep learning.

### **1.4.2. Specific Objectives**

In order to develop automatic system for classification of coffee bean first the researcher must achieve the following objectives specifically.

- ✓ Review different related literature to understand the state-of-the-arts in deep learning and bean classification;
- ✓ Collect required image data from Ethiopian coffee quality inspection and certification center data warehouse.
- ✓ Apply different image preprocessing techniques on collected image to get better result.
- ✓ Apply image segmentation to get region of interest easily.
- ✓ Identify which segmentation technique is the best.
- ✓ To design the architecture of the Coffee bean classification using deep learning;
- ✓ To build the convolutional neural network model which predict the class of coffee bean image.
- ✓ Select the best model which perform better accuracy.
- ✓ Identify to what extent the classification algorithm classified the coffee bean by using performance evaluation metrics.
- ✓ To develop a prototype for the Ethiopian Coffee bean classification using deep learning technique;

## **1.5. Significance of the Study**

Developed automated model for classification of coffee bean have great advantage for the Ethiopian coffee quality inspection and certification center or coffee board.

Application of the result on the intended organization provides the following benefits; -

- ✓ It would minimize the processing time and labor cost. This would also improve quality based export of coffee bean.
- ✓ It would enable Ethiopian coffee quality inspection and certification center to have the same standard across all products and quality control were been easy, because it gives a platform to conduct classifying at one specific place, centralization.
- ✓ Reduce capabilities of decision-making errors that were comes from human inspector physical condition such as fatigue and eyesight, mental state caused by biases and work pressure, and working conditions such as improper lighting, climate, etc.
- ✓ It would help different merchants to get fair quality control from the organization without any biases.
- ✓ It would benefit researchers who are motivated in achieving the goal of developing efficient digital image processing techniques for different agricultural products and other areas.

## **1.6. Scope and Limitation of the study**

The purpose of this research work is to apply image analysis techniques and approaches on Ethiopian coffee variety identification and quality analysis (sorting) based on their growing region. Our sample data comes from Ethiopian coffee produced in the 2019/20 production year. Coffee quality inspection in Ethiopia is divided into two parts. Such as, green analysis (visual test) and liquor analysis (cup test). To identify and classify coffee, the green analysis relies on the human sense of sight (eye). This method examines the physical characteristics of coffee, such as shape, size, color, uniformity or irregularity, and defect count. [12].

The cup test is the other. To recognize and classify coffee, it uses the human sense of taste (tongue). It looks into coffee's chemical properties. Acidity, body, and flavor are the parameters of the cup test. [12]. Based on this, this research uses only green analysis to classify coffee beans into Sidamo, Jimma, Nekempti, Kaffa, Gujji and Yirgacheffe as class labels and grading in not incorporated in this study because grading done by using cup test as the employee in the organization stated and other coffee varieties are not included because, lack of well-prepared dataset.

## **1.7. Organization of the Thesis**

This thesis work is organized in five different chapters, but interrelated. The remaining part of the thesis is organized as follows: - The first chapter is started by discussing the background of the

study which incorporate the domain area mainly agriculture specific to coffee beans. This is followed by the motivation of the study, statement of the problem, general as well as specific objectives, scope and limitation, advantage of this research for organization, employees and farmers is discussed and methodology of study is stated. In Chapter Two, different literature is reviewed to understand the area and techniques used to conduct this study. The first section is deal with background information of coffee like the origin, methods of quality control, coffee processing types and physical as well as chemical characteristics of coffee. This is followed by discussing background of image, artificial neural network, machine learning and deep learning. Next to this, the basic concept and theory of CNN and other relevant image classification tasks are reviewed and the other related works which are conducted in coffee bean is discussed. Chapter three attempted to discuss the research design which is about design science research methodology in detail including the framework and steps included in the design science. This is followed by the proposed system architecture which includes data acquisition, preprocessing of images, segmenting images to get region of interest, data augmentation and classification for both training and test phases. Finally, in this chapter we discussed about evaluation metrics in detail. Chapter four presents the experimentation and evaluation of developed model, experiment design, selecting appropriate hyper-parameter, coding the experiment based on the proposed CNN architecture, training the model and testing the trained model. At the end of this chapter, we developed web application to make the user of the organization make interact with developed model. Chapter five describes the conclusion part and recommendation for those who are motivated to conduct further study in this area.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.1. Overview**

This chapter primarily focus on understanding the theoretical background of the coffee in Ethiopia as well as in the world. In addition to this, coffee classification and grading are reviewed, two types of coffee quality inspection, different coffee varieties and their physical and chemical properties are discussed in detail. Introduction to image and image processing steps, deep learning techniques, different state-of-the-art and their architectures are discussed. Finally, different related works were reviewed and discussed related to coffee and other agricultural crop and CNN algorithm.

#### **2.2. Coffee in Ethiopia**

Coffee is a brewed drink prepared from roasted coffee beans, which are the seeds of berries of the Coffee plant. The major exportable and consumed commodity originates from the land of Ethiopia. The word "coffee" comes from the name of a region of Ethiopia where coffee was first discovered – ‘Kaffa’. The name ‘Kaffa’ is inherited from the hieroglyphic nouns ‘KA’ and ‘Afa’. ‘KA’ is the name of God, ‘AFA’ is the name of the earth and all plants that grow on earth. So the meaning of Koffee (Coffee) from its birth-place bells on as the land or plant of God. Botanically, coffee belongs to the family Rubiaceae in the genus Coffea [15] [16].

Although the genus Coffea includes four major subsections such as forest and semi-forest (10%), garden coffee (85%), and plantation coffee (5%) are the major conventional production systems in Ethiopia [17]. 66% of the world production mostly comes from Coffea Arabica and 34% from Coffea canophora Pierre ex Froehner (Robusta type), respectively [12]. Ethiopia is the home and the cradle of biodiversity of Arabica coffee seeds. Arabica exist in Ethiopia than anywhere else in the world, which has lead botanists and scientists to agree that Ethiopia is the center of origin, diversification and dissemination of the coffee plant.

##### **2.2.1. Classifications and Grading of Ethiopian Coffee**

Depend on its characteristics and growing region, Ethiopian coffee is classified under the following. Ethiopian coffee has unique flavors: Spicy of Sidamo coffee, Winy of Limu coffee, Fruity of Nekempti Coffee, Floral of Yirgacheffe Coffee, Mocha of Nekempti Coffee and many

more. The primary issues of coffee grading are country (region) of origin, physical characteristics and sensory standards (taste). There is no universal coffee grading system except the recommended standards. Each producing country has its own national grading system standard [11].

In Ethiopia, there are two main major components of the coffee quality inspection. They are green analysis (visual test) and tasting liquor (cup test). These two approaches are universally appropriate methods adapted to the quality management systems of the respective countries in both coffee producing and consuming countries [12].

Green analysis accounts for 40% of a coffee's overall grading, with the remaining 60% determined by cup test. The green analysis uses additional coffee identification and classification procedures as well as the human sense of sight (eye). This method checks physical aspects of coffee, such as shape, size, color, consistency or regularity, and the number of defects in the beans. The parameters moisture content and bean size are preliminarily assessed in coffee grading. The maximum moisture content allowed is 11.5 percent. The Ethiopian coffee bean has a screen size restriction of 14 units, where one unit equals 1/64 of an inch. If these two characteristics are not met, the coffees are considered of lower grade, and no further grading examination is performed. For the case of moisture content, it is recommended to reprocess the coffees to minimize the moisture content [11] [12].



*Figure 2. 1 Moisture Apparatus [16]*

To assess the size of each coffee bean, a sieve-like instrument is used to conduct a screen size examination as shown in figure 2.2. The analysis is carried out by placing 300g of coffee beans in the device and shaking them repeatedly. The number of coffee beans that pass through the holes is then weighted to determine the fraction of the coffee bean that falls inside the prescribed screen size. The size of a coffee bean screen is typically reported to be 14 to 20. The numbers represent

the size of the sieve's holes, which are 1/64 of an inch. For example, screen size 14 indicates that the hole's diameter is 14/64 of an inch. [11].



*Figure 2. 2 Screen Size Apparatus [12]*

After the beans have passed the preliminary tests, other parameters related to their processing type will be evaluated. The parameters shape, odor, and color are used to grade washed coffee. Color, odor, and shape parameters account for 15%, 10%, and 15%, respectively, of the total weight of grading by green analysis, which is 40%. Similarly, the parameters of unwashed coffee are defect count and odor, which contribute 30% and 10% respectively to the total weight of grading by green analysis, which is 40% [12].

The cup test is the other component of grading. To identify and classify coffee, it is based on the human sense of taste (tongue). It focuses on the chemical properties of coffee. The cup test parameters are acidity, body, and flavor. Acidity is a primary coffee taste sensation produced when acids in coffee combine with sugars to increase the overall sweetness of the coffee. The body of coffee refers to its texture and sensation in the mouth; for instance, coffee can feel light or heavy. Flavor refers to the aroma or smell perception of the elements present in roasted coffee. Each of these parameters accounts for 20% of the total weight of the grading by cup test, which is 60% [12].

According to [18], Ethiopian coffee is graded using a 40 percent green analysis and a 60 percent cup test. Based on the overall test value of each parameter for either unwashed or washed coffee, the final grade is determined using the Ethiopian coffee grading system's national standard. For



example, a grade one coffee has a cumulative value of 81-100 percent from both the visual and cup tests. The grades are set specific to the growing region.

According to [19], in reality, grading and classification systems are usually based on some, or all, of the following criteria. In reality, most grading and classification systems are based on some or all of the following criteria. This means that most systems are often very detailed and diverse, leaving room for misunderstanding and misinterpretation regarding the 'transferability-' of certain descriptions and terminologies between producing countries:

- ✓ Altitude
- ✓ Region
- ✓ Botanical variety
- ✓ Preparation method (wet vs dry)
- ✓ Bean (screen) size
- ✓ Bean shape and color
- ✓ Number of defect counts
- ✓ Density density
- ✓ Cup quality

## **2.2.2. Coffee Varieties in Ethiopia**

### **2.2.2.1. Sidama**

The Sidama region is in southern Ethiopia. Sidama's name is sometimes pronounced "Sidamo," and the two names are commonly used interchangeably. Sidama's coffee growing regions are located in the well-known Great Rift Valley, which runs through Ethiopia and Kenya. The countryside is generally lush and green. This region of Ethiopia, while largely rural, is densely populated. Several large freshwater lakes dot the landscape, flowing in a long chain across the valley. Sidama's coffee has an unusually diverse range of coffee flavors. Many different grades of washed and unwashed coffee are produced, and there can be significant differences between cities [20].

A kaleidoscope of different flavors is produced by varying soil types, micro climates, and particularly the countless heirloom coffee tree varieties. Another notable feature of Sidama coffees is that even washed coffees retain a prominent fruity flavor while having significantly more clarity

and brightness than their unwashed counterparts. Although some coffees are sold through the Soddo hub, Hawassa is the main arrival point for Sidama [21].

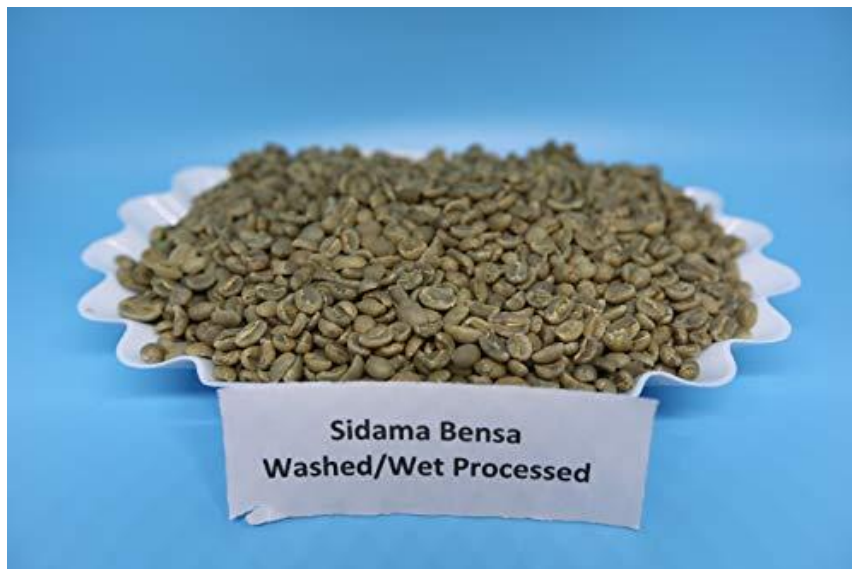


Figure 2. 3 Sidamo coffee bean image [4]

#### **2.2.2.2. Yirgacheffe**

According to [20], Yirgacheffe is a small microregion within the much larger region of Sidama. Yirgacheffe coffees, on the other hand, are so distinct and well-known that they have their own category. The quality of Yirgacheffe coffee, though much, much smaller than the other regions, has enabled it to become as well known or even better known as the big popular coffee producing regions of Harrar and Sidama proper. Yirgacheffe is a small town of about 20,000 people, geographically located somewhat centrally in relation to the other coffee growing areas of Sidama, between the large towns of Dilla and Agere Maryam.

Many characteristics are shared by the best Yirgacheffe coffees and the best Sidama coffees. Fruit flavors, a bright acidity, and a silky mouthfeel are some of its distinguishing characteristics. Yirgacheffe grows both washed and unwashed coffee beans. While it is best known for its washed coffees, it has recently begun to export some highly sought-after top-quality unwashed coffees as well.

Yirgacheffe's premium washed coffees are known for their vibrant citrus acidity, mostly lemony in flavor, and excellent sweetness. Other distinguishing features of the coffee include a light,

herbaceous consistency that complements the fruit flavors well, and the nearby large town of Dilla serves as the distribution center for all Yirgacheffe coffees. [21].



*Figure 2. 4 Yirgacheffe Coffee Bean Image [19]*

### **2.2.2.3. Southwest Regions**

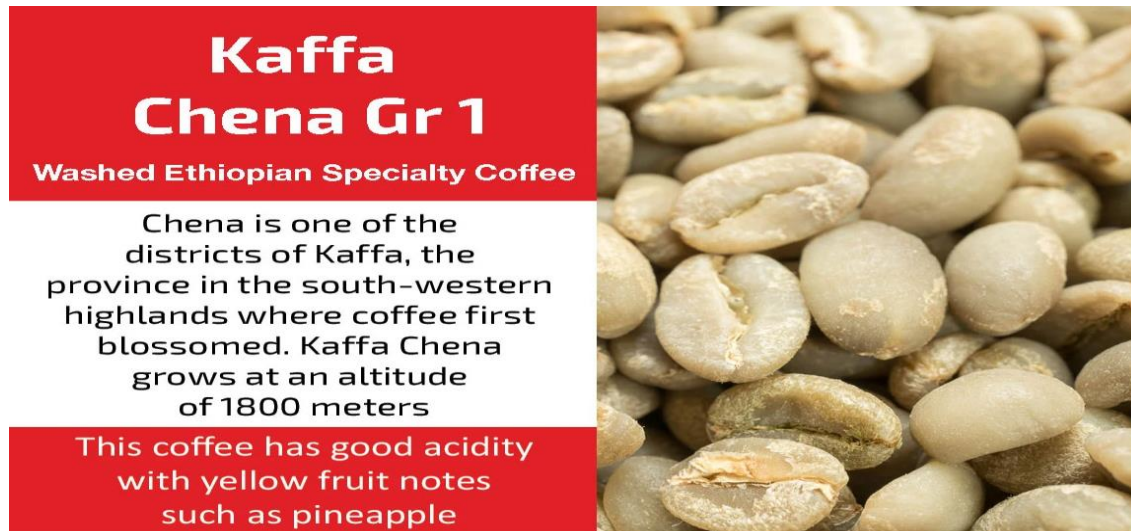
Limu Coffee is grown in the southwest of Ethiopia at elevations ranging from 3,600 to 6,200 feet. Limu coffee (all washed) has a milder acidity than Sidama and Yirgacheffe; a healthy and clean cup is typically characterized by the taste. Traditionally, washed Limu coffees were sold under that name; unwashed Limu coffees were typically offered in the Jimma category. Jimma, also spelled “Djimmah,” is Ethiopia's largest basket of unwashed coffees, encompassing all unwashed coffee produced in Ethiopia's southwestern region. The region is home to a plethora of indigenous varieties, the quality of which can vary greatly. [21].



*Figure 2. 5 Jimma Coffee Bean Image [19]*

More than a hundred Ethiopian investors have built estates and farms in the Bonga district, a town in the Kaffa zone, to grow high-quality Arabica coffee. It has ideal Agroecological conditions for specialty coffee production. The soil's altitude ranges between 1600 and 1900 meters, it's red in

color, and the temperatures are ideal for coffee production. The area is known for its high precipitation levels and is thus regarded as one of Ethiopia's rainiest regions. The majority of farms, estates, and cooperatives supply both washed and natural sun-dried coffee to international markets. Because of its distinct flavor and bean appearance, many cuppers associate Kaffa washed coffee with coffee from the Borena region, while others compare its flavor to that of neighboring Limu coffee [21].



*Figure 2. 6 Kaffa Coffee Bean Image [20]*

#### **2.2.2.4. Nekempti**

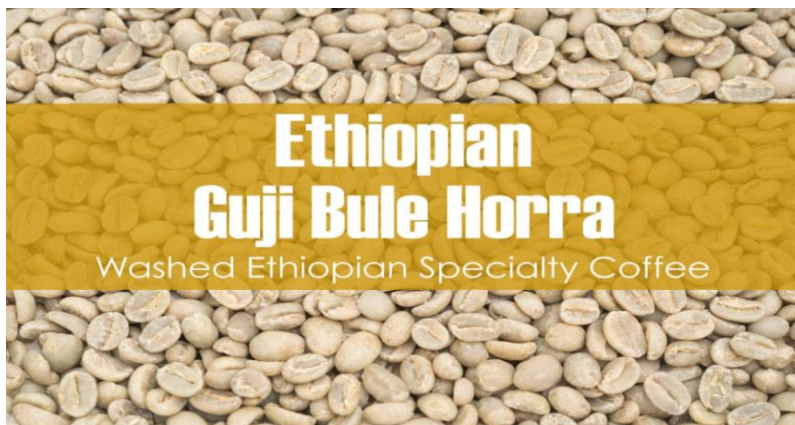
Nekempti is an area located within the state of Wollega, also known as Nekempti. Usually, this coffee will be sold as "Nekempti," a coffee trade name for Western Ethiopian coffees traded through the town of Nekempti, although the coffee actually comes from East Wollega, also known as "Misraq Wollega," which is the Gimbi woreda. Nekempti is a sun-dried natural bean originally grown in western Ethiopia. The coffee is distinguished by its large bean size, and the flavor can have a strong perfume-like aftertaste. Wollega's coffee processing methods have traditionally been sun-dried natural [21].



*Figure 2. 7 Nekempti Coffee Bean Image [15]*

#### **2.2.2.5. Gujji**

Gujji is another fantastic region. Coffee from Gujji, located in the southern Sidamo region, is sought after by some of the world's best roasters. Sweet floral notes, such as jasmine with melon and peach, and a tea-like body can be found in the cup.



*Figure 2. 8 Gujji Coffee Bean Image [15]*

Table 2. 1 Coffee Bean Characteristics

No	Coffee Type	Preparation	Shape & Make of Bean	Color	Roast	Liquor Quality	Bean Size
1	Sidamo	Washed	Oval to Round	Greenish to Grayish	White Center cut	Spicy & sweet	Medium to Bold
2	Gujji	Washed	Elongated	Greenish to Grayish	Bright roast	Exotic Flavour	Medium to Bold
3	Jimma	Unwashed	Pointed bean	Greenish	Dullish	Winy Flavour	Medium to Bold
4	Kaffa	Washed	Roundish	Grayish	Bright	Sweaty	Medium
5	Nekempti	Unwashed	Long & pointed ends	Greenish, Brownish slightly coated	Bright, Brownish	Fruity Flavour	Bold
6	Yirgacheffe	Washed	Oval to slightly elongated	Bluish	White Center cut	Flora & spicy	Bold

Source: Ethiopian coffee quality inspection and auction center

### 2.2.3. Ethiopian Coffee Processing

According to [18], Ethiopia produces a lot of coffee, and there are two types: sun-dried natural and fully washed. Nekempti and Jimma coffees are mostly sun-dried naturals, whereas Sidamo and Yirgacheffe coffees are mostly washed. Ethiopia, unlike some other countries, has both washed and sundried processed natural coffees. To reveal the green coffee bean, the skin, pulp, and parchment from the outer layers of the coffee cherry are removed. Coffee pulp, also known as mucilage, is extremely sticky and high in sugars. To remove the mucilage from the beans, special processes are required.

Red cherry is picked and freshly sorted before pulping. Over-ripe and under-ripe beans are handpicked and separated before processing. Fresh red cherries are supplied to the washing stations. Coffees are pulped and allowed to ferment naturally. The fermented coffee is washed in clean running water, soaked in clean water, and dried to retain approximately 11.5 percent moisture. Dried parchment coffee is stored in a field warehouse until it is transported to Addis Abeba for further processing. The husks are removed from the parchment coffee in a dry

processing warehouse, and the clean beans are packaged in label bags (60 Kg bags/132 lbs) for export [4].

### **2.2.3.1. Washed Processing**

It is the common processing for premium coffees. After the red cherries are picked and the coffee is further sorted by immersion in water. Less dense cherries will float and others will sink. Eco-pulpers remove the skin of red cherries to produce parchment coffee; however, the parchment coffee retains a significant amount of mucilage. To remove the mucilage, the parchment coffee will be fermented for 2 to 3 days, depending on the temperature and humidity of the area. After the mucilage is removed, the coffee is transferred to a soaking tank [18].

The coffee soaks in the tank for about 12 hours before being transferred to the raised bed, where it dries to the proper moisture level for about two weeks. After the coffee has been dried to the appropriate moisture level, it will be handpicked to remove any exposed or damaged coffee. The dried parchment will be transported to the cooperative warehouse where the coffee will be stored before being loaded into the Addis Ababa warehouse for dry processing [5].

The outer skin of the coffee cherry is removed immediately after harvesting, usually on the same day the cherries were picked, in the washed coffee processing. This is done using machines which "pick" or scrape away just the very outer layer of the cherry, leaving behind the parchment coffee covered in sticky mucilage. The mucilage-coated beans are then fermented with water in large tanks made of cement. The process of fermentation breaks down the sugars in the mucilage and frees it from the parchment. Fermentation usually takes around 24 hours [11], though shorter or longer fermentation times are possible, depending on the local climate, altitude, and other factors.

Once fermentation is complete, the coffee is removed from the fermentation tank and physically pushed down lengthy pipes using flowing water. Any leftover mucilage is freed and separated from the parchment coffee by agitation. The coffee enters another tank at the end of the channels, where it is rinsed with new water. The result is wet coffee in parchment, free of the sticky mucilage [18].

From the final washing tank, the wet parchment coffee [22] is taken to dry in the sun on raised beds. This process of drying happens quickly, because there is no skin or mucilage between the sun and the parchment. After one or two days in the sun, the coffee is removed from the beds and

stored in sacks in a warehouse. When it is to be exported, the coffee is usually taken to a larger central mill where the parchment is removed, and the coffee is sorted and bagged for export.

Washed coffee has a clarity of flavor and scent that natural coffees frequently lack. Many cuppers believe that with washed coffees, the influence of soil and varieties is more immediately evident. The acidity is more noticeable, and the cup is generally cleaner. The cleanest, highest-quality, high-altitude washed coffees can have an intensely refreshing flavor; however, the washed process requires a large amount of water and more infrastructure. The washing procedure is simply not practical in many places [12].

### **2.2.3.2. Natural (Unwashed) Coffee Processing**

First, the arrival cherry is hand-sorted to separate the less dense cherries. The good cherry is then taken to a raised bed to dry in the sun. This could take up to 21 days. When the coffee has been dried with cherry, it is milled to remove the husks and stored in a warehouse before being transported to the final processing warehouse. It is shipped with a 60 Kg bag inside [18] [11].

According to [22], The cherries' skin and sticky juices dry out in the sun. Depending on the temperature and intensity of the sun, this process can take several days to a few weeks. The coffee is covered up at night or when it rains. The cherries shrink in size during the drying process and eventually become hard and completely dry. Following the completion of the process, sacks of dried cherries are transported to a hulling station for the removal of the outer cherry.

Care must be taken to ensure that the cherries are dried uniformly and that no contaminating substances come into contact with the cherries, such as direct contact with soil. Inadequate attention to this information may result in muddy, dirty, or fermented flavors in the cup. Natural processing has the significant advantage of not requiring any water or elaborate machinery or facilities. As a result, more naturally processed coffees can be found in drier areas, as well as poorer or more remote areas [4].

## **2.3. Image Representation**

Image defined in the “real world” is considered to be a function of two real variables, such as  $a(x,y)$  with an image as the amplitude (e.g. brightness) of the image at the *real* coordinate position  $(x,y)$  [23]. The additional concept reflects the fact that images often contain object collections, each of



which can be the basis for a region. In a sophisticated image processing system, specific image processing operations should be applicable to selected regions.

A digital image  $a[m,n]$  described in a 2D discrete space is derived from an analog image  $a(x,y)$  in a 2D continuous space through a sampling process that is frequently referred to as digitization.

The 2D continuous image  $a(x,y)$  is divided into  $N$  rows and  $M$  columns. The intersection of a row and a column is termed a *pixel*. The value assigned to the integer coordinates  $[m,n]$  with  $\{m=0,1,2,\dots,M-1\}$  and  $\{n=0,1,2,\dots,N-1\}$  is  $a[m,n]$ . In fact, in most cases  $a(x,y)$  – which we might consider to be the physical signal that impinges on the face of a 2D sensor – is actually a function of many variables including depth ( $z$ ), color ( $\lambda$ ), and time ( $t$ ).

An image captured by a sensor or a camera is expressed as a continuous function  $f(x, y)$  defined on continuous variables  $x$  and  $y$ . And the function values refer to the amplitude at that point  $(x, y)$  [3]. To convert such image to digital form, we have to sample the continuous image in both coordinates and the amplitude. Digitizing the coordinate values  $x$  and  $y$  is sampling that provide the set of pixels. To convert such an image to digital form, we must sample it in both coordinates and amplitude. Sampling is the process of digitizing the coordinate values  $x$  and  $y$  in order to provide the set of pixels. Quantization is the process of digitizing the amplitude, which is the gray level. Quantization is the process of converting a continuous graylevel (amplitude) into a discrete quantity [12].

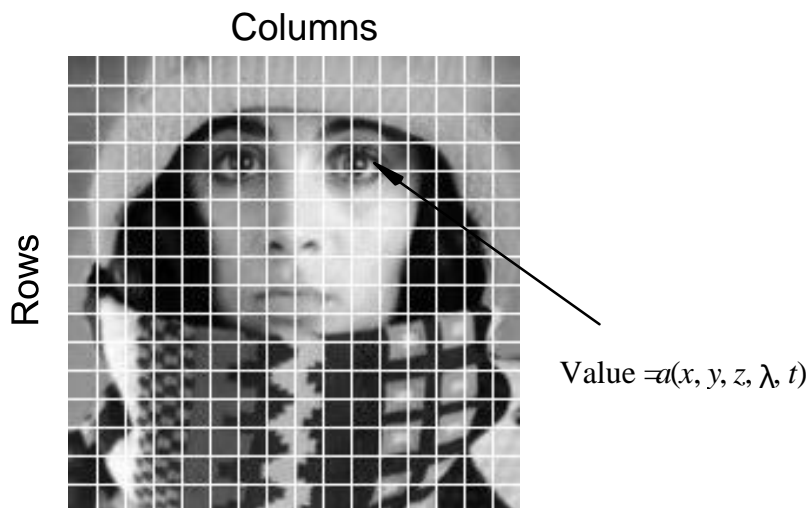


figure 2. 9 Digitization of a continuous image. [21]

Digital Image Processing (DIP) deals with manipulation of digital images through a digital computer. DIP focuses on developing a computer system that is able to perform processing on an

image. This digital image processing has been employed in number of areas such as image classification, pattern recognition, remote sensing, image-sharpening, colour and video processing and medical [7]. A digital image can be considered as a discrete representation of data possessing both spatial (layout) and intensity (colour) information [24].

An image contains one or more colour channels that define the intensity or colour at a particular pixel location  $(m, n)$ . In the simplest case, each pixel location only contains a single numerical value representing the signal level at that point in the image. The conversion from this set of numbers to an actual (displayed) image is achieved through a colour map. A colour map assigns a specific shade of colour to each numerical level in the image to give a visual representation of the data. The most common colour map is the greyscale, which assigns all shades of grey from black (zero) to white (maximum) according to the signal level.

The most commonly used color channels in the image are red, green and blue and gray-scale color channels or colour space. Colour space is considered as a mathematical entity. An image is really only a spatially organized set of numbers with each pixel location. Gray-scale (intensity) or binary images are 2-D arrays that assign one numerical value to each pixel which is representative of the intensity at that point. They use a single-channel colour space that is either limited to a 2-bit (binary) or intensity (grey-scale) colour space. By contrast, red, green and blue or true-colour images are 3-D arrays that assign three numerical values to each pixel, each value corresponding to the red, green and blue components respectively [24].

The RGB color space is the most commonly used for digital image representation because it corresponds to the three primary colors that are mixed for display on a monitor or similar device. A common misconception is that items perceived as blue, for example, will only appear in the blue channel, and so on. While items perceived as blue will undoubtedly appear brightest in the blue channel (i.e. contain more blue light than the other colors), they will also contain milder red and green components.

A simple transform can be used to convert an RGB color space to a greyscale image. Many image analysis algorithms begin with grey-scale conversion because it essentially simplifies (i.e. reduces) the amount of information in the image. Although a greyscale image contains less information than a color image, the majority of important, feature-related information, such as edges, regions, blobs, junctions, and so on, is preserved. [24].

## 2.4. Image Processing

Digital image processing refers to analyzing digital images by means of a digital computer to do some operation to improve the images and extract important image information or representations, it's the extraction of meaningful information from image which contains element or pixels [25].

Image processing and image analysis are at the heart of computer vision, with a plethora of algorithms and methods available to achieve the necessary classification and grading. According to this viewpoint, digital image processing is divided into two major tasks. These include improvement of pictorial information for human interpretation; and the other task is processing of image data for storage, transmission and representation for autonomous machine perception [3].

A computer-vision application using image processing techniques involves five basic steps such as image acquisition, preprocessing, segmentation, feature extraction, classification and interpretation [3] [26].

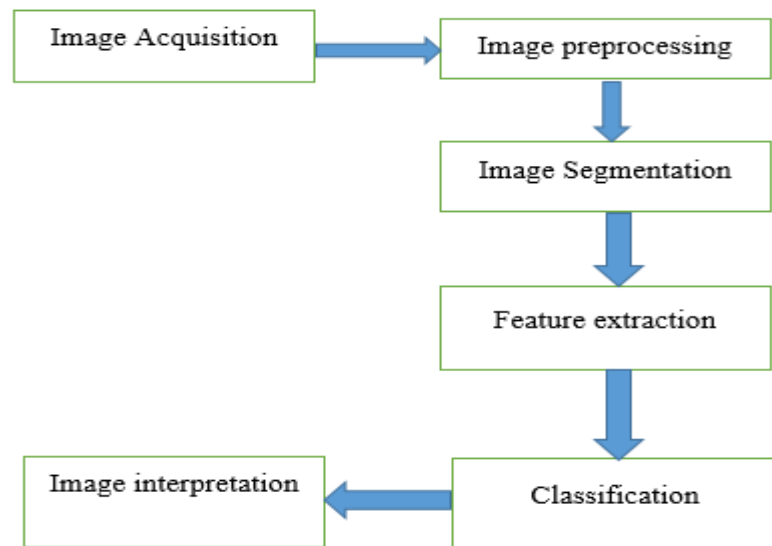


Figure 2. 10 Fundamentals steps of digital image processing

### 2.4.1. Image Acquisition

The concept of image acquisition is to gather or retrieve images from various sources, most of the time from a reliable repository or captured by using a camera. It is the first and most important step in digital image processing because without an image, it is impossible to say image processing and perform image analysis to enhance or extract important information. Image acquisition is the most important factor in the success of image analysis applications [25] [27]. Image acquisition is

a process of retrieving a digital image from a physical source which is captured using sensors or cameras since the quality of images will be affected through different factors [3].

### **2.4.2. Image Pre-processing**

This refers to the initial processing of the raw image. The images captured or taken are transferred onto a computer and are converted to digital images. Digital images though displayed on the screen as pictures, they are digits which are readable by the computer and are converted to tiny dots or pixel (picture elements) representing the real objects [12]. The majority of image-processing techniques involve the removal of low-frequency background noise, normalization of the intensity of individual particle images, removal of reflections, and masking portions of images.

Image preprocessing is the process of improving data images before they are processed computationally [26]. In recent years, the widespread use of digital cameras has eliminated the need for an additional component to convert images captured by photographic charge coupled device (CCD) cameras or other sensors to a format readable by computer processors. Images taken with a digital camera retain their features with little noise due to their variable resolution. In this study, images were captured using a digital camera [12], and image processing is also possible. Some of image pre-processing tasks are:

#### **2.4.2.1. Image Re-sampling**

It is the process of changing the coordinate system of a sampled image (changing the pixel dimension of an image). The display size of an image is also affected by re-sampling [26]. Image re-sampling includes image cropping, which is required to center crop the images in order to delete the irrelevant regions and reduce the computational time for the deep learning network [28].

#### **2.4.2.2. Image Enhancement**

Image enhancement is one of the most straightforward and appealing aspects of digital image processing. The goal of enhancement techniques is to bring out detail that is obscured in an image, or simply to highlight certain features of interest in an image. When we increase the contrast of an image to make it look better, this is an example of enhancement. It is critical to remember that image enhancement is a highly subjective area of image processing [26].

### **2.4.2.3. Image Restoration**

According to [26], image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration technique tends to be based on mathematical or probabilistic models of image degradation.

### **2.4.2.4. Noise Removal**

The term "noise" refers to the fact that the pixels in an image display varying intensities rather than the true pixel values obtained from the image. The process of removing or reducing noise from an image is referred to as a noise removal algorithm. By smoothing the entire image and leaving areas near contrast boundaries, noise removal algorithms reduce or remove the visibility of noise. The following are the primary sources of noise in digital images: a) During image acquisition, the imaging sensor may be affected by environmental conditions. b) Inadequate light levels and sensor temperature may cause noise in the image. c) Interference in the transmission channel may also cause the image to be corrupted. d) If there are dust particles on the scanner screen, they can cause noise in the image. [29]. This noise generally arises during the process of image acquisition. The suppression of noise is a challenging task in digital image processing.

### **Types of Noise**

Noise is an unwanted interference that degrades image quality. It causes undesirable effects on the image such as blurring, artifacts, edge distortion, and so on. Before applying any filter, it is critical to understand the type of noise in the image [30]. There are various types of noise depending upon the source and cause that can affect the image.

### **Amplifier Noise (Gaussian noise)**

The standard model of amplifier noise is additive, Gaussian, dependent at each pixel and dependent of the signal intensity, caused primarily by Johnson–Nyquist noise (thermal noise), including that which comes from the reset noise of capacitors ("kTC noise"). It is a simplified version of white noise caused by random fluctuations in the signal. Amplifier noise accounts for a significant portion of image sensor noise, that is, the constant noise level in dark areas of the image. Each pixel in the image is changed from its original value by a (usually) small amount in Gaussian noise

[30]. A normal distribution of noise is shown by a histogram, which is a plot of the amount of distortion of a pixel value against the frequency with which it occurs. While other distributions are possible, due to the central limit theorem, which states that the sum of different noises tends to approach a Gaussian distribution, the Gaussian (normal) distribution is usually a good model. This type of noise directly attacks on grey scale values of an image and so it is described as a PDF function:

$$P(x) = \frac{1}{\sqrt{2\pi i}} e^{-\frac{(x-m)^2}{2\epsilon^2}} \dots \dots \dots \text{Equation 2. 1}$$

where,  $x$  is intensity,  $m$  is mean value, and  $\epsilon^2$  is the variance of  $x$ .

### Salt and Pepper Noise

Salt and pepper noise can also be named as impulse valued noise or data drop noise as it causes drop in original data values. Consider a 3x3 matrix of an image and assume that the value of upper white colored pixel value (255 intensity) is corrupted by salt and pepper noise. So, this noise inserts dead pixel in the image and that dead pixel can either be dark or light as shown in figure 2.11 [29].



Figure 2. 11 White Color Pixel Value Corrupted with Salt & Pepper Noise [27]

### Denoising Techniques

We use the filtering process to enhance and denoise the image. There are various types of filters, such as the Gaussian filter, median filter, morphological filter, Wiener filter, and so on. Gaussian noise can be removed using the Gaussian denoising algorithm [31]. Salt-and-Pepper can be removed by the median filter, combined median and mean filters algorithm.

**Gaussian Filter:** Gaussian denoising is used to smooth images more effectively. It is the first step in detecting noise removers, but it is ineffective at removing salt-and-pepper noise. It is based on the Gaussian distribution [32].

**Median Filter:** We use a median filter instead of a Gaussian filter to improve edge estimation. It is a nonlinear filter that is used to remove salt-and-pepper noise [33].

#### 2.4.2.5. Categorical encoding

The dataset might contain text or categorical values (basically non-numerical values). For example, color feature having values like red, orange, blue, white etc. Most of the algorithms work better with numerical inputs. Therefore, the main challenge faced by an analyst is to convert text/categorical data into numerical data and still make an algorithm/model to make sense out of it. Neural networks, which is a base of deep-learning, expects input values to be numerical.

There are two main ways to convert categorical values into numerical values. One-Hot-Encoding and Label-Encoder. Both of these encoders are part of SciKit-learn library (one of the most widely used Python library) and are used to convert text or categorical data into numerical data which the model expects and perform better with [34].

### Label Encoding

Label encoding is a transformation of string value to numerical value [28]. It is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering. In this encoding, each category is assigned a value from 1 through N (here N is the number of categories for the feature [35]).

*Table 2. 2 Class index*

Image ID	Classes	Numerical Value
Img1.jpg	Gujji	0
Img2.jpg	Jimma	1
Img3.jpg	Kaffa	2
Img4.jpg	Nekempti	3
Img5.jpg	Sidamo	4
Img6.jpg	Yirgacheffe	5

## One-Hot Encoding

Once we have label encoding, we might confuse our model into thinking that a column has data with some sort of order ( $0 < 1 < 2$ ) or hierarchy when we clearly don't have it. To avoid this, this column is 'One - Hot Encoding '. What one hot encoding does is take a column with categorical data, which has been encoded as a label, and then split the column into multiple value. The numbers are replaced by 1s and 0s, depending on which column has which value [8].

Table 2. 3 One-Hot encoding

Image ID	Gujji	Jimma	Kaffa	Nekempti	Sidamo	Yirgacheffe
Img1.jpg	1	0	0	0	0	0
Img2.jpg	0	1	0	0	0	0
Img3.jpg	0	0	1	0	0	0
Img4.jpg	0	0	0	1	0	0
Img5.jpg	0	0	0	0	1	0
Img6.jpg	0	0	0	0	0	1

### 2.4.3. Image segmentation

In image processing, image segmentation can be defined as a "process of partitioning a digital image into multiple segments" (sets of pixels, also referred to as super pixels). The goal of image segmentation is to simplify and or change the representation of an image, which is more meaningful and easier to analyze. Image segmentation methods are categorized on the basis of two properties of discontinuity and similarity. Methods based on discontinuities are called boundary-based methods, and methods based on similarity are called region-based methods. The output of the segmentation is either a limitation of the object from the background or the region itself. In the color image segmentation, different color spaces such as RGB, HSI and CIELab are used [14], with the image segmentation [27].

Image segmentation technique is used to partition an image into meaningful parts having similar features and properties. The main aim of segmentation is simplification i.e. representing an image into meaningful and easily analyzable way. Image segmentation is necessary first step in image analysis. The goal of image segmentation is to divide an image into several parts/segments having similar features or attributes. The basic applications of image segmentation are: Content-based image retrieval, Medical imaging, Object detection and Recognition Tasks, Automatic traffic control systems and Video surveillance, etc. The image segmentation can be classified into two



basic types: Local segmentation (concerned with specific part or region of image) and Global segmentation (concerned with segmenting the whole image, consisting of large number of pixels) [36].

## CLASSIFICATION OF IMAGE SEGMENTATION TECHNIQUES

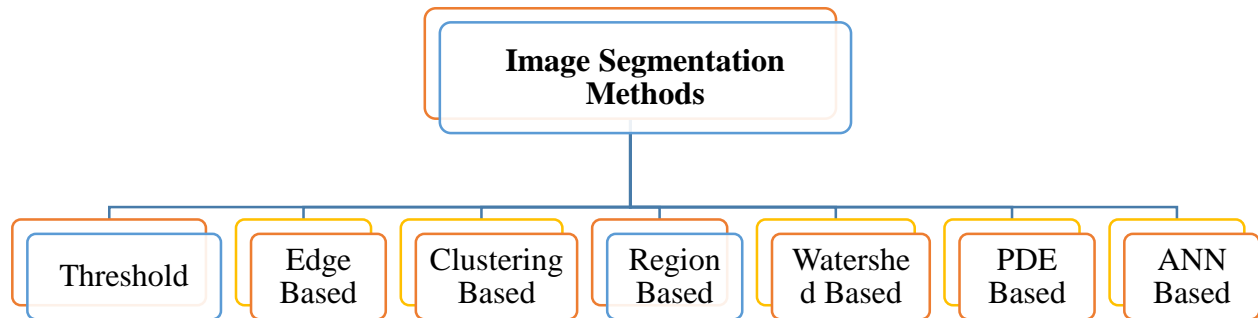


Figure 2. 12 classification of image segmentation techniques

### 2.4.3.1. Watershed Image Segmentation Technique

Watershed Transformation in mathematical morphology is a powerful tool for image segmentation. the markers selection and extraction are not so easy. Some pictures may be very noisy and image processing becomes more and more complex. In this cases, the objects to be detected may be so complex and so varied in shape, grey level and size that it is very hard to find reliable algorithms enabling their extraction. Watershed transformation based segmentation is generally marker controlled segmentation. This method of image segmentation includes image enhancement and noise removal techniques [6]. Segmentation using the watershed transform works better if you can identify, or "mark," foreground objects and background locations. Marker-controlled watershed segmentation [37] follows this basic procedure:

1. Compute a segmentation function. This is an image whose dark regions are the objects you are trying to segment.
2. Compute foreground markers. These are connected blobs of pixels within each of the objects.
3. Compute background markers. These are pixels that are not part of any object.
4. Modify the segmentation function so that it only has minima at the foreground and background marker locations.
5. Compute the watershed transform of the modified segmentation function.

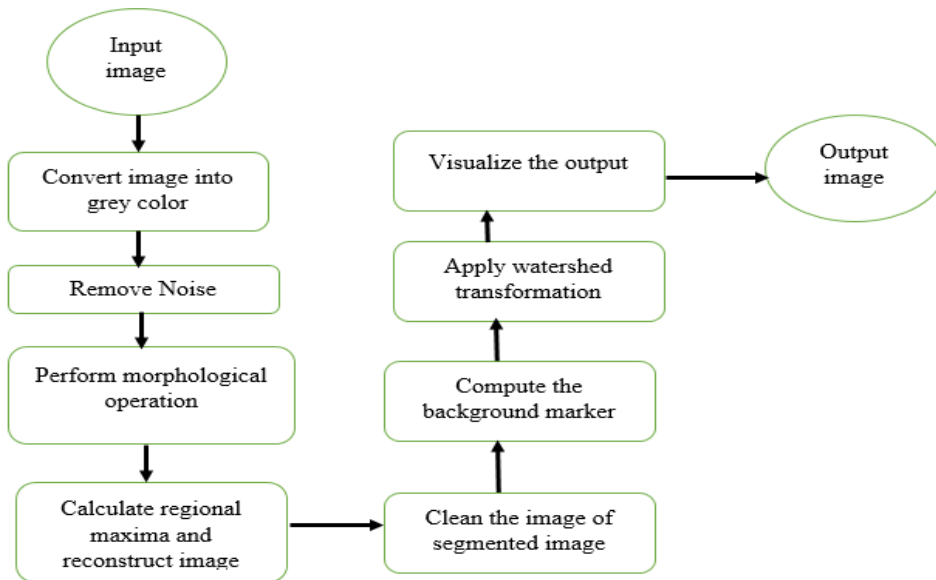


Figure 2. 13 Watershed segmentation process

#### 2.4.3.2. Otsu Thresholding Algorithm

Otsu's thresholding method corresponds to the linear discriminant criteria that assumes that the image consists of only object (foreground) and background, and the heterogeneity and diversity of the background is ignored [38]. the Otsu's method segments the image into two light and dark regions  $T_0$  and  $T_1$ , where region  $T_0$  is a set of intensity level from 0 to  $t$  or in set notation  $T_0 = \{0, 1, \dots, t\}$  and region  $T_1 = \{t, t + 1, \dots, l - 1, l\}$  where  $t$  is the threshold value,  $l$  is the image maximum gray level (for instance 256).

Suppose that a gray-level image  $f$  can take  $K$  possible gray levels  $0, 1, 2, \dots, K - 1$ . Define an integer threshold,  $T$ , that lies in the gray-scale range of  $T$  lies between  $(0, 1, 2, \dots, K - 1)$ . The process of thresholding is a process of simple comparison: each pixel value in  $f$  is compared to threshold,  $T$ . Based on this comparison; a binary decision is made that defines the value of the corresponding pixel in an output binary image  $g$ . If  $g(x, y)$  is a thresholded version of  $f(x, y)$  at some global threshold  $T$ .

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{cases} \dots\dots\dots \text{Equation 2. 2}$$

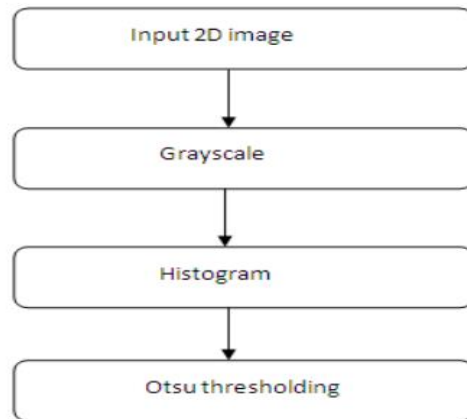


Figure 2. 14 Otsu-Thresholding

**Algorithm:**

Step 1: Compute histogram for a 2D image.

Step 2: Calculate foreground and background variances (measure of spread) for a single threshold.

i) Calculate weight of background pixels and foreground pixels.

ii) Calculate mean of background pixels and foreground pixels.

iii) Calculate variance of background pixels and foreground pixels.

Step 3: Calculate “within class variance”

Table 2. 4 Descriptions of segmentation Techniques

Segmentation technique	Description	Advantages	Disadvantages
Thresholding Method	based on the histogram peaks of the image to find particular threshold values	no need of previous information, simplest method	highly dependent on peaks, spatial details are not considered
Edge Based Method	based on discontinuity detection	good for images having better contrast between objects	not suitable for wrong detected or too many edges

Region Based Method	based on partitioning image into homogeneous regions	more immune to noise, useful when it is easy to define similarity criteria	expensive method in terms of time and memory
Clustering Method	based on division into homogeneous clusters	fuzzy uses partial membership therefore more useful for real problems	determining membership function is not easy
Watershed Method	based on topological interpretation	results are more stable, detected boundaries are continuous	complex calculation of gradients
PDE Based Method	based on the working of differential equations	fastest method, best for time critical applications	more computational complexity
ANN Based Method	based on the simulation of learning process for decision making	no need to write complex programs	more wastage of time in training

#### **2.4.4. Feature Extraction**

Feature extraction is a low-level image processing application. For a picture, the feature is the "interest" part [27]. In the pattern recognition literature, the name feature is frequently used to designate a descriptor. Repeatability is the desirable property of a feature detector. After image segmentation, the next step is to extract image features useful in describing target object.

After all the representation of the image important feature of the image is extracted using various types of feature extraction with respect to images, the similar features together form a feature vector to identify and classify an object [25]. Various features can be extracted from the image: color, shape, size, texture. There are some local feature detector and visual descriptor, which are used for object recognition and classification. Some of them are Speeded Up Robust Features (SURF), Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP) [27].

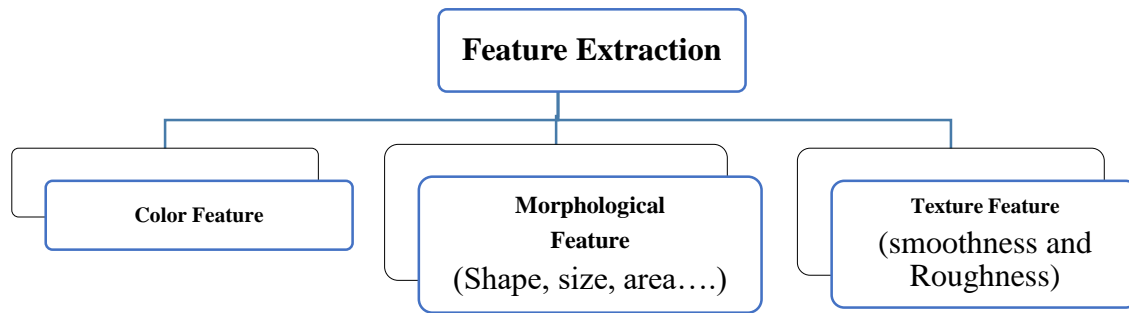


Figure 2. 15 Classification of Feature Extraction Method

#### 2.4.4.1. Color Feature Extraction

In image classification, the color is the most important feature. The color histogram represents the most common method to extract color feature. It is regarded as the distribution of the color in the image. The efficacy of the color feature resides in the fact that is independent and insensitive to size, rotation and the zoom of the image [39].

#### 2.4.4.2. Morphological Feature Extraction

##### Size feature extraction

For size feature extraction, different size measures, which are most commonly used, are area, perimeter, weight, height (length), width and volume. Some other measures for size feature extraction are radius, equatorial diameter, and major and minor axes [27].

##### Shape feature extraction

Shape features are very used in the literature (in object recognition and shape description). The shape features extraction techniques are classified as: region based and contour based. The contour methods calculate the feature from the boundary and ignore its interior, while the region methods calculate the feature from the entire region [39].

#### 2.4.4.3. Texture feature extraction

Texture feature extraction is very robust technique for a large image which contains a repetitive region. The texture is a group of pixel that has certain characterize. The texture feature methods are classified into two categories: spatial texture feature extraction and spectral texture feature extraction [40]. Texture analysis attempts to enumerate instinctive qualities defined by terms such as rough, smooth, silky, or bumpy as a function of the pixel intensities spatial variation. Texture is the repetition of pattern or patterns over a region in an image [41].

## **2.5. Machine Learning**

Machine learning (ML) is a subset of artificial intelligence that involves teaching a machine to make decisions using input data. It is used to build models with self-learning capabilities. These models are automatically trained to learn from experience or past data over and over again, and then provide knowledge to a machine. This technique is widely used for forecasting the future or categorizing data to assist people in making important decisions [35].

According to [28], The primary goal of machine learning (ML) is to create systems that can change their behavior on their own based on experience. ML methods employ training data to generate general models capable of detecting the presence or absence of new (test) data patterns. In the case of images, training data can be in the form of a set of pixels, regions, or images that can or cannot be labeled. Machine learning is used in a variety of fields, including hospitals, manufacturing industries, robotics, computer games, pattern recognition, natural language processing, image processing and classification, data mining, traffic prediction, product recommendation, marketing, medical diagnosis, agriculture advisory, e-mail spam filtering, crime prediction through video surveillance systems, and the like [35].

### **2.5.1. Classification approaches**

Image classification is defined as the process of assigning all pixels in an image to specific classes or themes based on spectral information represented by digital numbers (DNs) [42]. Modern computer vision and pattern recognition rely heavily on classification. The classifier's job is to assign an image or region of interest (RoI) to a category using the feature vector [43]. The classified image is a thematic map of the original image composed of a mosaic of pixels, each of which belongs to a specific theme. There are two approaches to image classification in general:

#### **2.5.1.1. Supervised Classification**

It is the process of identification of classes within a remote sensing data with inputs from and as directed by the user in the form of training data [42], and or it is methods approximate a mapping function  $f(x)$  which can predict the output variables  $y$  for a given input sample  $x$  [43].

#### **2.5.1.2. Unsupervised Classification:**

It is the process of automatic identification of natural groups or structures within a remote sensing data [42]. It is where one would only have input data  $X$  and no corresponding output variables

[43]. The degree of difficulty of the classification task depends on the variability in the feature values of images from the same category, relative to the difference between feature values of images from different categories. However, a perfect classification performance is often impossible. This is mainly due to the presence of noise (in the form of shadows, occlusions, perspective distortions, etc.), outliers (e.g., images from the category “buildings” might contain people, animal, building, or car category), ambiguity (e.g., the same rectangular shape could correspond to a table or a building window), the lack of labels, the availability of only small training samples, and the imbalance of positive/negative coverage in the training data samples. Thus, designing a classifier to make the best decision is a challenging task [43].

### **2.5.2. Artificial Neural Network**

Artificial Neural Networks (ANN) is an attempt to mimic the neurons of the brain. The models used, however, have many simplifications and thus do not reflect the true behavior of the brain. Development of ANN had its first peak in the 1940s, and development has since gone up and down [28]. According to [44], The learning part of ANN comes into play when determining how much weight each neuron should have in order to achieve the desired result. ANNs tend to find patterns in the examples provided during the learning process. Supervised learning is one method of learning with an ANN. The network provides labeled data for all possible inputs in supervised learning. Based on the labeled data, the weights are then calculated to produce answers. Overfitting may occur for the network if the examples are insufficient. This occurs when the ANN performs well on training data but poorly on unseen examples. There are 4 main learning rules, error correction is one of them. Error correction falls under supervised learning, and the basic idea is to use the error to update the weights and reduce the error over each iteration [28].

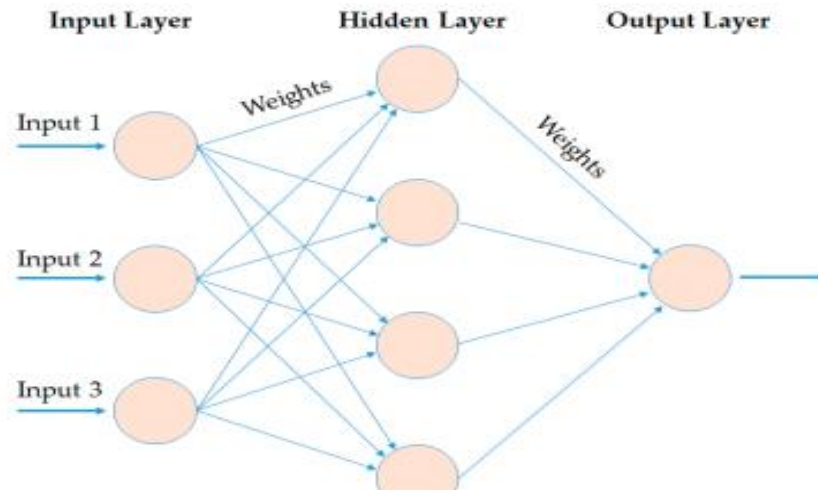


Figure 2. 16 Schematic of a typical Artificial Neural Network (ANN) architecture

## 2.6. Deep Learning Approach.

Deep learning is a subfield of machine learning, which is, in turn, a subfield of artificial intelligence (AI). For a graphical depiction of this relationship, please refer to figure 2.16. Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but nonlinear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. The key aspect of deep learning is that these layers are not designed by human engineers: they are learned from data using a general-purpose learning procedure [45].

As described in [35], at the end of the day we are using multi neural network architectures by creating deep learning neural networks, the main idea behind deep learning is to mimic the human brain like how human learn a thing by the help of human brain. Similarly, we are creating models that are learning these things using deep learning methods. The most popular deep learning methods have been widely used are Convolutional neural network(CNN), Denoising autoencoder (DAE), deep belief networks (DBNs), Long Short-Term Memory (LSTM) and denoising Auto Encoder (DAE).



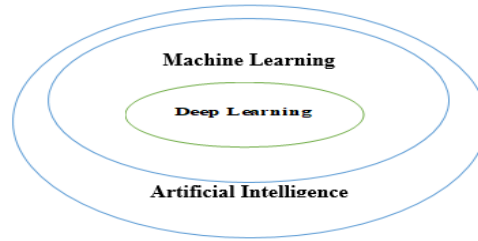


Figure 2. 17 Venn diagram which describes deep learning [9]

## Why Deep Learning?

While these machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical computations to large-scale data is a recent development. For example, with more computing power and a large enough memory, one can create neural networks of many layers, which are called deep neural networks.

The main advantage of deep learning is simplicity, which means that deep networks provide basic architectural blocks, network layers, which are repeated several times to generate large networks, and scalability, which means that deep learning models are easily scalable to massive datasets. Other competing methods, such as kernel machines, run into serious computational issues when the datasets are large, and domain transfer, which means that a model learned on one task is applicable to other related tasks and the learned features are general enough to work on a variety of tasks with scarce data [43].

### 2.6.1. Recurrent Neural Network (RNN)

The recurrent neural network was first developed in the 1980s [46]. Its structure consists of an input layer, one or more hidden layers, and an output layer. RNNs have chain-like structures of repeating modules with the idea behind using these modules as a memory to store important information from previous processing steps. Unlike feedforward neural networks, RNNs include a feedback loop that allows the neural network to accept a sequence of inputs. This means the output from step  $t - 1$  is fed back into the network to influence the outcome of step  $t$ , and for each subsequent step. Therefore, RNNs have been successful in learning sequences. Figure 2.18 shows the sequential [47].

Another widely used and popular algorithm in deep learning, especially in NLP and speech processing, is RNN [46]. Unlike traditional neural networks, RNN utilizes the sequential information in the network. This property is essential in many applications where the embedded

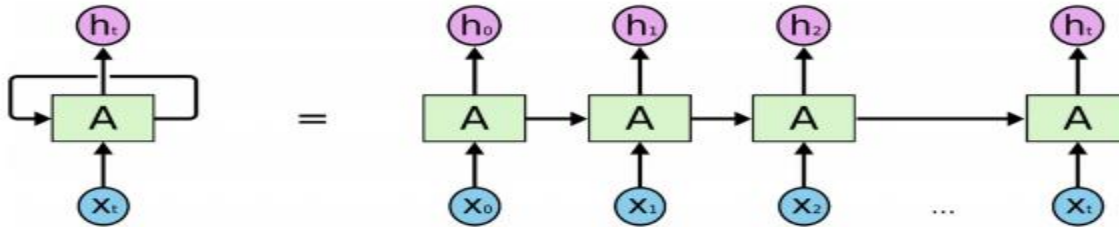


Figure 2. 18 RNN structure

structure in the data sequence conveys useful knowledge. For example, to understand a word in a sentence, it is necessary to know the context. Therefore, an RNN can be seen as short-term memory units that include the input layer  $x$ , hidden (state) layer  $s$ , and output layer  $y$ .

above Figure 2.17 illustrates a simple RNN with one input unit, one output unit, and one recurrent hidden unit expanded into a full network, where  $X_t$  is the input at time step  $t$  and  $h_t$  is the output at time step  $t$ . During the training process, RNN uses a backpropagation algorithm, a prevalent algorithm applied in calculating gradients and adjusting weight matrices in ANN. However, it will adjust and the weights following the modification of the feedback process.

One main issue of an RNN is its sensitivity to the vanishing and exploding gradients [47]. In other words, the gradients might decay or explode exponentially due to the multiplications of lots of small or big derivatives during the training. This sensitivity reduces over time, which means the network forgets the initial inputs with the entrance of the new ones. Therefore, Long Short-Term Memory (LSTM) is utilized to handle this issue by providing memory blocks in its recurrent connections.

### 2.6.2. Long Short-Term Memory (LSTM) Neural Network

Long Short-Term Memory, an evolution of RNN, was introduced by Hochreiter and Schmidhuber [37] to address problems of the aforementioned drawbacks of the RNN by adding additional interactions per module (or cell). LSTMs are a special kind of RNN, capable of learning long-term dependencies and remembering information for prolonged periods of time as a default. According to Olah, the LSTM model is organized in the form of a chain structure. However, the repeating module has a different structure. Instead of a single neural network like a standard RNN, it has

four interacting layers with a unique method of communication. The structure of the LSTM neural network is shown in Figure 2.19.

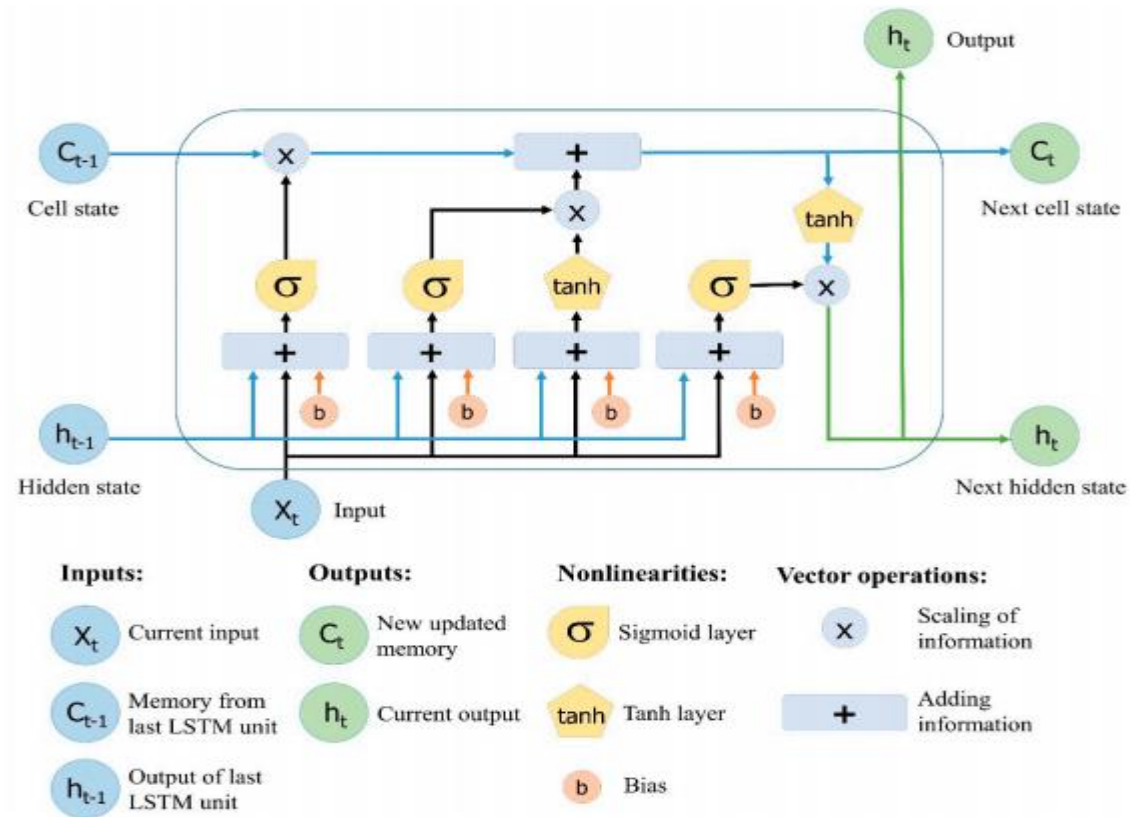


Figure 2. 19 Long Short-Term Memory neural network

In figure 2.19,  $C$ ,  $x$ ,  $h$  represent cell, input and output values. Subscript  $t$  denotes time step value, i.e.,  $t - 1$  is from previous LSTM block (or from time  $t - 1$ ) and  $t$  denotes current block values. The symbol  $\sigma$  is the sigmoid function and  $\tanh$  is the hyperbolic tangent function. Operator  $+$  is the element-wise summation and  $\times$  is the element-wise multiplication [47].

### 2.6.3. Convolutional Neural Network

Convolutional neural network (CNN) is a multi-layered neural network with a unique architecture designed to extract increasingly complex features of the data at each layer to determine the output. CNN's are well suited for perceptual tasks [48].

A convolutional neural network (CNN) is a deep, feed-forward artificial neural network in which the neural network preserves the hierarchical structure by learning internal feature representations and generalizing the features in the common image problems like object recognition and other

computer vision problems. It is not restricted to images; it also achieves state-of-the-art results in natural language processing problems and speech recognition [49].

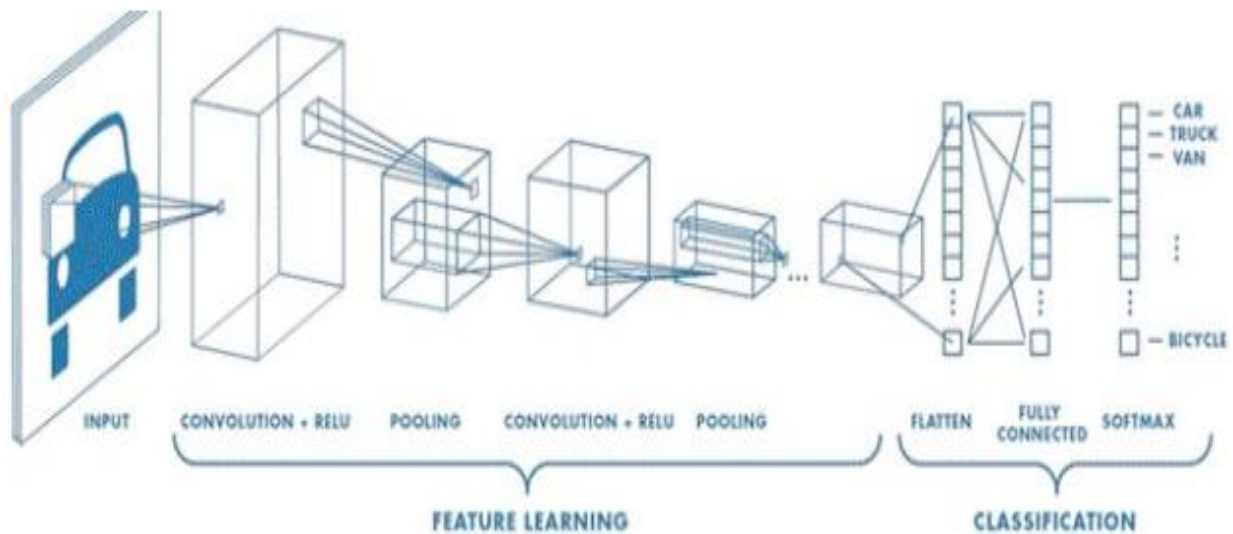


Figure 2. 20 Architecture of convolutional neural network

CNN is mostly used when there is an unstructured data set (e.g., images) and the practitioners need to extract information from it. For instance, if the task is to predict an image caption: CNN is typically used when practitioners need to extract information from an unstructured data set (e.g., images). For example, suppose the task is to predict an image caption: The CNN receives an image of, say, a cat; this image is a collection of pixels in computer terms. In general, one layer for a greyscale image and three layers for a color image. During feature learning (i.e., hidden layers), the network will identify unique features such as the cat's tail, ear, and so on. When the network has thoroughly learned how to recognize a picture, it can provide a probability for each image that it is familiar with. The label with the highest probability will become the prediction of the network [48]. Network consists of three types of layers namely convolution layer, sub sampling layer and the output layer [50].

### 2.6.3.1. Convolutional layer

The convolutional layers are the core building block of the convolutional network. Most heavy computational lifting is carried out in this layer. Parameters of convolutional layers are made of learnable filters, sometimes referred to as kernels. Each filter consists of a small area of an image, which is called a receptive field. The filter extends to a depth similar to the full depth of the input

volume. The filter is an array of numbers on a layer of the convolutional network with a size,  $w \times h \times d$  ( $w$  pixels width,  $h$  height, and depth  $d$ , color channels). As the filter slides or convolves across the width and height of the input image, it multiplies the values in the filter with the pixel values of the input [51].

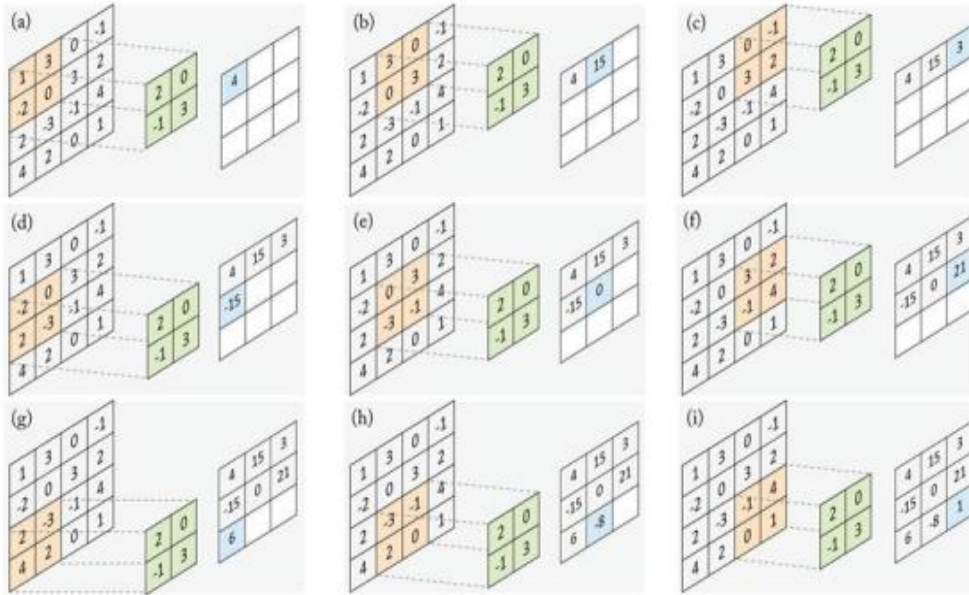


Figure 2. 2D convolution operation using same padding and one stride

Given a 2D input feature map and a convolution filter of matrix sizes  $4 \times 4$  and  $2 \times 2$ , respectively, a convolution layer multiplies the  $2 \times 2$  filter with a highlighted patch (also  $2 \times 2$ ) of the input feature map and sums up all values to generate one value in the output feature map. Note that the filter slides along the width and height of the input feature map and this process continues until the filter can no longer slide further [43].

Let us feature map =  $A_i$ , input image size =  $A_k$ , filter or kernel size =  $A_{ik}$ , the order to find the feature map we can use the formula below.

$$A_i = \sum(A_{ik} * A_k) \dots \dots \dots \text{Equation 2. 3}$$

**2.6.3.2. Pooling Layer**

Pooling is an important concept of CNN. It lowers the computational burden by reducing the number of connections between convolutional layers [52].

Pooling is a technique for reducing dimensionality by selecting the most representative feature. This allows us to reduce the number of parameters, which both shortens training time and combats overfitting. Pooling layers downsample each feature map independently, reducing the height and width while leaving the depth intact [53]. The most common type of pooling is max pooling, which simply takes the maximum value in the pooling window. One limitation of convolutional layer feature map output is that it records the precise position of features in the input. This means that even minor changes in the position of the feature in the input image result in a different feature map. This can occur when the input image is re-cropped, rotated, shifted, or other minor changes are made.

The most common type of pooling is max pooling which just takes the max value in the pooling window. Contrary to the convolution operation, pooling has no parameters. It slides a window over its input, and simply takes the max value in the window. Similar to a convolution, we specify the window size and stride [54]. The another type of pooling average pooling which compute the average value and takes that value in the pooling window.

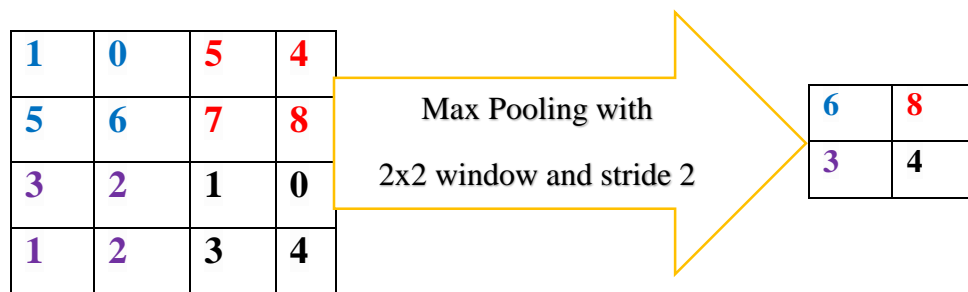


Figure 2. 22 Max Pooling

The above figure 2.22 show that, pooling window 2x2 and stride 2 and then compare the value takes the max value.

### 2.6.3.3. fully connected layer

Remember that the output of both convolution and pooling layers are 3D volumes, but a fully connected layer expects a 1D vector of numbers. So we flatten the output of the final pooling layer to a vector and that becomes the input to the fully connected layer. Flattening is simply arranging the 3D volume of numbers into a 1D vector, nothing fancy happens here [54].

#### 2.6.3.4. Activation Function

The idea of an activation function comes from the analysis of how a neuron works in the human brain. The neuron becomes active beyond a certain threshold, better known as the activation potential. It also attempts to put the output into a small range in most cases. Sigmoid, hyperbolic tangent (tanh), ReLU, softmax and LeakyReLU are most popular activation functions [49].

#### ReLU (Rectified Linear Unit) Function

The sigmoid is not the only kind of smooth activation function used for neural networks. Recently, a very simple function named ReLU (REctified Linear Unit) became very popular because it helps address some optimization problems observed with sigmoids. The function is zero for negative values and it grows linearly for positive values [53].

$$f(x) = \max(0, Z) \dots \dots \dots \text{Equation 2. 4}$$

#### Softmax

The softmax function of z, is a generalization of the sigmoid function which represents a probability distribution over a discrete variable with n possible values. The softmax function generally used to treat the output of each unit as the probability belonging to each class and described as Eq. (2.5).

$$f(x) = \frac{e^x}{\sum_{k=1}^k e^x} \dots \dots \dots \text{Equation 2. 5}$$

#### 2.6.3.5. Hyperparameters

**Stride:** It specifies how much we move the convolution filter at each step. By default, the value is 1 [54].

**Filter size:** The most typically used filter size are, 3x3 filters, but 5x5 or 7x7 are also used depending on the application [54].

**Padding:** It is commonly used in CNN to preserve the size of the feature maps, otherwise they would shrink at each layer, which is not desirable. The most common value for padding hyperparameter is same or zero which indicates the input and the output have the same size [54].

**Batch size:** The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

**Epoch:** The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.

**2.6.3.6. Loss Functions**

“loss function,” also called the “objective function,” to estimate the quality of predictions made by the network on the training data, for which the actual labels are known. These loss functions are optimized during the learning process of a CNN. A loss function quantifies the difference between the estimated output of the model (the prediction) and the correct output (the ground truth). The type of loss function used in our CNN model depends on our end problem.

**Cross-Entropy Loss**

The cross-entropy loss (also termed “log loss” and “soft-max loss”) is defined as follows:

$$f(p, y) = - \sum_{n=1}^n y_n \log(p_n) \quad n \in [1, N] \dots \dots \dots \text{Equation 2. 6}$$

where y denotes the desired output and p is the probability for each output category. There is a total of N neurons in the output layer, therefore  $p, y \in \mathbb{R}^n$  The probability of each class can be calculated using a soft-max function [43].

$$\sigma(\mathbf{z})_j = \frac{e^{\mathbf{z}_j}}{\sum_{k=1}^k e^{\mathbf{z}_k}} \dots \dots \dots \text{Equation 2. 7}$$

for  $j = 1 \dots K$ . We can see that the softmax function normalizes a  $K$  dimensional vector  $\mathbf{z}$  of arbitrary real values into a  $K$  dimensional vector  $\sigma(\mathbf{z})$  whose components sum to 1.

**2.6.3.7. Regularization of Convolutional Neural Network**

Since deep neural networks have a large number of parameters, they tend to over-fit on the training data during the learning process. By over-fitting, we mean that the model performs really well on the training data but it fails to generalize well to unseen data. It, therefore, results in an inferior performance on new data (usually the test set). Regularization approaches aim to avoid this problem using several intuitive ideas which we discuss below.



## Data augmentation

Data augmentation is the easiest, and often a very effective way of enhancing the generalization power of CNN models. Especially for cases where the number of training examples is relatively low, data augmentation can enlarge the dataset (by factors of 16x, 32x, 64x, or even more) to allow a more robust training of large-scale models. Data augmentation is performed by making several copies from a single image using straightforward operations such as rotations, cropping, flipping, scaling, translations, and shearing [55].

**Horizontal and Vertical Shift Augmentation:** A shift to an image is the movement of all pixels in one direction, such as horizontally or vertically, while keeping the image dimensions constant. This means that some pixels will be clipped from the image, and a region of the image will require new pixel values to be specified. The ImageDataGenerator constructor's width shift range and height shift range arguments control the amount of horizontal and vertical shift, respectively. These arguments can include a floating point value indicating the percentage (between 0 and 1) of the image's width or height to shift. Alternately, a number of pixels can be specified to shift the image. Specifically, a value in the range between no shift and the percentage or pixel value will be sampled for each image and the shift performed, e.g. [0, value]. Alternately, you can specify a tuple or array of the min and max range from which the shift will be sampled; for example: [-100, 100] or [-0.5, 0.5] [56].

**Horizontal and Vertical Flip Augmentation:** In the case of a vertical or horizontal flip, an image flip means reversing the rows or columns of pixels. A boolean horizontal flip or vertical flip argument to the ImageDataGenerator class constructor specifies the flip augmentation [56].

**Random Brightness Augmentation:** The brightness of the image can be increased by randomly darkening, brightening, or both. The goal is for a model to generalize across images trained under different lighting conditions. This can be accomplished by passing the brightness range argument to the ImageDataGenerator constructor, which specifies the min and max range as a float representing a percentage for determining the amount of brightening. Values less than 1.0 darken the image, e.g. [0.5, 1.0], whereas values larger than 1.0 brighten the image, e.g. [1.0, 1.5], where 1.0 has no effect on brightness [56].

**Random Zoom Augmentation:** A zoom augmentation randomly zooms in on an image and either adds new pixel values around it or interpolates pixel values. The zoom range argument to the ImageDataGenerator constructor can be used to configure image zooming. The zoom percentage can be specified as a single float or as a range as an array or tuple.

If a float is specified, then the range for the zoom will be  $[1-\text{value}, 1+\text{value}]$ . For example, if you specify 0.3, then the range will be  $[0.7, 1.3]$ , or between 70% (zoom in) and 130% (zoom out). The zoom amount is uniformly randomly sampled from the zoom region for each dimension (width, height) separately. The zoom may not feel intuitive. Note that zoom values less than 1.0 will zoom the image in, e.g.  $[0.5, 0.5]$  makes the object in the image 50% larger or closer, and values larger than 1.0 will zoom the image out by 50%, e.g.  $[1.5, 1.5]$  makes the object in the image smaller or further away. A zoom of  $[1.0, 1.0]$  has no effect [56].

## **Dropout**

During network training, each neuron is activated with a fixed probability (usually 0.5 or set using a validation set). This random sampling of a sub-network within the full-scale network introduces an ensemble effect during the testing phase, where the full network is used to perform prediction. Activation dropout works really well for regularization purposes and gives a significant boost in performance on unseen data in the test phase. Dropout has predominantly been applied to fully-connected(FC) layer.

### **2.6.3.8. Gradient-Based Convolutional Neural Network Learning and Optimizer**

The CNN learning process tunes the parameters such as weight and biases of the convolution layer of the network such that the input space is correctly mapped to the output space. As discussed before, at each training step, the current estimate of the output variables is matched with the desired output (often termed the “groundtruth” or the “label space”). This matching function serves as an objective function during the CNN training and it is usually called the loss function or the error function. In other words, we can say that the CNN training process involves the optimization of its parameters such that the loss function is minimized. The amount of parameter update, or the size of the update step is called the “learning rate.” Each iteration which updates the parameters using the complete training set is called a “training epoch.” We can write each training iteration at time  $t$  using the following parameter update equation:

$$\theta_t = \theta_{t-1} - \eta \delta_t \dots \dots \dots \text{Equation 2. 8}$$

$$\delta_t = \nabla_{\theta} F(\theta_t) \dots \dots \dots \text{Equation 2. 9}$$

Where  $F(\cdot)$  denotes the function represented by the neural network with parameters  $\theta$ ,  $\nabla$  represents the gradient, and  $\eta$  denotes the learning rate.

**Stochastic Gradient Descent**

Stochastic Gradient Descent (SGD) updates the parameters for each set of input and output in the training set. As a result, compared to batch gradient descent, it converges much faster. Furthermore, it can learn "online," where the parameters can be fine-tuned in the presence of new training examples. The only issue is that its convergence behavior is typically unstable, especially at relatively higher learning rates and when training datasets contain a variety of examples. When the learning rate is appropriately set, the SGD achieves similar convergence behavior to batch gradient descent for both convex and non-convex problems. [57].

**Adaptive Moment Estimation**

Adam is one of the most widely used optimization algorithms in deep learning. The name Adam is derived from adaptive moment estimation because it computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam combines the advantages of AdaGrad which works well with sparse gradients and RMSProp which works well in online and non-stationary settings. Adaptive Moment Estimation (ADAM) facilitates computation of learning rates for each parameter using first and second moment of gradient [57].

Being computationally efficient, ADAM requires less memory and outperforms on large datasets. It require  $p_2, q_2, t$  to be initialized to  $\mathbf{0}$ , where  $p_0$  corresponds to  $1^{st}$  moment vector i.e. mean,  $q_0$  corresponds to  $2^{nd}$  moment vector i.e. uncentered variance and  $t$  represents timestep. While considering  $f(w)$  to be the stochastic objective function with parameters  $w$ , proposed values of parameters in ADAM, are as follows:

$$\alpha = 0.001, m_1=0.9, m_2=0.999, \epsilon = 10^{-8}.$$

Another major advantage discussed in the study of ADAM is that the updation of parameter is completely invariant to gradient rescaling, the algorithm will converge even if objective function

changes with time. The drawback of this particular technique is that it requires computation of second-order derivative which results in increased cost [58].

The algorithm of ADAM has been briefly mentioned below –

Input: Weight of each neuron

```

do{
    Step 1: while  $w_t$  do not converges
    Step 2: Calculate gradient  $g_t = \frac{\partial f(x,w)}{\partial w}$ 
    Step 3: Calculate  $p_t = m_1 \cdot p_{t-1} + (1 - m_1) \cdot g_t$ 
    Step 4: Calculate  $q_t = m_2 \cdot q_{t-1} + (1 - m_2) \cdot g_t^2$ 
    Step 5: Calculate  $\hat{p}_t = p_t / (1 - m_1^t)$ 
    Step 6: Calculate  $\hat{q}_t = q_t / (1 - m_2^t)$ 
    Step 7: Update the parameter  $w_t = w_{t-1} - \alpha \cdot \hat{p}_t / (\sqrt{\hat{q}_t} + \epsilon)$ 
}
    Step 8: return  $w_t$ 

```

Output: Updated weight Value

Figure 2. 23 ADAM algorithms

## AdaDelta

The underlying idea of AdaDelta algorithm is to improve the two main drawbacks of AdaGrad: The continual decay of learning rates throughout training and the need for a manually selected global learning rate. To this end, AdaDelta restricts the window of past gradients to be some fixed size  $w$  instead of accumulating the sum of squared gradients over all time. As mentioned in the previous section, AdaGrad accumulates the squared gradients from each iteration starting at the beginning of training. This accumulated sum continues to grow during training, effectively shrinking the learning rate on each dimension. After many iterations, the learning rate becomes infinitesimally small. With the windowed accumulation AdaGrad becomes a local estimate using recent gradients instead of accumulating to infinity. Thus, learning continues to make progress even after many iterations of updates have been done [57].

## RMSProp

Another algorithm that modifies AdaGrad is RMSProp. It is proposed to perform better in the nonconvex setting by changing the gradient accumulation into an exponentially weighted moving average. AdaGrad shrinks the learning rate according to the entire history of the squared gradient. Instead, RMSProp uses an exponentially decaying average to discard history from the extreme past so that it can converge rapidly after finding a convex bowl [57].

## **2.6.4. Examples of CNN Architectures**

Various architectures have been developed and implemented in CNN. Practically most CNN architectures follow the same universal design philosophies of convolutional layers to the input image, pooling layer for extract important feature, finally, fully connected layer, activation function, and dropout layers are performed based on the convolution that gets the best architecture and some architectures used batch normalization for accelerating training of the model and reduce covariant shift [59]. Brief explanations of those architectures are explained below.

### **2.6.4.1. AlexNet Architecture**

AlexNet [33] [Alex Krizhevsky ImageNet Classification with Deep Convolutional Neural Networks] was the first large-scale CNN model which led to the resurgence of deep neural networks in computer vision. This architecture won the ImageNet LargeScale Visual Recognition Challenge (ILSVRC) in 2012 by a large margin. The main difference between the AlexNet architecture and LeNet were relatively smaller and were not tested on large-scale datasets such as the ImageNet dataset but, AlexNet have increased network depth, which leads to a significantly larger number of tunable parameters, and used regularization tricks (such as the activation dropout and data augmentation) [43].

It consists of three fully connected and five convolution layers, the outputs received are passed to the 1000-way softmax in order to classify 1.2 million high resolution images into 1000 distinct classes on the ImageNet dataset [60]. Note that dropout is applied after the first two fully connected layers in the AlexNet architecture, which leads to a reduced over-fitting and a better generalization to unseen examples. Another distinguishing aspect of AlexNet is the usage of ReLU nonlinearity after every convolutional and fully connected layer, which substantially improves the training efficiency compared to the traditionally used tanh function.

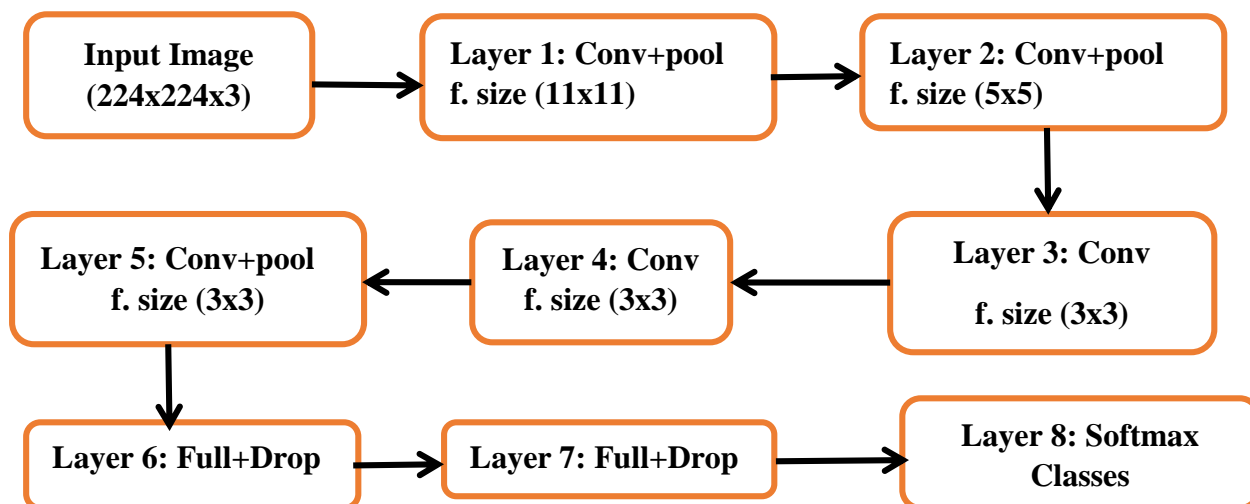


Figure 2. 24 AlexNet Architecture

### 2.6.4.2. VGG Architecture

In 2012 AlexNet sparked a new interest in CNN, by showing great results in the ImageNet challenge. Two years later, a new CNN model called VGG entered into the 2014 competition. Two different VGG models were used: VGG16 and VGG19 (the number represents the number of layers in the network). VGG19 managed to win the challenge. This was a new deeper model than what had previously been used, which led to deeper networks becoming prominent within the field.

The VGGnet architecture strictly uses 3x3 convolution kernels with intermediate max-pooling layers for feature extraction and a set of three fully connected layers toward the end for classification. Each convolution layer is followed by a ReLU layer in the VGGnet architecture. The design choice of using smaller kernels leads to a relatively reduced number of parameters, and therefore an efficient training and testing. Similar to AlexNet, it also uses activation dropouts in the first two fully connected layers to avoid over-fitting [43].

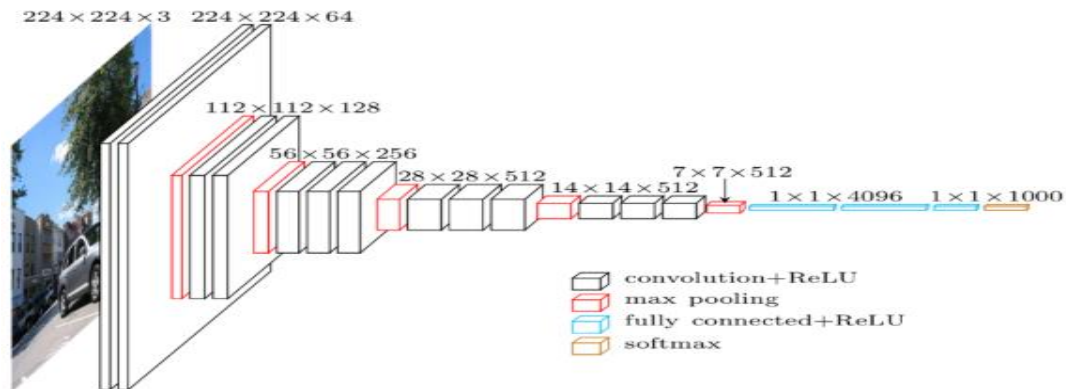


Figure 2. 25 VGGNet architecture

### 2.6.4.3. Inception (GoogLeNet) Architecture

In 2014, researchers at Google introduced the Inception network which took first place in the 2014 ImageNet competition for classification and detection challenges [59].

GoogleNet consists of a total of 22 weight layers. The basic building block of the network is the “Inception Module,” due to which the architecture is also commonly called the “Inception Network” [43]. in which to perform a series of convolutions at different scales and subsequently aggregate the results. In order to save computation, 1x1 convolutions are used to reduce the input channel depth [59].

Although the GoogleNet architecture looks much more complex than its predecessors, e.g., AlexNet and VGGnet, it involves a significantly reduced number of parameters (~6 million compared to 62 million in AlexNet and 138 million parameters in VGGnet). With a much smaller memory footprint, a better efficiency and a high accuracy, GoogleNet is one of the most intuitive CNN architectures which clearly demonstrates the importance of good design choices [43].

### 2.6.4.4. RESNET Architecture

The Residual Network from Microsoft won the ILSVRC 2015 challenge with a big leap in performance, reducing the top-5 error rate to 3.6% from the previous year’s winning performance of 6.7% by GoogleNet.

The remarkable feature of the residual network architecture is the identity skip connections in the residual blocks, which allows to easily train very deep CNN architectures [43].

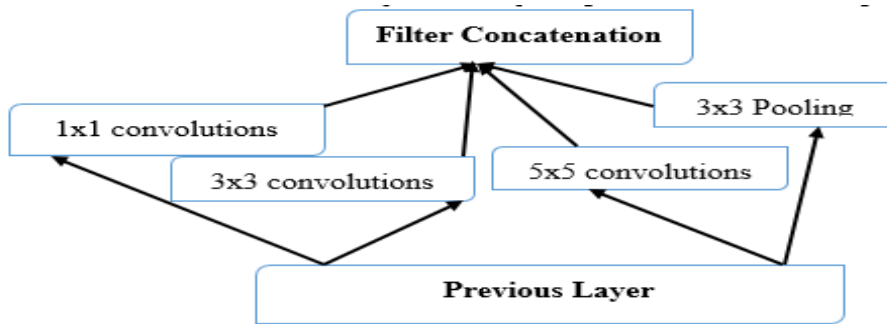


figure 2. 26 Inception module, naïve version

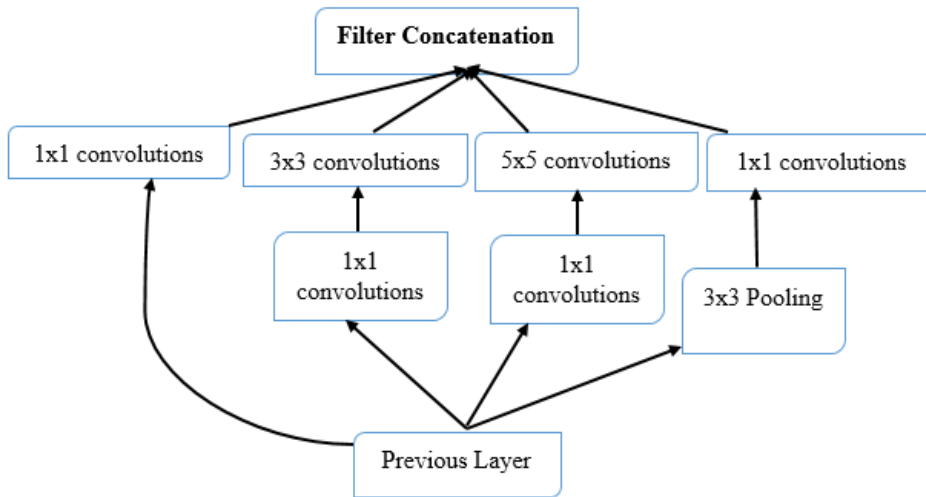


Figure 2. 27 Inception module with dimensionality reduction

## 2.7. Related Works

Correct classification and sorting of food or agricultural product raises expectations with respect to quality food and safety standards. Thus, computer vision and image processing were nondestructive, precise, and effective methods for achieving the goal of sample product classification and grading [3].

### 2.7.1. Related to Coffee and Crop Product

According to [61], they proposed Ethiopian Coffee Plant Diseases Recognition supported Imaging and Machine Learning Techniques. This research work focuses on three major sort of coffee disease which occurs on the leave a part of a coffee plant, these are Coffee Leaf Rust (CLR), Coffee Berry Disease (CBD), and occasional wilt (CWD). The aim of this paper is recognition of the three sorts of coffee disease using imaging and machine learning techniques. The image of Coffee plant diseases were taken from the regions of Ethiopia where more coffee is produced i.e. Southern Nations,



Nationalities, and Peoples, Jimma and Zegie. During this paper artificial neural network (ANN), k-Nearest Neighbours (KNN), Naïve and a hybrid of self-organizing map (SOM) and Radial basis function (RBF) are used. They were conducted experiment for every group of feature set so as to urge a highly correlated and therefore the more representing features. The total number of images is 9100. The general result showed that color features represent quite texture features regarding recognition of coffee plant diseases and therefore the performance of combination of RBF (Radial basis function) and SOM (Self organizing map) is 90.07%. This study, however, is limited to coffee leaf disease detection and identification using different machine learning approaches. Using machine learning algorithms also makes feature extraction process time-consuming which needs to be changed whenever the problem or the dataset changes.

According to [62], they proposed Ethiopian Roasted Coffee Classification Using Imaging Techniques. This research work aimed to classify roasted coffee supported their growing region into five different growing regions (Jimma, Wollega, Yirgacheffe, Wonbera, and Zegie) in Ethiopia. When the coffee is with roasted state the form, color, density, and weight are different from its previous state. For classification of roasted coffee sample images was taken from each region and from each region 30 images in total of 150 images were taken and 14 features are extracted 6 of color 5 of morphology and three of texture for classification purpose. The green coffee taken from each region was roasted for various interval (2,3,4,5 and 9 minutes). Through classification of every region roasted coffee, we've compared the classification results of the neural network classifier using features of color, morphology, texture and a mixture of every. From the entire of 150 images 100 images were used as training and 50 images were used as a test set. because the results of neural net classifier, 86.2%, 76.5%, 96.4%, 79.9% and 98.2% scores were observed for the samples parameter of color, texture, morphology, combination of morphology and texture, and a mixture of morphology and color respectively. The obtained results have reported that neural net classifier using the feature of samples' morphology and color combination of roasted coffee achieved better result. However, in this study the researcher is limited only on roasted coffee bean. The features for roasted and unroasted coffee bean is totally different.

According to [3], the classification and gradation of sesame grain has been proposed using image processing techniques based on the Ethiopian commodity exchange criteria. Based on

their growing area, the researcher aimed to sort and classify sesame grain into the corresponding three groups (white Humera, white Wollega and reddish Wollega and five grades (grade one, grade two, grade three, grade four and under grade. The researcher takes pictures of sample sesame grains and processes the image to line the classes and grades. A segmentation technique is proposed to segment the foreground from the background, partitioning both sesame grains and foreign particles. The segmentation process also forms the bottom work from which feature extractions are made. Color structure tensor is applied to return up with a far better preprocessing, segmentation and have extraction activities. Furthermore, watershed segmentation is applied to separate connected objects. The delta E standard color difference algorithm, which generates six color features, is employed for classification of sesame grain samples. These six color features are used as inputs for classification and therefore the system generates 3 outputs like classes (types) of Ethiopian sesame grains. Grading of sesame grain samples is performed employing a rule based approach, where the classification output is going to be fed with 4 inputs and five or six outputs, like the morphological (size and shape) features and grades, respectively. On top of that, calibration is introduced to standardize the whole system. Experiments were administered to gauge the performance of our proposed system design. The classifier achieved an overall accuracy of 88.2%. For grading of sesame grain samples, we got an accuracy of 93.3%, much better than the manual way of grading. However, used human interpretation of feature extraction from sesame grain images of the dataset is time consuming and only applicable for sesame classification and grading and this study highly dependent on color. According to [26], the researchers were developed image based flower disease detection and identification using artificial neural network so as to group the subsequent corresponding classes such as:- rose-aphid, rose-japanese- beetle, rose-rosettle-disease, rose-goldenrod-solidier, rose-mossay rose-gell-wasp, rose- normal, rose-rust, and rose-solidier-beetles. They mainly classified their work into two main phases. In first phase, normal and diseased flower image are to create a knowledge domain. During the creation of the knowledge domain, images are pre-processed and segmented to spot the region of interest. Then, seven different texture features of images are extracted using Gabor texture feature extraction. Finally, a man-made neural network is trained using seven input features extracted from the individual image and eight output vectors that represent eight different classes of disease to represent the knowledge domain. In second phase, the knowledge domain is employed to

spot the disease of a flower. so as to make the knowledge domain and to check the effectiveness of the developed system, they have used 40 flower images for every of the eight different classes of flower disease and that we have a complete of 320 flower images. From those images 85% of the Dataset is employed for training and 15% of the info set is employed for testing. The experimental result demonstrates that the proposed technique is effective technique for the identification of flower disease. The developed system can successfully identify the examined flower with an accuracy of 83.3%. However, feature extraction is time-consuming which needs to be changed whenever the problem or the dataset changes and feature extraction expansive since manual base feature extraction applied.

Asma Redi [11], produced raw quality, Ethiopian coffee bean value classification in the case of Wollega area. This research work uses various techniques to eliminate noise from the image in the picture. Subtraction of the background was performed to prevent blur, light reflections, and other sounds that may be created on the background due to lighting effects and certain external artifacts.

Image enhancement and histogram thresholding were used for the extraction of morphological features and color features from the sting images of the 7 grade levels of Wollega coffee beans. A combined morphological and color features aggregate function dataset were used to develop the classification model. The classification models were built with the Naïve Bayes, C4.5 and ANN yielding a performance of 82.72%, 82.09% and 80.25%, respectively. so on reinforce the classification performance discretization of the dataset into raw quality, value into three beans were used. Regression model for the relation between the raw quality values and thus the combined aggregate feature values of the sample coffee beans were designed to support suitability and accuracy of the dataset for classification. However, this research work was limited to classification model for raw quality, value classification purposes by utilizing a smaller number of datasets from each grade level of the coffee berry sample. In addition to this, this study is limited for only Wollega region coffee and it didn't consider other varieties of coffee since the physical and chemical characteristics of each variety is different. Also the researcher applied machine learning algorithm to classify the coffee bean into corresponding classes with small number of data. But, to extract features in deep and to increase the classification performance of model it is better to apply deep learning approach with large number of images.

Habtamu Minasie [12], has developed automated coffee berry classification by taking a coffee berry image using machine learning techniques supported morphological and color features was developed to classify different sorts of Ethiopian coffee supported their growing region such as: - (Bale, Nekempti, Jimma, Limu, Sidamo and Welega). The entire number of images taken was 309 which contain 4844 coffee beans. For the classification analysis, ten morphological and 6 color, features were extracted from each coffee berry images. The researcher used Naïve Bayes and Neural Network classification approaches of classifiers on each classification parameter of morphology, color and therefore the combination of the two. The arrangement was supervised like the predefined classes of the growing regions. The researcher was also shown that the discrimination power of morphological features was better than color features but when both morphology and color features were used together the classification accuracy was increased. the simplest classification accuracies (80.7%, 72.6%, 56.8%, 96.77%, 95.42% and 69.9% for Bale, Nekempti, Jimma, Limu, Sidamo and Welega respectively) were obtained using neural networks when both morphology and color features were used together. The general classification accuracy was 77.4%. This study, however, is limited to six coffee bean classes and the researcher didn't incorporate Yirgacheffe, Kaffa and Gujji. Deep learning approach is the best approach to solve hand-crafted feature extraction process of machine learning which requires human intervention during feature extraction. The extent in which the classifier classified the coffee bean is not that much promising. Deep learning based approaches have performed exceptionally well in solving complex problems, hence the feature of each coffee bean image is relatively similar.

### **2.7.2. Related to convolutional neural network approach**

Atnafu Solomon [8], has conducted on skin lesion classification in dermoscopic images using deep neural networks. the overall objective of this study is recognition of carcinoma at an early stage in dermoscopy images so as to significantly increases the survival rate. during this study, a deep learning method is employed for carcinoma classification in dermoscopic images. The researcher tried to classify disease of the skin into three supervised classes namely: - basal cell carcinoma, melanocytic nevi, and melanoma. Before conducting experiment researcher applied data cleaning, data cropping, data augmentation, one hot encoding and normalization image pre-processing. Two frameworks are proposed: one is to coach a deep neural network model from scratch, and therefore the other is to transfer learning

a pre-trained network model. The proposed frameworks are evaluated on the most important publicly available dermoscopy image database, HAM10000. The primary framework achieves approximately 94.90% sensitivity, 97.19% specificity, and 96.51% accuracy and therefore the second framework achieves about 94.33% sensitivity, 96.87% specificity, and 96.51% accuracy. and therefore the two frameworks are compared with the prevailing carcinoma classification methods and the experimental results show that the proposed frameworks have higher classification accuracy than the opposite methods, which indicates that the frameworks proposed during this paper are effective and feasible. However, this study is limited for only skin lesion classification and dataset used to conduct this study is directly downloaded from the internet, these lead to reliability under questions.

Sampada Gulavnai and Rajashri Patil [63], proposed that image based mango plant disease detection using deep learning approach. They identified techniques of detecting mango disease are required to market better control to avoid this crisis. By considering this, paper describes image recognition which provides cost effective and scalable disease detection technology. Paper further describes new deep learning models which give a chance for straightforward deployment of this technology. The image dataset which is termed because the “original mango dataset” comprised 8,853 images taken to conduct experiment. The four classes include four sorts of diseases. The image count within the original dataset of every class: mango anthracnose disease (1952 images), mango mildew disease (1217 images), red rust (3479 images) and mango golmich (2205 images). the photographs were then transformed using data augmentation techniques through python coding into separate images to make the secondary dataset. Transfer learning technique is employed to coach a profound Convolutional Neural Network (CNN) to acknowledge 91% accuracy.

Vaibhav Amit Patel and Manjunath V. Joshi [64], they aimed to classify rice into five corresponding group by using convolutional neural network. The main gaps what they identified is manual classification of rice is neither practical nor economically feasible. In addition to this, traders adulterating a particular type of rice with poor quality type and the mix may include broken rice, stones, damaged seeds, etc. They proposed two methods to classify the rice types. In the first

method, we train a deep convolutional neural network (CNN) using the given segmented rice images. In the second method, we train a combination of a pretrained VGG16 network and the proposed method, while using transfer learning in which the weights of a pretrained network are used to achieve better accuracy. Our approach can also be used for classification of rice grain as broken or fine. We train a 5-class model for classifying rice types using 4000 training images and another 2-class model for the classification of broken and normal rice using 1600 training images.

The overall accuracy of the model with 5-class and 2-class (broken and normal) classification with transfer learning are 94.20% and 99.31% respectively. The limitation of this study is, the researcher only applied one pretrained model which is VGG16. Many better performing CNN architectures have appeared in 2014. GoogleNet that won 2014 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

*Table 2. 5 Summary of related work*

	<b>Author(s)</b>	<b>Title</b>	<b>Algorithm</b>	<b>Accuracy</b>
<b>1</b>	Abrham Debasu, Dagnachew Melesew and Seffi Gebeyehu	Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques.	ANN, KNN, Naïve and a hybrid of self-organizing map (SOM) and Radial basis function (RBF)	combination of RBF (Radial basis function) and SOM is 90.07%
<b>2</b>	Birhanu Hailu and Tesfa Tegegne	Ethiopian Roasted Coffee Classification Using Imaging Techniques.	Neural Net	98.2%
<b>3</b>	Asma Redi Baleker	Raw Quality Value Classification of Ethiopian Coffee Using Image Processing Techniques	The Naïve Bayes, C4.5 and Artificial neural networks (ANN).	The classification accuracy of Naïve Bayes is the highest from ANN with 82.72%
<b>4</b>	Hiwot Desta Alemayehu	Development of Automatic Sesame Grain Classification and Grading System Using Image Processing Techniques	delta E standard color difference algorithm.	88.2% and 93.3% for classification and grading respectively

5	Getahun Tigistu	Automatic Flower Disease Identification using Image Processing.	Artificial Neural Network	83.3%
6	Habtamu Minassie	Image Analysis for Ethiopian Coffee Classification.	Naïve Bayes and Neural Network classifiers Ethiopian	77.4% using neural network classifier
7	Atnafu Solomon	Automatic Skin Lesion Classification in Dermoscopic Images using Deep Neural Networks	CNN	96.51%
8	Sampada Gulavnai, Rajashri Patil	Deep Learning for Image Based Mango Leaf Disease Detection	CNN	91%

## **CHAPTER THREE**

### **SYSTEM DESIGN AND ARCHITECTURE**

#### **3.1. Overview**

Recently defining and classifying objects into the corresponding class using image processing techniques is applied in different areas. This recognition and classification of an object go through many processes in order to produce the objects. The classification of an object by the use of an item's attribute or features. In this chapter, we will discuss, the methodology of research design and the proposed system architecture of the classification of coffee beans based on their growing origin.

#### **3.2. Design Science Research Methodology**

DS is important in a discipline focused on the creation of effective objects. Several IS scientists have pioneered design science research in information systems, but there has been little DS research done within the discipline in the last 15 years. The lack of a methodology to serve as a widely agreed-upon DS research framework, as well as a blueprint for its presentation, may have slowed its adoption. The Design Science Research Methodology (DSRM) presented here integrates the concepts, practices, and procedures required for such research and achieves three goals: It is compatible with previous literature, it provides a nominal DS research process model, and it provides a mental model for the presentation and evaluation of DS research in IS [14].

In this study, different literature and related works on agricultural products, primarily coffee and sesame, were reviewed in order to obtain concepts, theories, methods, and to understand the difference between what is expected in the domain and what is actually happening [35]. According to Ken Peffers [14], the DS process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication.



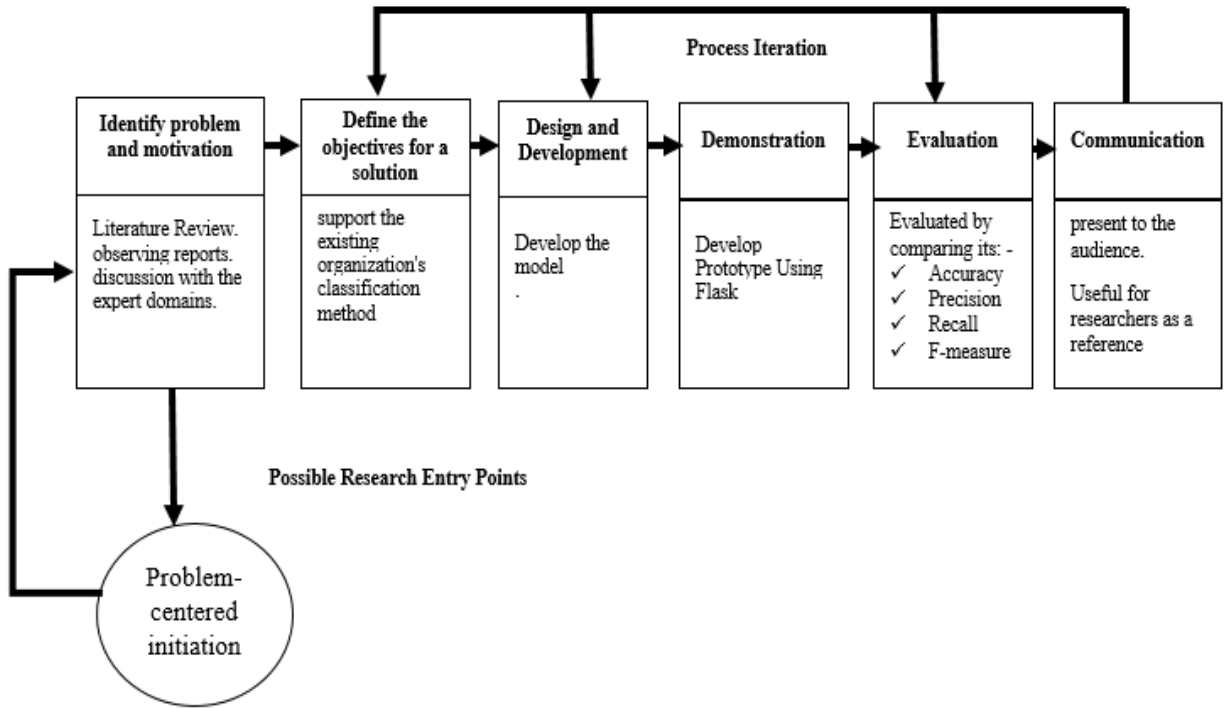


Figure 3. 1 Design Science Research Process Model Adopted From [63]

**Problem identification and motivation:** The research used a variety of ways to find out about the problem domain, by reviewing previous literature and observing reports from organization, through discussion with the expert domains, and other sources. This enabled the research to identify and understand the current problem. In this study, our country Ethiopia produces a large amount of coffee from various regions and exports it to various countries around the world. The classification and quality control mechanism, on the other hand, is still manual. As a result of this and other issues, Ethiopia is the world's fifth largest producer of coffee. Because of the scarcity and poor quality of coffee, Ethiopia did not gain the expected and planned foreign currency. As stated in Chapter One, a lack of automated technology causes a variety of problems.

**Define the objectives for a solution:** The method that the organization is currently using to classify coffee beans into corresponding regions is time consuming and inefficient. We have begun to develop an image-based classification model to classify green coffee beans based on their growing regions, which will support the existing organization's classification method and allow users to run cost-effective classification and reduce process time.

**Design and development:** in this study, we designed detailed architecture of the proposed system and convolutional neural network algorithm. Based on the architecture, we developed the model that can classify the coffee bean into corresponding regions. To do this, the most popular and easy

programming language is used to write code which is called python 3.7.11 version. Anaconda distribution is installed which have different types deep learning libraries like Keras 2.3 and tensorflow 2.0 for processing image in python. Integrated development environment is necessary to edit the source code. Spyder IDE is used to edit the coding part which is embedded in anaconda distribution. Finally, we developed the coffee bean classification model that reduce the processing time of the organization.

**Demonstration:** We will develop graphical user interface to make the communication between end users of the organization and developed model or artifacts using flask module.

**Evaluation:** The evaluation focuses on evaluating the results and determining how well the artifact solves the problem. At this point, the model is evaluated by comparing its accuracy in terms of confusion matrices such as precision, recall, and f1-score. Our developed artifact was evaluated in terms of functionality through providing the developed artifact to the employees of EQAIC. As a result, the system performance testing is carried out by importing various beans of coffee into the developed prototype, in order to test how effectively the designed coffee bean classification prototype saves time. On the other hand, user acceptance testing was carried out by preparing questions for users related to functionality and non-functionality requirements.

**Communication:** In this study, we will submit a document to Debre Berhan University's department of information systems and then present to the audience. Our study, as a research, provides various contributions for researchers who are motivated to conduct research in this area using deep learning to use as a reference.

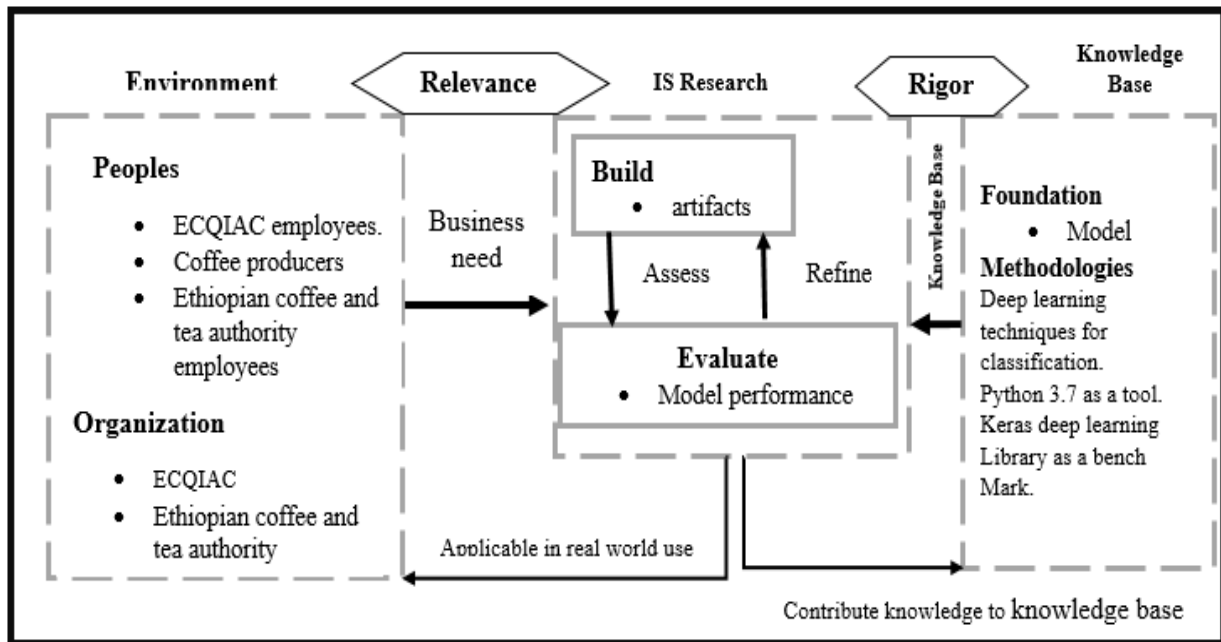


Figure 3. 2 Design Science Research Framework

### 3.3. The Proposed System Architecture

In this section, generally we have drawn the architecture of proposed coffee bean classification based on their growing region as well as provide detail description about each component included in the architecture as shown in figure 3.3 below.

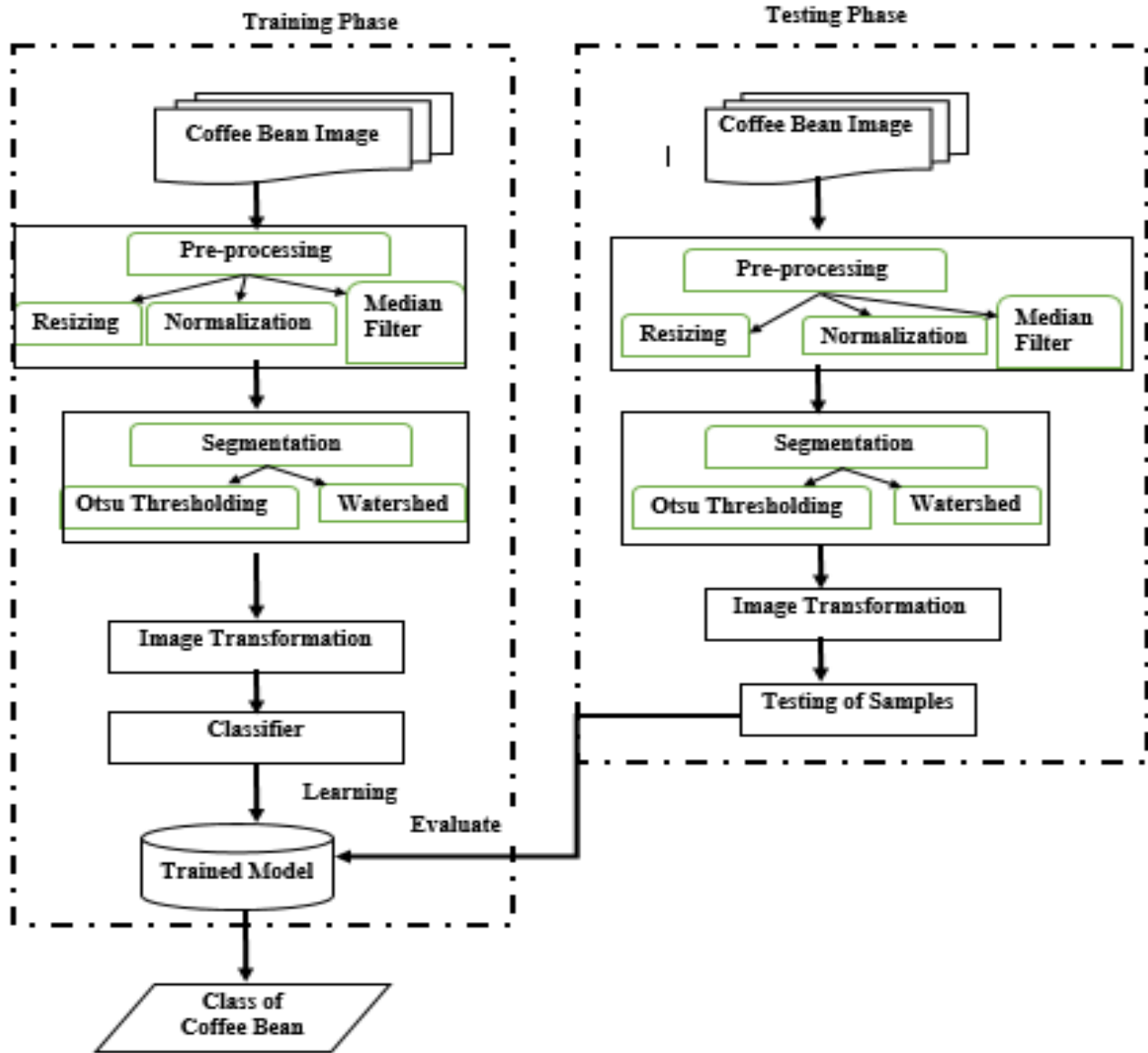


Figure 3. 3 Proposed System Process

After the important data was collected the next step is image pre-processing techniques were applied to the acquired images to extract useful features that are necessary for further analysis. After this, several analytical discriminating techniques are used to classify the images according to the specific problem at hand. In above figure 3.3 depicts the basic procedure of the proposed vision-based classification of coffee bean. Thus, the classification of coffee bean starts with acquiring images of coffee bean using a digital camera. In order to remove noises that occur during the image acquisition step, we have applied an image enhancement technique. Then, features that are best suited to represent the image are extracted from the image using an image analysis

technique. Based on the extracted features the training and testing data that are used to identify are extracted in order to classify coffee bean. Finally, appropriate deep learning identifier is selected to classify an image in to its origin of coffee bean [26].

### 3.3.1. Image Acquisition

Image processing technique starts with acquiring image data to train classifier model. The inputs of classifier model were digital images of green coffee beans. In this study, the image data samples were taken from Ethiopian Coffee Quality Inspection and Auction Center which are harvested in 2020 and from each class of coffee bean 250g samples were taken. Green coffee beans were placed on white paper before the photo-graphics were taken. The mobile phone Samsung camera was set in automatic mode with F/16, exposure time 1/60 s, ISO 200, exposure compensation 1.3, auto focus and the camera is put at 15cm above the surface of the beans.

The both front and back sides of the pictures of the beans were taken. Image preprocessing and image segmentation techniques were applied to isolate each beans. The size of each image was set to  $224 \times 224$  pixels. The images were manually labeled to the Gujji, Jimma, Kaffa, Nekempti, Sidamo and Yirgacheffe with the help of domain experts. All images of the samples were divided into two groups, training data and test data. The training data was used for the learning of the neural networks. The validation data was used to confirm the transition of classification accuracy in the learning phase of neural networks.

Table 3. 1 Total number of Image dataset

Classes of coffee bean	Numbers data
Washed Gujji	520
Unwashed Jimma	520
Washed Kaffa	520
Unwashed Nekempti	520
Washed Sidamo	520
Washed Yirgacheffe	520
<b>Total</b>	<b>3120</b>

### 3.3.2. Image Preprocessing

Data preprocessing is a vital step to deep learning to convert the raw data into clean dataset and it is helpful to improve model's ability to learn [28]. The main goal of preprocessing is to identify the important feature in an image and discard all other information other than the required

information [65]. Tasks which have been done in preprocessing are image resizing, converting image from RGB color to Grayscale, normalizing image, image enhancement, encoding label and one hot encoding vector.

### 3.3.2.1. Image Resizing

Image resizing is a key technique for displaying images on different devices, and it has attracted much attention in the past few years. Image resizing defines preserving an important region of an image, minimizing distortions, and improving efficiency [66].

<b>Input: Image</b>
<b>BEGIN:</b> Read all image from dataset Resize image using OpenCV module and resize method to specified width and height. <b>END</b> Return resized image
<b>Output: Resized Image</b>

Figure 3. 4 Image resizing Algorithm

Original Image



After Resizing



Figure 3. 5 Resized Image

It is necessary to resize the coffee bean images to delete the irrelevant regions as well as to reduce the computational time and space for the deep learning network [28].

### 3.3.2.2. Noise Removing

The exactness of classification, grading and sorting of agricultural products models mainly depends on the preprocessing process. The raw data is subjected to several preliminary processing

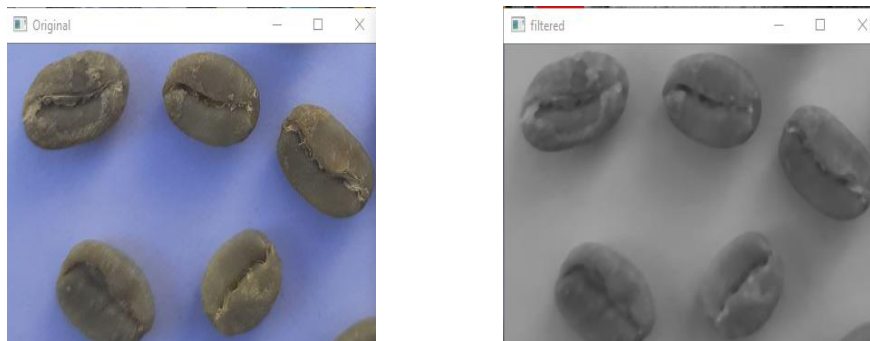
steps to make it functional in the descriptive stages of coffee classification. First, the images were converted into gray scale image. The noise occurred during image acquisition is removed using median filtering. Image smoothing techniques help in reducing the noise. In OpenCV, image smoothing (also called blurring) could be done in many ways. The collected and prepared images for training the network in this study were of poor quality (not free from noise) due to a variety of factors including the device (camera resolution) and uncontrolled environments or temperature effects. As a result, to remove salt and pepper noise from images, the median filtering technique was used. For the implementation of this preprocessing technique, the OpenCV programming function library, which is primarily aimed at real-time computer vision tasks, was used because it supports multiple programming languages, including Python, and multiple platforms [3].

```

Input: Noisy images
BEGIN:
For every pixel in an image
do
    Sort the value in the mask
    Pick the middle one in the sorted one
    Replace the pixel value with a median one
END
Output: Filtered Image

```

Figure 3. 6 Algorithm for median filtering



a) Before Median Filter

b) After Median Filter was applied

Figure 3. 7 Enhanced Imaged Through Median Filtering

### 3.3.2.3. Label Encoding

The most common approach to converting categorical features to a suitable format for use as input to a deep learning model [67]. In this study, the required datasets contain six class labels and these were in the form of strings. Therefore, these string values were changed into numerical value to make it the machine-readable.

Table 3. 2 Label Encoding

Coffee bean class labels (Text)	Coffee bean class labels (Numeric)
Gujji	0
Jimma	1
Kaffa	2
Nekempti	3
Sidamo	4
Yirgacheffe	5

### 3.3.2.4. Binary Encoding

Categorical data can be represented in a binary format by first assigning a numerical value to each category and then converting it to its binary representation [67]. The numbers were replaced by 1s and 0s, depending on which column has which value. In this study, the dataset has six classes, namely, Gujji, Jimma, Kaffa, Nekempti, Sidamo and Yirgacheffe. Therefore, the data that belongs to a particular class has a value of one and the remaining have zero.

Table 3. 3 One-Hot Encoding

Gujji	Jimma	Kaffa	Nekempti	Sidamo	Yirgacheffe
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

### 3.3.3. Image Segmentation

Image segmentation is an essential component in many visual understanding systems. It involves partitioning images (or video frames) into multiple segments or objects. Image segmentation can be formulated as a classification problem of pixels with semantic labels (semantic segmentation) or partitioning of individual objects (instance segmentation). Numerous image segmentation algorithms have been developed in the literature, from the earliest methods, such as thresholding, histogram-based bundling, region growing, k-means clustering, and watersheds. In this study, we applied watershed segmentation algorithm and Otsu binary thresholding. Watershed is the ridge that divides areas drained by different river systems [68].



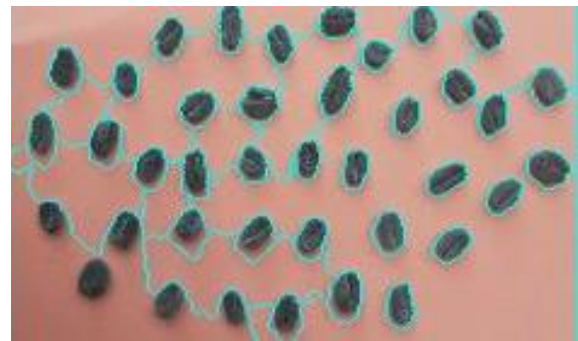
The Watershed Transform is a unique technique for segmenting digital images that uses a type of region growing method based on an image gradient. The concept of Watershed Transform is based on visualizing an image in three dimensions: two spatial coordinates versus gray levels.

The Watershed Transform effectively combines elements from both the discontinuity and similarity based methods. Since its original development with grey-scale images, the Watershed Transform has been extended to a computationally efficient form (using FIFO queues) and applied to colour images. In addition to this, it requires low computation time, fast, simple and intuitive method and able to produce a complete division of the image in separated regions [69].

Original Image



After Segmentation



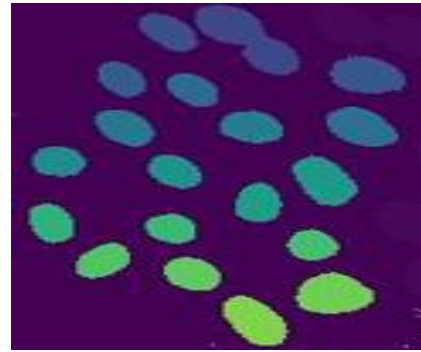
*Figure 3. 8 Image after watershed segmentation was applied*

As we have seen clearly figure 3.9, Based on the threshold pixel values of an image within an object of interest is set to one and the remaining (the background) is set to a pixel value of 0. The pixel value of 1 indicates white (object of interest) and a pixel value of 0 indicates black (background). Finally, we have obtained each object of interest in the image has been isolated from the background and labeled in-order to ease image analysis from the binarized image.

Original Image



After Otsu segmentation



*Figure 3. 9 Coffee bean Image after Otsu applied*

### **3.3.4. Data augmentation**

A general way to reduce overfitting is to introduce more training data. In many cases, however, it is both practical and beneficial to systematically alter the existing training data to generate more examples while preserving the label. In the case of image classification, the input images can be transformed in the following ways: horizontal and vertical flipping, cropping, scaling, translating, rotating, color jittering or shifting, and contrast and brightness changes [70].

In this study, we have used rotation range=40, width\_shift\_range=0.2, height\_shift\_range=0.2, shear range=0.2, zoom range=0.2, horizontal\_flip augmentation techniques. Different images are generated from the original image by using the above techniques. During transformation, images are not stored in a disk and don't require memory for storage because transformed images are generated at run-time. These augmentation schematics are effective in computation and memory custom. Based on this technique, we will overcome the overfitting problem and also increase testing performance.

### **3.3.5. Normalization**

Normalizing is another preprocessing technique we use before further process. For this study, normalization has used to ensure that each input parameter (in this case pixel of image) has a similar data distribution or changes the range of pixel intensity values. The main reason why this preprocessing technique has applied in this study it makes the networks very faster and easier while it trains the network when the data is not normalized, the shared weights of the network have different calibrations for different features, which can make the cost function to converge very slowly and ineffectively. Image pixel values are integer between the ranges of 0 to 255. Although

these pixel values can be presented directly to the model, it can result slower training time and overflow. Overflow is what happens when numbers get too big and the machine fails to compute correctly. So, we normalize our data values down to a decimal between 0 and 1 by dividing the pixel values with 255 [71].

```
Input: Resized Image
BEGIN:
    Import all resized image from stored place
    Then each floating point intensity value of an image is divided by
    255
Return Image
END
Output: Normalized Image
```

Figure 3. 10 Image normalization algorithm

In the above figure 3.10, the all resized images were used as the input iteratively and the individual image was changed into float and divided by 255 in order to range the pixel value of images between 0 and 1.

### **3.3.6. Convolutional Neural Network Model Architecture**

The architecture of the model composes layers which are responsible for feature extraction and classifying the input image into one of the categories. In this study, (CNN) architectures were created by two methods: such as: - Training the model from the scratch and Transfer learning

#### **3.3.6.1. Training the model from the scratch**

The CNN architectures created in this study are composed of the number of convolutional layers, pooling layers, fully connected layers, and dropout layers. Convolutional layers apply a convolution operation to the input image and the layer performs mathematical operations for each sub-region, to produce a single value in the output of the feature map. Pooling layers perform down sampling operation on the size of the feature maps by using some function to summarize sub-regions. Max pooling is used to construct the CNN architectures that extract sub-regions of the feature map which is done by applying a max filter to keeps their maximum value and disregard all other values. Fully-connected layers utilize the features extracted by convolutional layers and perform the classification. The purpose of applying dropout layer is to reduce overfitting and generalization error [28].

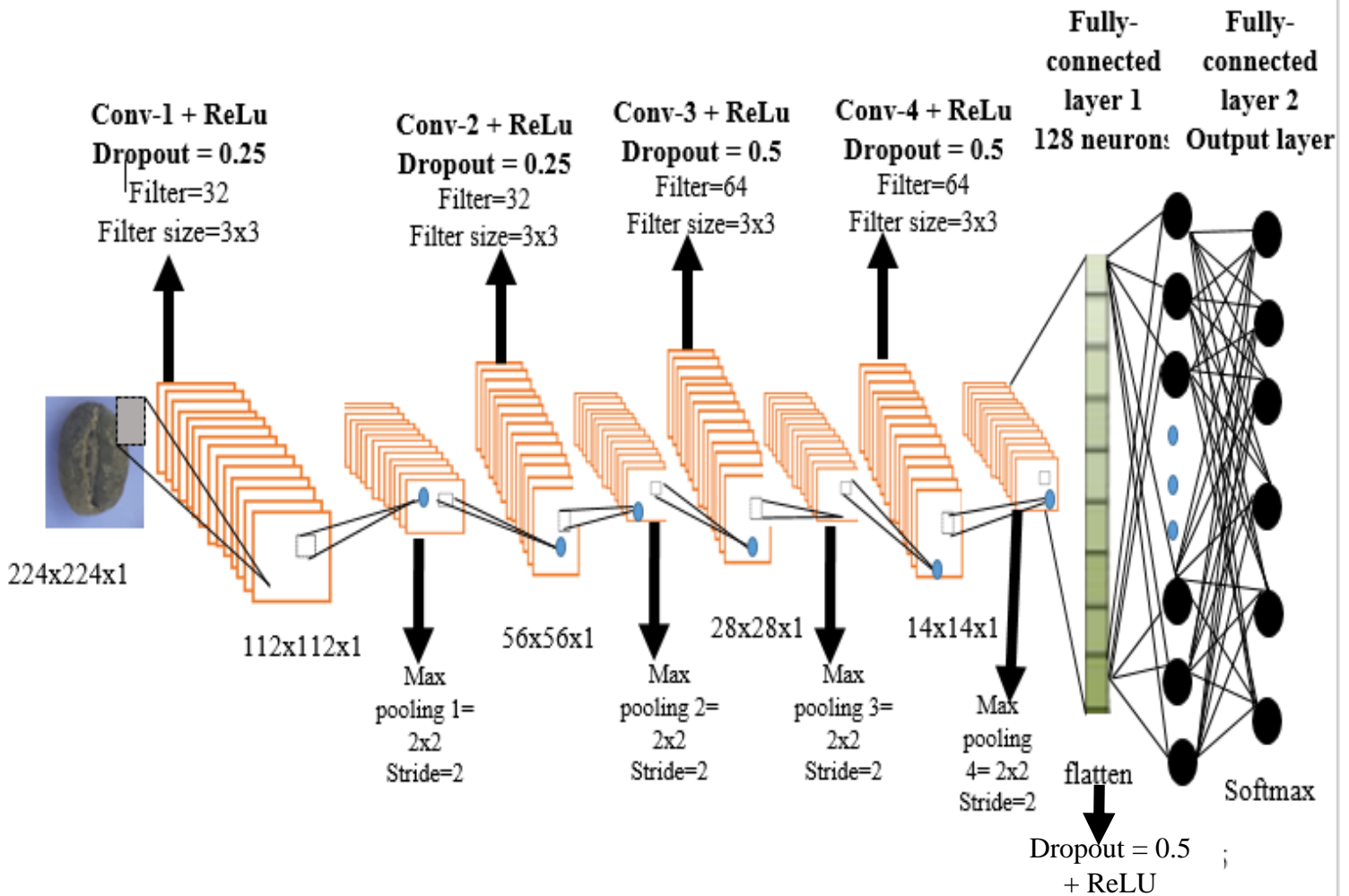


Figure 3. 11 Proposed CNN classification model

As shown in figure 3.11, the input layer consists of 224 by 224 pixel images which mean that the network contains 50,176 neurons as input data and the input pixels are grayscale. Here, this model of CNN has nine hidden layers. The first hidden layer is the convolution layer 1 which is responsible for feature extraction from an input data. This layer performs convolution operation to small localized areas by convolving a filter with the previous layer. In addition, it consists of multiple feature maps with learnable kernels and rectified linear units (ReLU). The kernel size determines the locality of the filters. ReLU is used as an activation function at the end of each convolution layer as well as a fully connected layer to enhance the performance of the model. The next hidden layer is the pooling layer 1. It reduces the output information from the convolution layer and reduces the number of parameters and computational complexity of the model. The different types of pooling are max pooling, min pooling and average pooling. Here, max pooling

is used to subsample the dimension of each feature map. Third and fourth hidden layer is Convolution layer 2 and pooling layer 2 respectively, which has the same function as convolution layer 1 and pooling layer 1 and operates in the same way except for their feature maps and kernel size varies. Convolution layer 3 and pooling layer 3 which has the same function as convolution layer 2 and pooling layer 2 and operates in the same way except for their number of filter and dropout value. Flatten layer is used after the pooling layer which converts the 2D featured map matrix to a 1D feature vector and allows the output to get handled by the fully connected layers. A fully connected layer is another hidden layer also known as the dense layer. It is similar to the hidden layer of Artificial Neural Networks (ANNs) but here it is fully connected and connects every neuron from the previous layer to the next layer. In order to reduce overfitting, dropout regularization method is used at fully each convolution layer and fully connected layer 1. It randomly switches off some neurons during training to improve the performance of the network by making it more robust. This causes the network to become capable of better generalization and less compelling to overfit the training data. The final fully connected layer compute the class scores, resulting in volume of size  $1 \times 1 \times 6$ , where the 6 numbers correspond to a class score, among the 6 categories coffee bean. Since the output layer uses an activation function such as softmax, which is used to enhance the performance of the model, classifies the output coffee bean based on their growing region such as Gujji, Jimma, Kaffa, Nekempti, Sidama and Yirgacheffe which has the highest activation value as shown in figure 3.11.

### **3.3.6.2. Training Model in pre-trained model**

As we discussed in chapter 2 section 2.9 different researchers designed different CNN architectures, which called transfer learning. It is the way to retrain the CNN networks which is previously trained on more than a million images and can classify images into 1000 object categories then to classify new images. It often used with CNN to keep all layers of pre-trained model except the last one, which is trained for the specific problem. This method can be particularly suitable for faster epoch processing times since the layers are frozen and loaded from a previously trained network as shown in figure 3.12. For this study, VGG16 and VGG19 architectures which is pre-trained on a larger dataset were considered in this study.

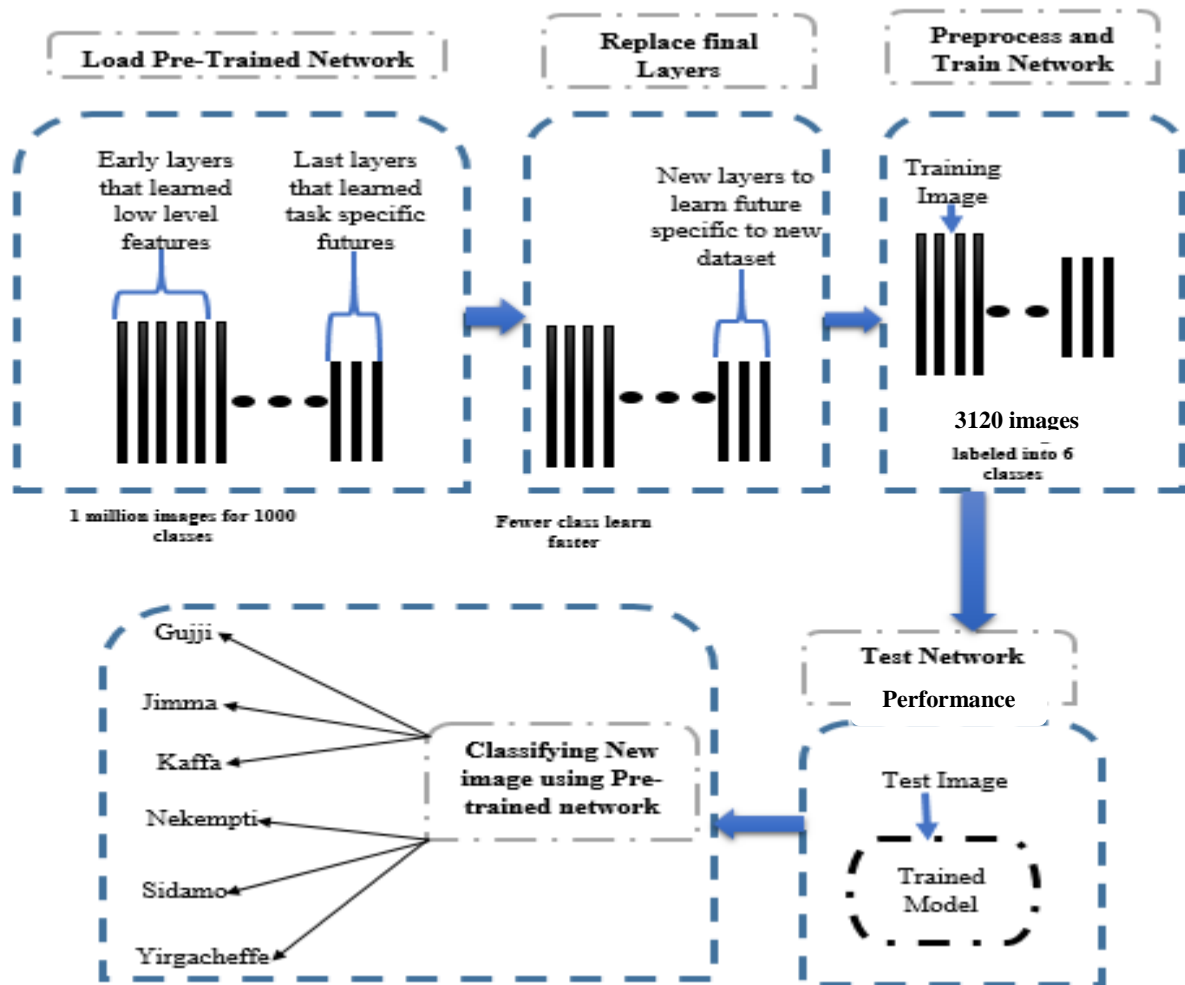


Figure 3. 12 Customized pretrained model

### 3.3.7. Hyperparameters

Hyperparameters and their selection are very important concepts in machine learning, especially in the context of neural networks since these types of models employ a variety of them. Intuitively, Hyperparameters can be seen as handles that have to be tuned independently of the model parameters, and are typically chosen before the learning process begins. In this study, we will mainly discuss optimization-specific hyperparameters, layer-specific hyperparameters, and regularization-specific hyperparameters [70].

Table 3. 4 Hyperparameters Description

Hyperparameters	Values	Description
Filter	32,32, 64, 64	Those four are convolution layer hyperparameters and among from those strides and filter size are pooling layer hyperparameters which can be applied before training process begin.
Kernel size	3x3	
Depth	1	
Padding	Valid	
Stride	1	
Learning rate	0.0001	These are applied during the training process, i.e. They are hyperparameters for the fully connected layer
Learnable parameters		
Optimizer	Adam	
Batch size	64	
Epoch	50	
Loss function	Categorical cross-entropy	

### 3.3.8. Classification

After learnable feature is extracted, the next step is identification and classification of coffee bean based on their origin using softmax classifier in to six classes. SoftMax is actually a kind of normalizing function because it deals with the probability of a sample coming from a particular class. The sum of all the node outputs coming out from SoftMax classification equal to one. since add 0.166 that is the probability of each six class ( $0.166+0.166+0.166+0.166+0.166+0.166=1$ ) [25].

### 3.4. Model Evaluation Metrics

The performance evaluation of machine learning algorithms is assessed based on predictive accuracy, which is often inappropriate in case of imbalanced data and error costs vary remarkably. Machine learning performance evaluations involve certain level of trade-off between true positive and true negative rate, and between recall and precision [72]. A confusion matrix is a technique for the summary of the prediction result or the performance of a classification algorithm [35]. In this study, the efficiency of CNN model is determined using measures such as precision, recall and F1-score.

Table 3. 5 Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

**True positive (TP):** - represents the number of images that are correctly classified as positive by the developed classifier model.

**False-positive (FP):** -represents the number of images that are classified as positive in predicted class but they are actually negative.

**False-negative (FN):** - represents the number of images that are classified as negative in predicted class but they are actually positive.

**True-negative (TN):** - represents the number of images that are correctly classified as negative in both actual and predicted classes.

Accuracy compares how close a new test value is to a value predicted by if ... then rules.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots \dots \dots Equation 3. 1$$

Sensitivity measures the ability of a test to be positive when the condition is actually present. It is also known as True Positive rate [72].

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \dots \dots \dots Equation 3. 2$$

Specificity measures the ability of a test to be negative when the condition is actually not present. It is also known as True Negative rate [72].

$$Precision = \frac{TP}{TP+FP} \dots \dots \dots Equation 3. 3$$

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \dots \dots \dots Equation 3. 4$$



# CHAPTER FOUR

## EXPERIMENT AND EVALUATION

### 4.1. Introduction

This chapter describes the dataset used in this study, experimental design, performance evaluation matrices, train and evaluate our model, final train and evaluate state of the art CNN architecture and developing web application.

### 4.2. Description and Preparation of Dataset

The steps involved in preparing data for the classification model can be stated as follows: collect data, preprocess data, segment image, and transform data. To prepare the dataset required, we use an iterative technique with multiple loops. The first step is concerned with collecting available data needed to solve the problem. Image data are collected from the Ethiopian coffee quality inspection and auction center, which is found in Addis Ababa. From the stated organization we got 250g coffee bean sample per class and we prepared the necessary things to capture images like environment setup and adjusting the camera. The image was captured by using Samsung mobile phone SM-M205F camera with the following parameter.

- ✓ The distance between the images and the camera was to 15cm
- ✓ For all images, the pictures are taken before sunrise and after the sunset.
- ✓ All the images are in JPEG (Joint Photographer Expert Group) file format
- ✓ All the captured images were taken in the same controlled environment in-order to avoid external effects of sunlight and other environmental conditions,
- ✓ All images have a size of 3096x4128 pixels width and height respectively.
- ✓ The background of the image is white paper
- ✓ The storage size of a single image is 3 MB as well as focal length is 3.58mm.

Secondly, is about getting the collected data into a form that can be easy to work. We follow resizing, enhancement, and normalizing steps to preprocess the data.

Thirdly, we transform the data collected. This step is related to making the dataset suitable for the algorithm [71].

### **4.3. Experimental Design**

After preparing the dataset in best way to feed for the network, the next step is designing the model. In this study, we have used the tensorflow framework and Keras API Deep learning library as a benchmark because Keras provided an already pre-trained model, trained over ImageNet dataset.

In the process of model building, the total image used to train the model is partitioned into different test options for training, validation and testing respectively, and, total of 3120 image used for six class. Accordingly, two distinct convolutional neural network methods were applied to develop classification models. The first cases, training from scratch (i.e. setting different hyperparameters into the network from the scratch as discussed in chapter three hyperparameters section 3.3.6.1). Secondly, transfer learning technique (i.e. through loading VGG16 and VGG19 pre-trained model), then fine-tuning theses networks, finally build a new build. In this study, classification of a coffee bean image based on their growing region has developed using one of the deep learning-based approaches with convolutional neural networks.

### **4.4. Simulation Environment**

In this research, we tried to classify the coffee bean through on their growing region have been also identified by the end user of the ECQIAC (Ethiopian coffee quality inspection and auction center) organization's employees. In order to do this, creating the means of communication between those end users and the developed model is the vital thing which is called user interface. User interface is the means of communication with the end-user that enable them to interact easily with the developed coffee bean classification system. To do this, the end-user of the system have to open any kind of browser which is installed on their computer device. Now start the web application by writing the URL of the coffee bean classification system on web browser, after that, the end-user fills the required information for the developed classification system, and the web browser sends HTTP requests to the webserver and then the webserver gives back an HTTP response to the browser.

The proposed model is implemented with Python 3.7.0 based on deep learning framework for python called Keras [73] with Tensorflow backend and HP with Intel Core i7-7020@3.20GHz processor with 8 GB of RAM, and 64-bit operating system type . In addition to Keras [73], and Tensorflow, Flask which is a python framework for web application development, HTML, CSS

and JavaScript were used as backend(Flask) and frontend (HTM, CSS and JavaScript) to production.

## 4.5. Experimental Results

### 4.5.1. Experiments on Training from the scratch

In this study, we were conducted three experiments.

#### 4.5.1.1. Experiment 1: Without segmentation but with Data augmentation.

In this experiment, coffee image of size 224x224x1 with associated class label were fed in to the CNN architecture with 32, 32, 64 and 64 convolutional filters, there is one pooling layer and one dropout layer of 0.25 respectively after each convolutional layer, except the first CNN layer with the dropout value of 0.25. Flatten layer, fully-connected layers, one dropout layer of 0.25 and again fully-connected layers with Softmax activation function as we mentioned in CNN architecture. As shown in Figure 4.1 and figure 4.2, a model training history of the experiment. Before proceeding to the training the network the dataset was partitioned in three class namely, training, validation and test set with the percentage value of 80% for training and 20% for testing the trained model. Training data was also divided into two which is 70% for training and the remaining for validation. During the training of this network the highest Training accuracy is **79.09%** and highest validation accuracy of **78.80%** were recorded at epoch 50. Finally, after training the network test accuracy is **76.3%**. However, the CNN performance graph indicates there is model overfitting in this experiment even after we applied different regularization mechanisms. Also we tried conducting experiment through changing different hyperparameters values and dropout values, but we didn't get better accuracy.

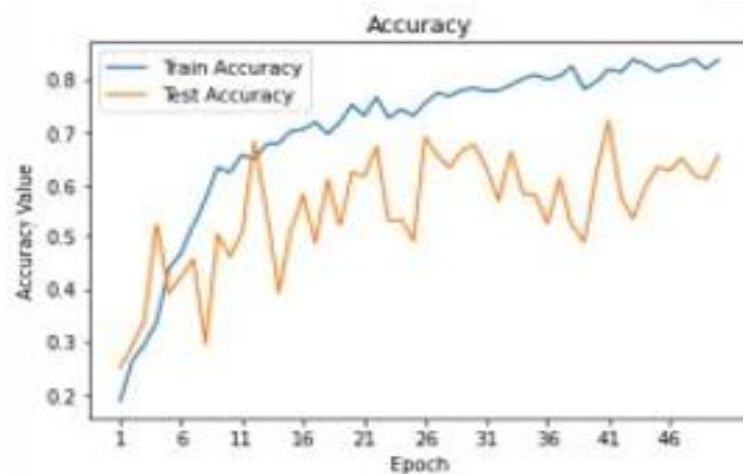


Figure 4. 1 Graph of training and Test accuracy for experiment one

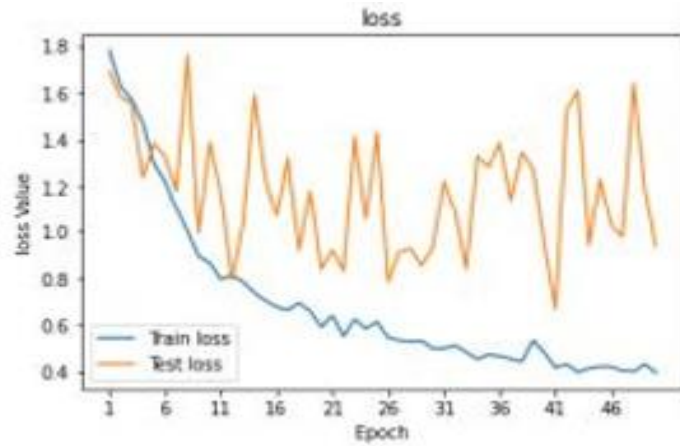


Figure 4. 2 Training and Test Loss for experiment one

Table 4. 1 Confusion matrix of experiment one

		Predicted Label					
		Types of coffee	Gujji	Jimma	Kaffa	Nekempti	Sidamo
Actual Label	Gujji	<b>8</b>	0	0	0	16	10
	Jimma	1	<b>45</b>	0	0	0	0
	Kaffa	0	0	<b>38</b>	2	0	0
	Nekempti	0	0	0	<b>53</b>	0	0
	Sidamo	0	0	0	0	<b>10</b>	20
	Yirgacheffe	0	0	4	0	0	<b>43</b>

Table 4. 2 Performance result of experiment one

Performance Metrics			
Types of coffee	Precision	Recall	F1-Score
Gujji	89%	24%	38%
Jimma	100%	98%	99%
Kaffa	90%	95%	92%
Nekempti	96%	100%	98%
Sidamo	38%	33%	35%
Yirgacheffe	59%	91%	72%
Weighted Average	79%	74%	72%

#### 4.5.1.2. Experiment 2: After applying Watershed Segmentation Algorithm

The proposed watershed segmentation algorithm used to separate region of interest from the background and removed unnecessary part of the coffee image in order to simplify learn features from the input image and computationally free to analysis image information. Before proceeding

to the training the network the dataset was partitioned in two class namely, training and test set with the percentage value of 70% for training and 30% for testing the trained model. In this case, as you see the percentage split is different from experiment one, but we also applied different percentage split 80% by 20% and 75% by 25% for training and test respectively but, we have gotten the high accuracy with the percentage split of 70% for training and the rest for test set.

We can see the training and validation accuracy and loss progress of the model after applied image segmentation in figure 4.3. it's gotten 99% training accuracy and validation accuracy is 98%. After training and validation, the model was evaluated using 936 images and it's gotten 97.8% test accuracy. Since the dataset given to the model after watershed segmentation minimizing feature learning process by removing unwanted part from the images.

Table 4. 3 Confusion matrix of experiment two

		Predicted Label					
		Types of coffee	Gujji	Jimma	Kaffa	Nekempti	Sidamo
Actual Label	Gujji	<b>157</b>	0	0	0	1	0
	Jimma	0	<b>169</b>	0	5	0	0
	Kaffa	1	2	<b>159</b>	0	0	0
	Nekempti	0	0	0	<b>141</b>	0	0
	Sidamo	9	0	0	0	<b>140</b>	0
	Yirgacheffe	0	1	0	1	0	<b>150</b>

Table 4. 4 Performance results of experiment two

Performance Metrics			
Types of coffee	Precision	Recall	F1-Score
Gujji	94%	99%	97%
Jimma	98%	97%	98%
Kaffa	100%	98%	99%
Nekempti	96%	100%	98%
Sidamo	99%	94%	97%
Yirgacheffe	100%	99%	99%
Weighted Average	98%	98%	98%

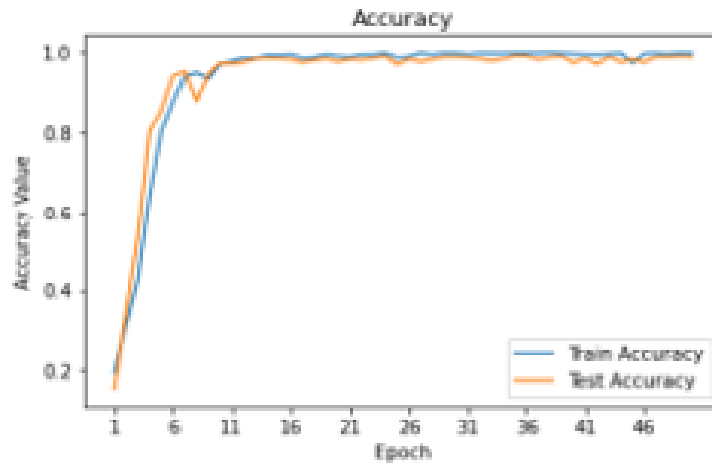


Figure 4. 3 Graph of training and Test accuracy for experiment two

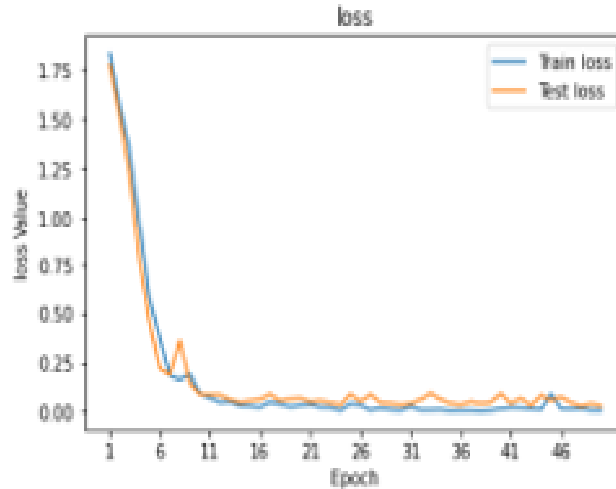


Figure 4. 4 Graph of training and Test Loss for experiment two

We can see figure 4.3 the training progress show increase in the training accuracy and simultaneously decrease in the loss as the number of epochs increases. During the training and test, the loss is the summation of error for each sample in the training and test sets. The lower the loss, the better the model and recognition result. As we can see the result in figure 4.3, the classification accuracy of training and test is better than experiment one. Training and test loss are also decreased from epoch to epoch. Therefore, we can say that our model generalization capability became much better since Training accuracy is better than test accuracy and test loss is greater than training loss. In this experiment there is no model overfitting.

### 4.5.1.3. Experiment 3: After applying Otsu Thresholding Segmentation Algorithm

Thresholding is a significant part of image segmentation to make binary images. Binary image analysis is useful for image feature extraction and it shortens the computation of geometrical features of an image. Hence, for this research work, we have used thresholding-based segmentation as it is simple and computationally inexpensive [25]. During conducting the experiment, from total dataset 70% was used for training the model and the remaining used to test the trained model.

As we shown in figure 4.5, both accuracy and loss indicates the model have some overfitting in which the training accuracy is less than test accuracy and test loss is less than training loss starting from epoch 1 up to 19. But, after the epoch number 20 the performance of the model after application of thresholding-based image segmentation, the training and test accuracy are increased parallel. The highest training accuracy is obtained at the epoch 44 and 47 which is 98.4% and test loss is 0.03 at the epoch value 44. And also testing accuracy 97.1% and test loss is 0.17.

Table 4. 5 Confusion matrix of experiment three

		Predicted Label					
		Types of coffee	Gujji	Jimma	Kaffa	Nekempti	Sidamo
Actual Label	Gujji	<b>153</b>	1	0	0	4	0
	Jimma	2	<b>172</b>	0	0	0	0
	Kaffa	0	0	<b>162</b>	0	0	0
	Nekempti	0	1	0	<b>140</b>	0	0
	Sidamo	5	3	1	1	<b>139</b>	0
	Yirgacheffe	4	1	2	0	2	<b>143</b>

Table 4.6 Performance result of experiment three

Performance Metrics			
Types of coffee	Precision	Recall	F1-Score
Gujji	93%	97%	95%
Jimma	97%	99%	98%
Kaffa	98%	100%	99%
Nekempti	99%	99%	99%
Sidamo	96%	93%	95%
Yirgacheffe	100%	94%	97%
Weighted Average	97%	97%	97%

```

Epoch 47/50
2184/2184 [=====] - 198s 91ms/step - loss: 0.0399 - accuracy: 0.9840 - val_loss: 0.1513 -
val_accuracy: 0.9765

Epoch 00047: val_accuracy did not improve from 0.97650
Epoch 48/50
2184/2184 [=====] - 204s 93ms/step - loss: 0.0687 - accuracy: 0.9739 - val_loss: 0.1960 -
val_accuracy: 0.9605

Epoch 00048: val_accuracy did not improve from 0.97650
Epoch 49/50
2184/2184 [=====] - 197s 90ms/step - loss: 0.0637 - accuracy: 0.9730 - val_loss: 0.1846 -
val_accuracy: 0.9669

Epoch 00049: val_accuracy did not improve from 0.97650
Epoch 50/50
2184/2184 [=====] - 199s 91ms/step - loss: 0.0518 - accuracy: 0.9780 - val_loss: 0.1706 -
val_accuracy: 0.9712

Epoch 00050: val_accuracy did not improve from 0.97650
936/936 [=====] - 12s 12ms/step
Test: accuracy = 0.971154 ; loss = 0.170586

```

Figure 4. 5 Trained Model Sample Screenshot of experiment three

Also we conducted the other experiment by using 80% split for training the model. The result is actually the same, but this one shown some increment in model training accuracy as shown below table 4.6. During this percentage split the number of dataset used for training the model is 2496 and the remaining for test. As we shown in the table 4.6 below, the performance of the model after application of thresholding-based image segmentation, the training and validation accuracy are increased parallel. The highest training accuracy is obtained at the epoch 44 which is 98.64% and training loss is 0.03 at the epoch value 44. And also testing accuracy 97.5% and test loss is 0.14.

Table 4. 6 Confusion matrix for binary thresholding test size 0.2

Performance Metrics			
Types of coffee	Precision	Recall	F1-Score
Gujji	97%	98%	98%
Jimma	98%	99%	99%
Kaffa	95%	99%	97%
Nekempti	99%	97%	98%
Sidamo	98%	96%	97%
Yirgacheffe	99%	96%	98%
Weighted Average	98%	98%	98%



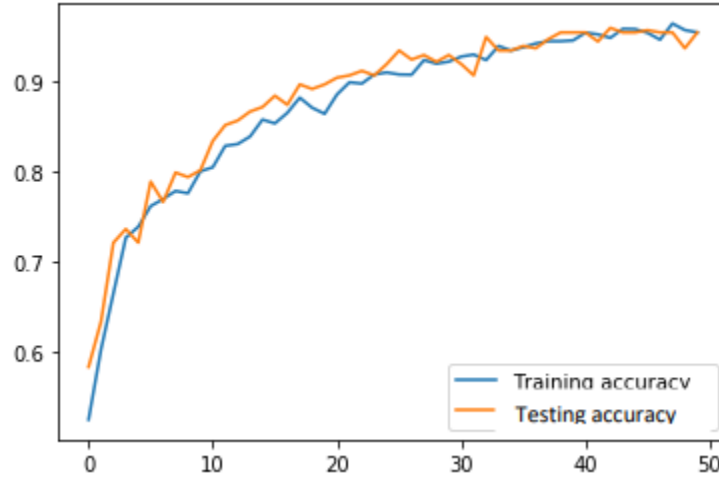


Figure 4. 6 Graph of training and Test accuracy for experiment three

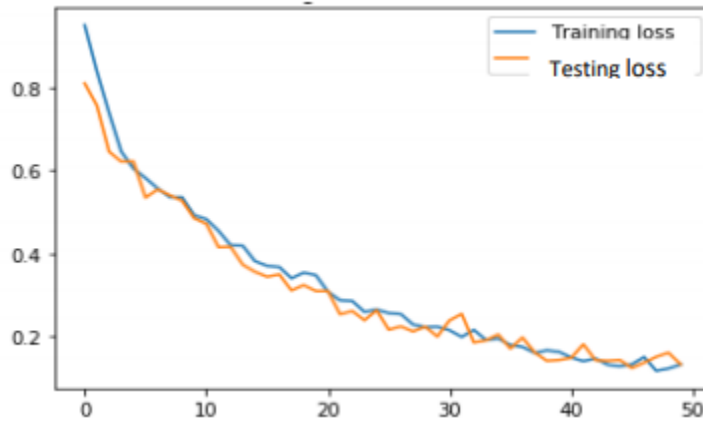


Figure 4. 7 Graph of training and Test loss for experiment three

#### 4.5.2. Training and testing state-of-the-art CNN architecture

Our goal compared or trained the state-of-the-art CNN architecture is to get the best classification performance and computationally free for coffee bean classification.

These methods were used for different situations through different researchers in order to build classifier model for the specified problem. But, these methods are also important for different problems through using their weights either customizing or changing only the last output layer.

Therefore, the researcher has utilized the weight of these pre-trained models and has better accuracy in image classification. Accordingly, VGG-19 and VGG-16 pre-trained models were adopted to recognize coffee bean varieties and classify as Gujji, Jimma, Kaffa, Nekempti, Sidamo, and Yirgacheffe. For our case, we chose the VGG19 and VGG16 models for some reasons. First,

even though it didn't win ILSVRC, it took the 2nd place showing nice performance. we only need 6 categories of images, so we thought VGG19 is enough for our dataset. Second, VGG19 architecture is very simple. If you understand the basic CNN model, you will instantly notice that VGG19 looks similar. Third, we have 8GB memory. This one is not the best choice, but we thought it would be enough to run VGG19 even though VGG19 is a big in size. Lastly, since a lot of people uses VGG16, the reason why initiated to apply VGG-16 [74].

#### 4.5.2.1. Training and Validation Accuracy of VGG19 (Visual Geometry Group)

As discussed in the above, the first selected pre-trained model was VGG-19. It has 19 layers including convolutional layers, pooling layer, and three fully connected layers with three blocks and two of them with 4096 neurons and 1000 neurons which is the output of the class probabilities. Besides, this state-of-art architecture, hyper-parameters, and parameters like the number of filters, filter size, stride, and padding were taken as it is with the default value to train the network. However, the last layers of this network were modified as per the number of the in this study. As discussed in the methodology part, the data augmentation technique also applied to the network to reduce overfitting problems. For this experiment, test options were tested during training the network by splitting the collected dataset into 80% of the images for training and 20% for testing percentage split test options. Finally, the performance of the developed model that has been developed using this test option was examined. So, we trained the model with 2496 sample images and 625 sample images for testing the trained model.

We can see figure 4.6 the highest training accuracy is obtained on epoch 47 which 87.34% and the test accuracy is 86%. Even after we applied image augmentation the graph shows us some validation accuracy and loss line fluctuates down and up. The interesting point in this experiment is training graph shows it increase from epoch to epoch without fluctuation. At the end the trained model was tested by using 624 sample image and we got 86%.

Table 4. 7 confusion matrix of VGG19

		Predicted Label					
		Types of coffee	Gujji	Jimma	Kaffa	Nekempti	Sidamo
Actual Label	Gujji	<b>71</b>	0	0	1	28	0
	Jimma	0	<b>116</b>	0	0	0	0
	Kaffa	0	0	<b>114</b>	0	0	0
	Nekempti	0	0	0	<b>93</b>	0	0
	Sidamo	3	0	3	0	<b>87</b>	0
	Yirgacheffe	0	0	4	3	16	<b>86</b>

Table 4. 8 classification report of VGG19

Performance Metrics			
Types of coffee	Precision	Recall	F1-Score
Gujji	96%	71%	82%
Jimma	100%	100%	100%
Kaffa	94%	100%	97%
Nekempti	96%	100%	98%
Sidamo	66%	94%	78%
Yirgacheffe	100%	79%	88%
Weighted Average	92%	91%	91%

```

Epoch 45/50
39/39 [=====] - 168s 4s/step - loss: 0.4361 - accuracy: 0.8349 - val_loss: 0.2641 - val_accuracy:
0.8640

Epoch 00045: val_accuracy did not improve from 0.86458
Epoch 46/50
39/39 [=====] - 169s 4s/step - loss: 0.4010 - accuracy: 0.8494 - val_loss: 0.2937 - val_accuracy:
0.9132

Epoch 00046: val_accuracy improved from 0.86458 to 0.91319, saving model to VGG19_Model_-model.hdf5
Epoch 47/50
39/39 [=====] - 168s 4s/step - loss: 0.3346 - accuracy: 0.8734 - val_loss: 0.6025 - val_accuracy:
0.8125

Epoch 00047: val_accuracy did not improve from 0.91319
Epoch 48/50
39/39 [=====] - 174s 4s/step - loss: 0.4102 - accuracy: 0.8462 - val_loss: 0.5284 - val_accuracy:
0.8472

Epoch 00048: val_accuracy did not improve from 0.91319
Epoch 49/50
39/39 [=====] - 162s 4s/step - loss: 0.3888 - accuracy: 0.8542 - val_loss: 0.3928 - val_accuracy:
0.8125

Epoch 00049: val_accuracy did not improve from 0.91319
Epoch 50/50
39/39 [=====] - 161s 4s/step - loss: 0.3764 - accuracy: 0.8478 - val_loss: 0.3540 - val_accuracy:
0.8750

20/20 [=====] - 65s 3s/step
Test loss: 0.26032745838165283
Test accuracy: 0.8589743375778198

```

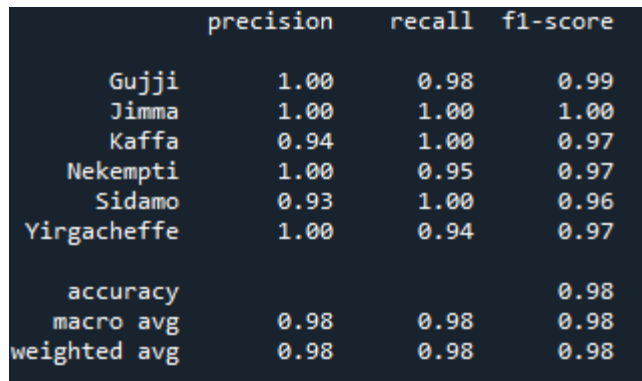
Figure 4. 8 sample Output of VGG19

We also conducted the experiment by splitting the dataset by 70% for training and 30% for testing. In this case the training dataset is 2184 and 936 is for testing the trained model. Before conducting the experiment all parameters were adjusted without any customization of the pre-trained network.

Finally, we got 85.7% training accuracy and 85.8% test accuracy as well as the training loss is 0.33% and test loss is 0.26%.

#### 4.5.2.2. Training and Validation Accuracy of VGG16 (Visual Geometry Group)

We have implemented VGG16 model by using the architecture described in the chapter 2 section 2.6.4.2. We have trained the model using our dataset for 50 epochs. VGG16 obtains 89.78 % training accuracy and 87.7% testing accuracy. This classification accuracy is obtained when the model was trained by data augmentation and percentage split of 0.1(10%) for test set. However, the result of both training and test accuracy as well as loss indicates this VGG16 pre-trained model needs different regularization techniques like batch normalization and dropout. Because as you have seen the accuracy of training is less than validation means the model have some overfitting.



	precision	recall	f1-score
Gujji	1.00	0.98	0.99
Jimma	1.00	1.00	1.00
Kaffa	0.94	1.00	0.97
Nekempti	1.00	0.95	0.97
Sidamo	0.93	1.00	0.96
Yirgacheffe	1.00	0.94	0.97
accuracy			0.98
macro avg	0.98	0.98	0.98
weighted avg	0.98	0.98	0.98

Figure 4. 9 Screenshot image of VGG16 classification report

We were tried to conduct the experiment through changing the percentage split value from 10% to 20% for test set. In this case, the training accuracy is 89.82% and 88.4% the testing accuracy as well as loss is 0.98 and 0.05 for training and test respectively. When compared to the pervious one which is 10% percentage split this one increased with 0.04 training accuracy and 0.7 for validation accuracy.

## 4.6. Summary of performance comparison

The summary for all experiments conducted in this research is shown in table 4.10 below. The table summarizes the result of training accuracy and testing accuracy using different train\_test percentage split value. The model is trained for 50 epochs for each experiment. Our proposed model architecture is achieved better accuracy through applying watershed segmentation algorithm. As shown table below the accuracy of model by using binary thresholding is also good.

Our proposed model is used deep CNN feature learning process. Because the depth (number of layers) is higher.

Table 4. 9 Comparison Table of All Experiment

Model	Test Option	Training Accuracy	Testing Accuracy
Proposed Model without Segmentation	70:30	79.09	76.3
Proposed Model using Watershed	70:30	<b>99</b>	<b>97.8</b>
Proposed Model using Otsu Thresholding	70:30	98.4	97.1
Proposed Model using Otsu Thresholding	80:20	98.6	97.5
VGG19 with Data Augmentation	80:20	87.34	86
VGG16 with Data Augmentation	80:20	89.82	88.4
VGG16 with Data Augmentation	90:10	89.78	87.7

Above table 4.9 indicates, the proposed model after applying watershed algorithm with the test accuracy of 97.8% was gotten. The proposed model has no model overfitting compared with pre-trained model.

#### 4.7. Comparison of Related Works with this Study

In this section, the previous published studies have compared with our current proposed study. As we discussed in chapter two section 2.7 mainly there are two published studies were conducted on green coffee bean classification based on their growing origin. The discussion has focused on the major findings of previous works to compare the findings of this study. Therefore, the detailed results of the comparison are shown in the following Table 4.10

Table 4.10 shows that the proposed method has better performance than the related works method which presented a system that done with machine learning techniques to create a model that can classify coffee bean. The overall experimental evaluation, conducted through the performance measure of coffee bean classification shows good result. To minimize computational resource, we applied two image segmentation algorithm to get the region of interest. As we can see, the immediate input fed into our proposed classification algorithm is the result of classifying coffee bean into distinct region. Though, human visual inspection is invaluable in determining the class of coffee bean, false estimations might also occur as bias on or loosing concentration are the natural

behavior of human being. Our algorithm to classify was tested using sample data selected from the dataset. On top of that, comparison of the proposed automated approach has performed better with respect to the manual system. As can be seen in Table 4.10, the performance of coffee bean classification model achieved 97.8%, which is a promising result.

Table 4. 10 Comparison of Related Works with this Study

No	Author	Title	Aim	Proposed Method	Accuracy
1	Habtamu Minassie	Image Analysis for Ethiopian Coffee Classification	To classify coffee into Bale, Harar, Jimma, Limu, Sidamo and Welega based on their growing regions.	neural networks classifier and Naïve Bayes classifier	<b>77.4%</b>
2	Asma Redi	Raw Quality Value Classification of Ethiopian Coffee Using Image Processing Techniques: In the case of Wollega region.	To identify coffee bean features which enable to measure the raw quality grade level of coffee beans	The Naïve Bayes, C4.5 and Artificial neural networks (ANN)	<b>82.72%</b>
<b>3</b>	Our Study	Image Based Coffee Bean Classification Using Deep Learning Technique	To classify coffee bean into distinct regions	Convolutional neural network using training from the scratch and watershed segmentation algorithm.	97.8%

#### 4.8. User Acceptance Testing

The user acceptance testing evaluation is carried out by the system's possible end-users during this study. Three domain experiments from the Ethiopian coffee quality inspection and auction center, as well as two domain experts from the Ethiopian commodity exchange, have been intentionally selected to review the prototype. Therefore, to analyze the system performance with user evaluations, the researcher assigned a value for each word to within scale. Such as Excellent=5, Very Good=4, Good=3 Fair=2, and Poor=1. Based on the given Likert scale, system evaluators

provide a value for each checklist. Thus, this method helps us to manually examine user acceptance based on the evaluator's response.

Table 4. 11 User Acceptance Evaluation Criteria and Their Results

No	Criteria's to Evaluate the Prototype	Poor	Fair	Good	Very Good	Excellent	Average score	Average Performan
1	Usability	0	0	2	2	1	3.8	76
2	Efficiency and effectiveness of the system	0	0	0	3	2	4.4	88
3	Attractiveness of the prototype	0		1	2	2	4.2	84
4	Simplicity to use	0	0	0	2	3	4.6	92
5	Accuracy	0	0	1	3	1	4	80
6	Error Tolerance	0	0	0	2	3	4.6	92
7	Importance of the system	0	0	0	3	2	4.4	88
	<b>Total Average</b>						<b>4.28</b>	<b>85.7%</b>

As shown in table 4.11, 40 % of the evaluators assessed the prototype system usability as Good, 40% gave it a Very Good rating and the remaining 20% rated it as Excellent. In the second criteria of evaluation, the prototype effectiveness and efficiency 40% of the evaluators gave it an Excellent rating and 60% gave it a Very Good rating. In the third category, which is Attractiveness of the prototype, 40% of the evaluators gave it an Excellent rating, 40% gave it a Very Good rating, and 20% gave it a Good rating. In fourth phase, 60% of respondents assessed its simplicity to use and interaction with its evaluation criteria as Excellent, while 40% ranked it as Very Good.

Related to accuracy of the developed system, 20% of the evaluators gave it an Excellent rating, 60% gave it a Very Good rating, and 20% gave it a Good rating. On the other hand, 60% of respondents assessed the system's capacity to prevent errors as Excellent, and 40% of respondents ranked it as Very Good. The final evaluation criterion is Importance of the system, which was scored as Very Good by 60% of the evaluators and with 40% of replies rated as Excellent. Finally, according to the domain experts' evaluation results, the prototype system's average performance is

4.28 out of 5. This result indicates that the coffee bean classification prototype overall average performance is 85.7%, which is above Very Good.

#### **4.9. Discussion of the Research Questions**

To the best of our knowledge on the area agricultural product classification to their predefined growing regions, limited study has been done. However, in related to coffee classification to their predefined growing regions using deep learning approach no work had been done. In this section the discussion has been on the findings of the result. Image resizing with the width and height value of 224x224, denoising image using median filtering and converting into grayscale or one channel was done. In this study, the researcher mentioned or listed three questions in chapter one to be answered at the end of the study. So, here we are going to answer the questions one by one.

**Question #1:** Which method is best from convolutional neural network algorithm for classification of Ethiopian coffee bean?

In this study, we applied two convolutional neural network training methods namely, training from scratch and transfer learning or pre-trained models which are developed by different researchers for various situation. Accordingly, we were designed one our own convolutional neural network architecture for the experiment using training from scratch by making a difference in parameters and its corresponding value. Although, we applied two segmentation algorithms namely watershed and binary thresholding by using the proposed convolution neural network architecture. Secondly, two pre-trained models were adopted. Comparatively, training from the scratch with watershed segmentation algorithm with the classification accuracy of 97.8% has given promising result for the classification of the coffee bean based on their growing regions than transfer learning. Therefore, we highly recommend that, applying segmentation on convolutional neural network will help the model to easily identify the high level features from the given images.

**Question #2:** Which segmentation algorithm is best for classification of Ethiopian coffee bean?

In this study, the researcher applied two different image segmentation algorithm to make the convolutional neural network easily learn the important features from the given images. Currently, different researchers applied Otsu thresholding and watershed segmentation algorithm [25] in order to find the region of interest and to minimize computation time and space in deep learning. As we have seen applying segmentation helps us it easily differentiates background from



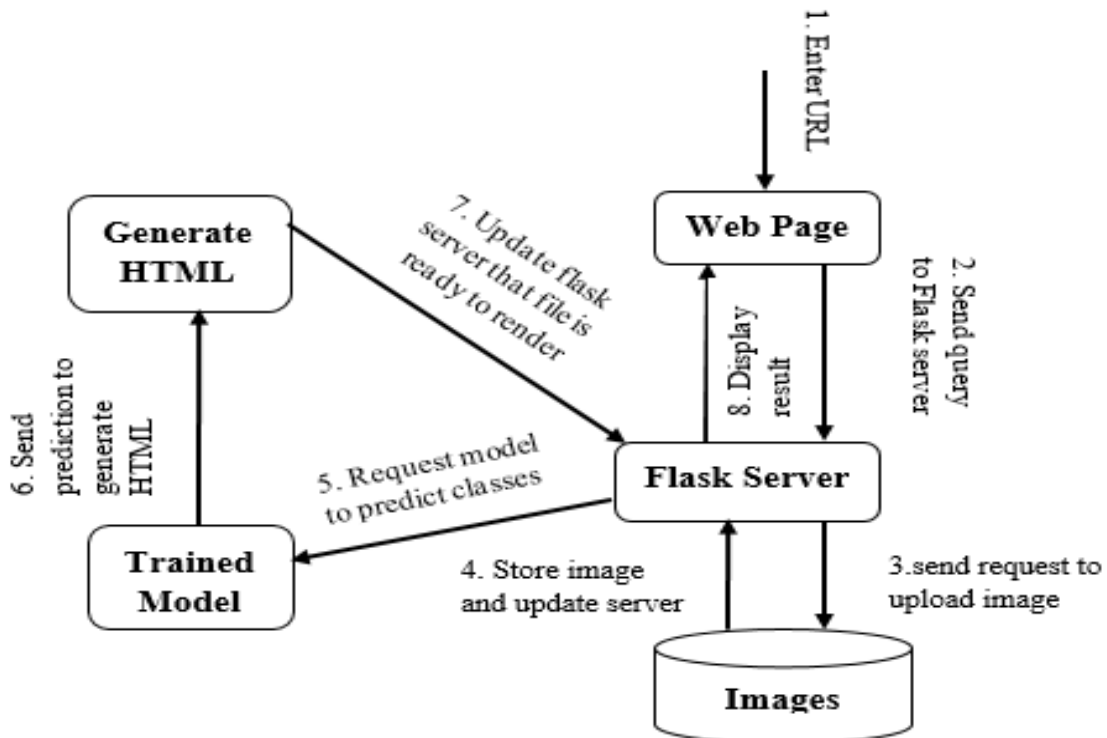
foreground. Comparatively, watershed algorithm with proposed convolutional neural network architecture achieved better result with the performance accuracy of 97.8% than Otsu thresholding.

**Question #3:** To what extent the deep learning algorithm classify the coffee bean images?

Developing classifier model is not enough. After developing classifier model you have to recognize the model predicting ability through comparing with previous related work. In this study, the performance of the developed classifier model was evaluated using a confusion matrix test. Besides, the most widely used classifier model performance evaluation metrics of precision, recall, and F-measure also applied. Comparatively, training from the scratch with watershed segmentation algorithm with the classification accuracy of 97.8% has given promising result for the classification of the coffee bean based on their growing regions than transfer learning.

#### 4.10. Web application design and Implementation

This section discusses the implementation to deploy the best performed Deep CNN classification model generated in the previous section 4.6.1 to be used in the development of flask web applications. The development and implementation process goes through different steps.



Generally, look at the following steps how to develop web application using flask:

**Step1:** import the different module specifically, load model module for loading trained model, Flask module, Request module to send the requests and render template module to display the designed HTML template

**Step2:** specify Flask name using flask function or method

**Step3:** specify the path of trained model and load the model by using load\_model method which embedded in load\_model module.

**Step4:** define the method which have variables in which the first variable specifies path of image and the second variable specifies the path of trained model. After specifying the path different image preprocessing and image segmentation which are stated in proposed architecture is applied.

**Step5:** IF-Else IF conditional statement is used to predict the classes of coffee bean.

**Step 6:** define @app. route method used to specify the way how to display the HTML page and define how to render the template.

**Step 7:** finally, use app. Run method in order to execute the written program. During execution don't forget specifying the value for port and debug variables.

## User Interface

User interface is the means of communication with the end user that enable to interact easily with the designing skin lesion classification system. The end user of the system can start the web

```
(base) PS C:\WINDOWS\system32> cd C:\Users\YG\Downloads\coffee_demo
(base) PS C:\Users\YG\Downloads\coffee_demo> python demo.py
2020-12-22 08:11:17.520121: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic
library cudart64_110.dll
* Serving Flask app "demo" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
2020-12-22 08:12:00.462071: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic
library cudart64_110.dll
* Debugger is active!
* Debugger PIN: 149-077-382
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 4. 10 Command prompt window to launch Flask server and to get IP address

application by write the URL of the coffee bean classification system on any web browser, after that the web browser send HTTP request to the web server and then the web server gives back HTTP response to browser.

Figure 4.13 shows that the user must enter the URL of coffee bean prediction system which is <http://127.0.0.1:5000/> to get the home page.

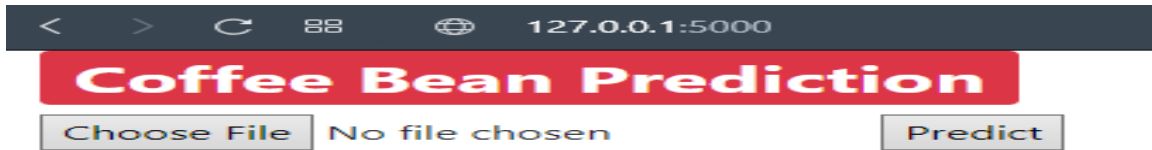


Figure 4. 12 Home page of proposed classification system

After end user enter the URL of coffee bean prediction system the home page displayed in above fig. After the interface illustrating in figure 4.15 displayed the user can click predict button and import the coffee bean image to know the category of coffee. Once the user import image the image will be displayed and the button called predict will be added in the interface. Figure 4.16 below shows the user import the image from local disk.

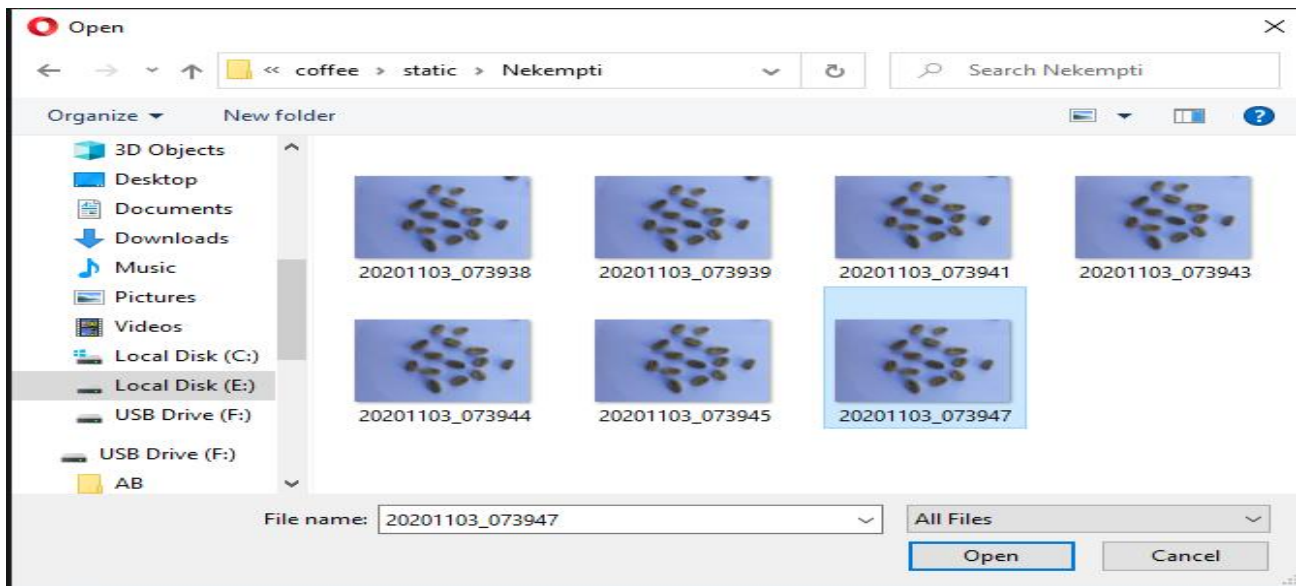


Figure 4. 13 Uploading coffee bean image from local disk

Figure 4.17 shows the modified Home page of green coffee bean classification system after importing coffee bean image. Here the figure 4.17 below indicates that, the user imported coffee bean image.

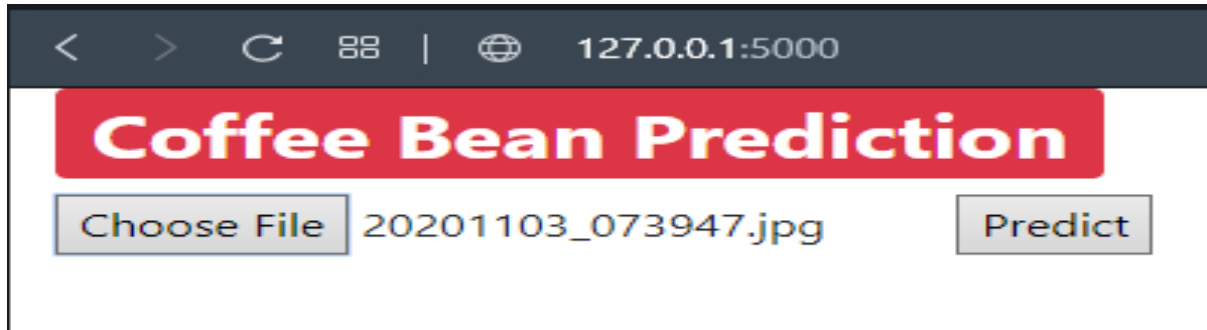


Figure 4. 14 Home page after the coffee bean image uploaded

After the interface illustrating in figure 4.17 displayed the user can click predict button and see the classification result which contain the predicted name of coffee class and shows the sample result to the given coffee bean image.

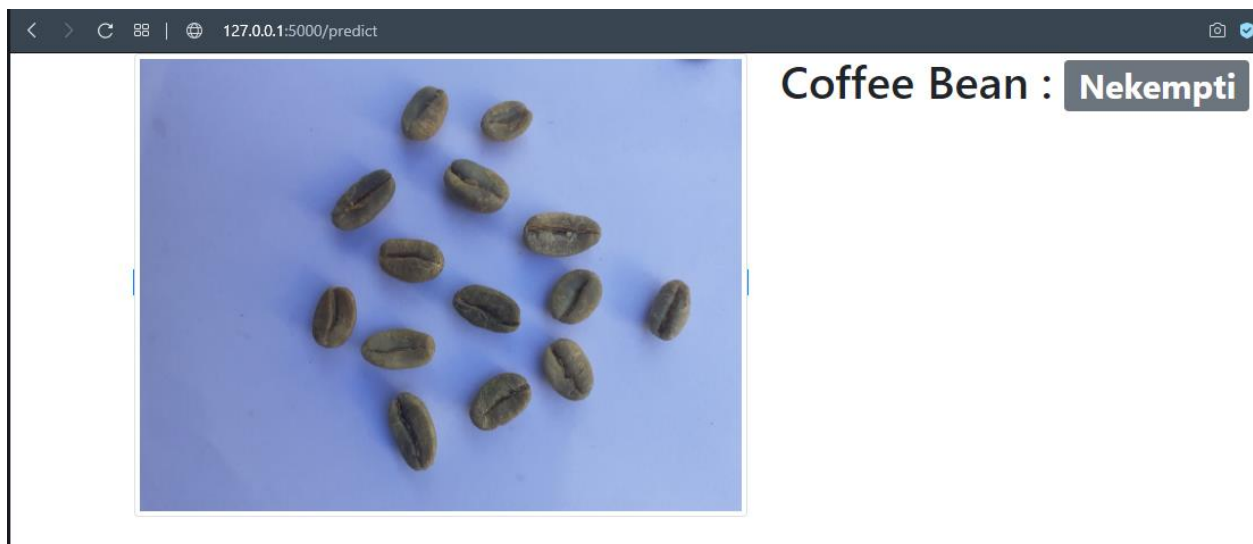


Figure 4. 15 Sample Result of classification system

In this study, we tried to test all classes for the sake of documentation we were documented two sample screenshot.

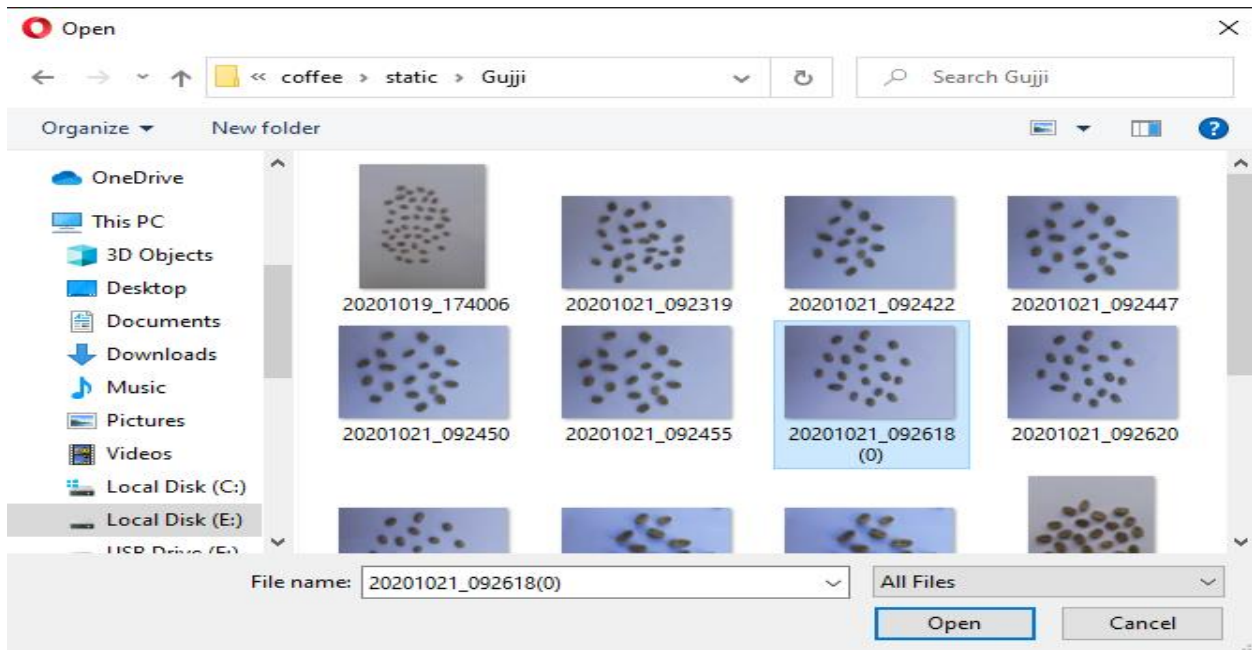


Figure 4. 16 uploading sample Gujji image



Figure 4. 17 uploaded Gujji image interface

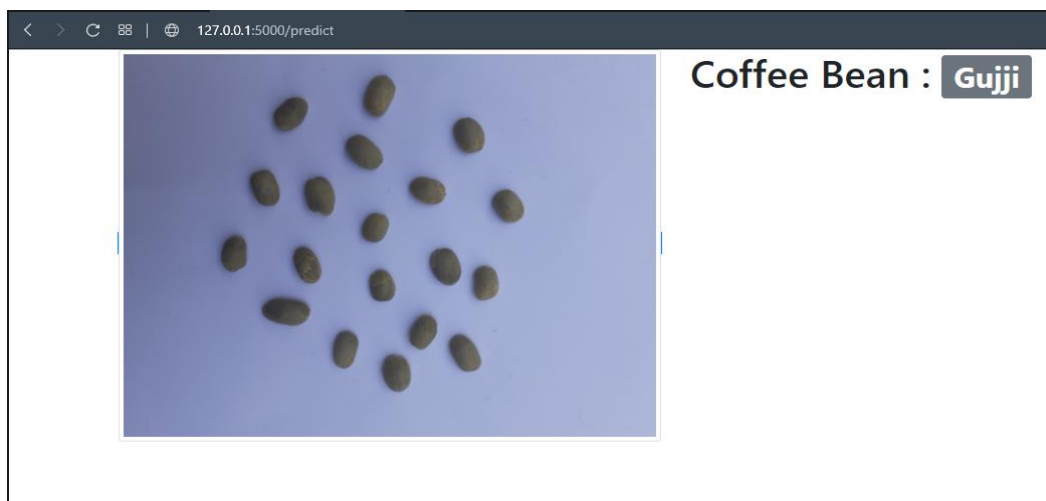


Figure 4. 18 Classification Result

## **CHAPTER FIVE**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1. Conclusion**

Coffee is a commercial commodity which, among Ethiopia's export commodities, plays a major role in earning foreign currency. Due to its importance in commercial activities, the sub-sector attracts governmental and nongovernmental attention. During current periods, the brand patent development of each coffee variety based on growing region was a problem. For instance, one of these issues has been the recent controversy over Starbucks' Yirgacheffe coffee brand [12]. Automated classification systems for agricultural products are proven to be less costly, efficient and non-destructive. Accordingly, this research has focused on using image processing techniques and approaches to classify raw quality value of sample coffee beans.

In this research, the methodology was a design science research approach that grows from relevance and rigor. Therefore, the researcher has tried to follow relevance for identifying problems, opportunities that exist in the business, and applicable knowledge that allows us to develop an artifact from rigor.

We proposed convolutional neural network, which is recently achieving promising accuracy in image processing and due to its automatic feature extraction. Image preprocessing and image segmentation was applied to improve the proposed model classification performance. Based on their quantity or production amount and availability of coffee bean in ECQIAC we were limited to conduct this study for six classes and the others are beyond our scope. In the classification problem of Ethiopian coffee based on growing region, morphological and color features were automatically extracted from a coffee bean images taken from six regions of Ethiopia – Gujji, Jimma, Kaffa, Nekempti, Sidamo and Yirgacheffe – by using image analysis techniques 520 images per class and totally, 3120 sample coffee bean images are collected from ECQIAC to conduct this research.

To perform this analysis a deep learning-based techniques was used. Specifically, proposed CNN architectures and transfer learning CNN techniques have been used to create a model that can classify coffee bean based on their region of origin. Also, the augmented images were provided to mitigate the overfitting problem during training the model, we have generated another image dataset by modifying the original images. VGG-16 and VGG-19 pre-trained models utilized and

the last layers of these model are modified as this study problem and softmax layer and fully connected layers are added to the network architectures. Accordingly, to develop a classification, model 80%:20%, 70%:30%, and 90%:10% percentage split test options have been utilized and a confusion matrix and classification report to visualize the model performance.

In order to produce computationally effective and higher accuracy in the CNN model, we have applied image segmentation separate part of image or region of interest from the background of an image. The result demonstrated are satisfactory and, the proposed model can obtain a classification test accuracy of 97.8 % which is higher than the recognition ability of other states of the art CNN architecture that are selected to compare VGG16 and VGG19. And finally, developed CNN model was test through developing user interface and achieved **85.7%** user acceptance accuracy.

## **5.2. Contribution of this study**

As a contribution to the scientific world or the knowledge, the proposed CNN model offered a systematic approach for coffee bean classification based on their region of grown. According to best of our knowledge, there is no research was done in this area using a deep learning approach. Considering the CNN model, we proposed the model that predict more accurate than the state-of-the-art model. In this work, therefore, we improved the performance, the proposed model achieves better result in terms of accuracy, loss, training time and model size. In terms of accuracy we have got 97.8% testing accuracy that are far from above the state-of-the-art model.

Secondly, improved performance and four new classes are different from previous coffee bean classification by considering different parameters, the previous coffee bean classification achieved 77.4% accuracy these fill to local minima but in our work, we have 97.8% test accuracy.

## **5.3. Recommendation**

Based on the investigation and findings of the study, we recommended for future and further research works:

- ✓ To improve the performance of the model, future works need to integrate increasing and cleaning the dataset by continuing to collect more from the field. Future studies are need to examine the CNN model by restructuring the layers with deep hidden layer with better dataset size, to examine the CNN model by with better architecture like GoogLeNet and ResNet with large dataset.

- ✓ Due to different constraints like budget and time we didn't incorporate various coffee bean varieties such as; - Harar, Wollega and more. So, we highly recommend this as future research direction.
- ✓ Image processing technology has grown significantly over the past decade. The developed model only applicable on web or desktop. However, its application on low-power mobile devices has been the interest of a wide research group related to newly emerging contexts. With the emergence of general-purpose computing on embedded GPUs and their programming models like OpenGL ES 2.0 and OpenCL, mobile processors are gaining a more parallel computing capability. So, an advanced image-processing mobile application for coffee bean classification can be recommended as further research work.



## References

- [1] E. Bechere, "Agricultural Research and Development in Ethiopia," *International conference on Africa Development Archives*, 2007.
- [2] S. Ponte, "The 'Latte Revolution'? Regulation, Markets and Consumption in the Global Coffee Chain," *Web Development*, 2002.
- [3] H. Desta, "Development of Automatic Sesame Grain Classification and Grading System Using Image Processing Techniques," 2017.
- [4] M. Mogese, "Connecting Ethiopian Coffee to Sustainable Market," March 2016. [Online]. Available: <https://etbuna.com/ethiopian-coffee/ethiopian-coffee-processing/>. [Accessed 22 February 2021].
- [5] A. Y. a. L. B. Demelash Teferi, "Assessment Report on Effect of Dust on Coffee Production at Biftu and Quarry Site," Ethiopian Institute of Agricultural Research, Biftu, February 06/2018.
- [6] D. Kaur, "Various Image Segmentation Techniques," *International Journal of Computer Science and Mobile Computing*, May 2014.
- [7] S. M. a. P. Prabhu, "Digital Image Processing Techniques – A Survey," *Golden Research Thoughts*, 2016.
- [8] A. S. Getachew, "Automatic Skin Lesion Classification in Dermoscopic Images using Deep Neural Networks," College of Software Nankai University, China, May, 2019.
- [9] E. Alpaydin, "Introduction to Machine Learning.," 2010.
- [10] S. R. Karan Chauhan, "Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures Using Tensorflow and Keras," *International Journal of Advance Engineering and Research Development*, vol. 5, No. 02, February -2018.
- [11] A. R. Baleker, "Raw Quality Value Classification of Ethiopian Coffee Using Image Processing Techniques: In the case of Wollega region," Addis Ababa University, Addis Ababa, 2011.
- [12] H. Minassie, "Image Analysis for Ethiopian Coffee Classification," 2008.
- [13] A. D. . D. P. L. J. A. V. A. Jr., Design Science Research, Springer, 2015.
- [14] M. A. a. T. T. Ken Peffer, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, January, 2018.

- [15] F. Alemu, "Assessment of the Current Status of Coffee Diseases at Gedeo and Sidama zone, Ethiopia," *International Journal of Advanced Research*, vol. 1, No. 8, 2013.
- [16] "Ethiopian Coffee Origin and History," [Online]. Available: <https://etbuna.com/ethiopian-coffee/ethiopian-coffee-origin-and-history>. [Accessed 11 3 2012].
- [17] mekuria, "The Status of Coffee Production and The Potential for Organic Conversion in Ethiopia," in *International Agricultural Research For Development*, 2004.
- [18] "ECX Coffee Contracts," Addis Ababa, march 2018.
- [19] "Food and Agriculture Arganization in the United Nation," [Online]. Available: [www.fao.org](http://www.fao.org). [Accessed 22 february 2021].
- [20] "33 tone coffee," [Online]. Available: <https://www.33toncoffee.com/learn/coffee-research/ethiopia-coffee-research.html>. [Accessed Tuesday February 2021].
- [21] W. J. Boot, "Practical Guideline for Purchasing and Importing Ethiopian Specialty Coffee Bean," in *United States Agency for International Development*, Addis Ababa, March 2011.
- [22] P. G. a. D. N. Venkatachalapathy, "Processing and Drying of Coffee – A Review," *International Journal of Engineering Research & Technology*, vol. 3, No. 12, December-2014.
- [23] J. J. G. a. L. J. v. V. Ian T. Young, *Fundamentals of Image Processing*, 2007.
- [24] T. B. Chris Solomon, *Fundamentals of Digital Image Processing*, Wiley Blackwell, 2011.
- [25] D. A. Worku, "Automatic Flower Disease Identification Using Deep Convolutional Neural Network," February 2020.
- [26] G. Tigistu, "Automatic Flower Disease Identification Using Image Processing," Addis Ababa University, Addis Ababa, 2015.
- [27] S. N. a. B. Patel, "Machine Vision based Fruit Classification and Grading - A Review," *International Journal of Computer Applications*, vol. 170, International Journal of Computer Applications.
- [28] A. Solomon, "Automatic Skin Lesion Classification in Dermoscopic Images using Deep Neural Network," *Nankai University*, May, 2019.
- [29] Abdalla Mohamed Hambal, Dr. Zhijun Pei, Faustini Libent Ishabailu, "Image Noise Reduction and Filtering Techniques," *International Journal of Science and Research*, vol. 6, no. 3, March 2017 .

- [30] "Statistical Approach to Compare Image Denoising Techniques in Medical MR Images," *International Conference on Pervasive Computing Advances and Applications*, p. 367–374, 2019.
- [31] A. Buades, B. Coll, and J. M. morel, "A Review of Image Denoising Algorithms, with A New," *Society for Industrial and Applied Mathematics*, vol. 4, no. 2, 2010.
- [32] Kumar, A., & Sodhi, S. S., "Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoencoder," *International Conference on Computing for Sustainable Global Development*, 2020.
- [33] I. S. a. G. E. H. Alex Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, 2012.
- [34] D. Yadav, "Categorical encoding using Label-Encoding and One-Hot-Encoder," *Towards Data science*, 6 December 2019. [Online]. Available: <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd>. [Accessed 9 December 2020].
- [35] A. B. Mekonnen, "A Deep Learning-Based Approach for Potato Leaf Diseases Classification," *Debre Berhan University*, June 2020.
- [36] D. K. a. Y. Kaur, "Various Image Segmentation Techniques: A Review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, p. 809 – 814, May 2014.
- [37] A. Bala, "An Improved Watershed Image Segmentation Technique using MATLAB," *International Journal of Scientific & Engineering Research*, vol. 3, no. 6, 2012.
- [38] J. Yousef, "Image Binarization using Otsu Thresholding Algorithm," *University of Guelph, Ontario, Canada*, April 18, 2011.
- [39] S. A. Medjahed, "A Comparative Study of Feature Extraction Methods in Images Classification," *I.J. Image, Graphics and Signal Processing*, 2015.
- [40] D. p. Tian, "A Review on Image Feature Extraction and Representation Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, July, 2013.
- [41] T. A. a. S. Mansi, "Feature Extraction for Object and Image Classification," *International Journal of Engineering Research and Technology*, vol. 2, pp. 1238-1246, 2013.
- [42] A. Anand, "Image classification," January 2017.
- [43] H. R. S. A. A. S. a. M. B. Salman Khan, "A Guide to Convolutional Neural Networks for Computer Vision," in *Synthesis Lecture on Computer Vision*, Morgan and Claypool, 2018.
- [44] A. Habeeb, "Artificial intelligence," *Discover the world's scientific knowledge*, June, 2018.

- [45] D. A. Rosebrock, *Deep Learning for Computer Vision with Python*, Pyimagesearch, 2017.
- [46] H. V. H. G. L. a. S. J. Xuan-Hien Le, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," *MDPI*, 2019.
- [47] S. Pouyanfar, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM Computing Surveys*, vol. 51, no. 5, 2018..
- [48] K. Rungta, *Tensorflow in one Day*, 2018.
- [49] N. K. Manaswi, *Deep Learning with Applications using Python*, Apress, 2018.
- [50] S. K. Deepika Jaswal, "Image Classification Using Convolutional Neural Networks," *International Journal of Scientific & Engineering Research*, vol. 5, no. 6, June, 2014.
- [51] V. D. a. F. Visin, *A Guide to Convolution Arithmetic for Deep Learning*, January, 2018.
- [52] J. Gu, "Recent Advances in Convolutional Neural Networks," *Elsevier*, October 2017.
- [53] A. K. a. S. P. Antonio Gulli, *Deep Learning with TensorFlow 2 and Keras*, Packt, December 2019.
- [54] A. Dertata, "Towards Data Science," [Online]. Available: <https://towardsdatascience.com>. [Accessed 9 3 2020].
- [55] C. S. a. T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, 2019.
- [56] J. Brownlee, "Machine Learning Mastery," 5 July 2019. [Online]. Available: <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>. [Accessed Tuesday December 8 2020].
- [57] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," *International Journal of Pattern Recognition and Artificial Intelligence*, 08 May 2020.
- [58] A. Saha, "ADAM (Adaptive Moment Estimation) Optimization | ML," 14 January 2020. [Online]. Available: <https://www.geeksforgeeks.org/adam-adaptive-moment-estimation-optimization-ml/>. [Accessed 08 December 2020].
- [59] J. Jordan, "Data science," [Online]. Available: <https://www.jeremyjordan.me>. [Accessed 3 9 2020].
- [60] S. Indoliaa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *International Journal of Scientific & Engineering Research*, 2018.

- [61] D. M. A. a. s. G. M. Abraham Debasu Mengistu, "Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques," *International Journal of Database Theory and Application*, vol. 9, pp. 79-88, (2016).
- [62] B. H. a. T. Tegegne, "Ethiopian Roasted Coffee Classification Using Imaging Techniques," in *International Conference on the Advancement of Science and Technology*, Bahirdar, 2015.
- [63] S. G. a. R. Patil, "Deep Learning for Image Based Mango Leaf Disease Detection," *International Journal of Recent Technology and Engineering*, vol. 8, November 2019.
- [64] V. A. Patel, "Convolutional Neural Network with Transfer Learning for Rice Type Classification," -September 2017.
- [65] D. T. Ayane, "Automatic Plant Species Identification Using Image Processing Techniques," *Addis Ababa University*, October, 2018.
- [66] P. C. D. a. S. K. Guru, "Survey on Image Resizing Techniques," *International Journal of Science and Research*, vol. 3, no. 12, December 2014.
- [67] C. Seger, "An Investigation of Categorical Variable Encoding Techniques in Machine Learning: Binary Versus One-Hot and Feature Hashing," in *KTH Royal Institute of Technology*, Sweden, 2018.
- [68] Y. B. a. A. P. Shervin Minae, "Image Segmentation Using Deep Learning: A Survey," in *Cornell University*, Cornell, 15 Nov 2020.
- [69] N. A. a. R. K. Kulkarni, "Efficient Image Segmentation Using Watershed Transform," *International Journal of Computer Science And Technology*, vol. 4, no. 2, April - June 2013.
- [70] F. Schilling, "The Effect of Batch Normalization on Deep Convolutional Neural Networks," *KTH Royal Institute of Technology*, Sweden, 2016.
- [71] B. Lake, "Mobile Based Expert System for Diagnosis of Cattle Skin Diseases with Image Processing Techniques," *Addis Ababa University*, October 2019.
- [72] K. Danjuma, "Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients," *International Journal of Computer Science Issues*, vol. 12, no. 2, March 2015.
- [73] fchollet, "Keras," 28 July 2018. [Online]. Available: <https://Keras.io/>. [Accessed 10 December 2020].
- [74] P. Chansung, "Transfer Learning in Tensorflow," 6 june 2018. [Online]. Available: <https://towardsdatascience.com/transfer-learning-in-tensorflow-9e4f7eae3bb4>. [Accessed 29 December 2020].

- [75] F. A. A. a. H. N. Mohammed, "Efficient Way of Web Development Using Python And Flask," *Reference International Journal of Advanced Research in Computer Science* , vol. 6 , no. 2, 2015.
- [76] "Assessment Report on Effect of Dust on Coffee Production at Biftu," Biftu.
- [77] S. Ponte, "The 'Latte Revolution'? Regulation, Markets and Consumption in the Global Coffee Chain," *World Development*, 2002.
- [78] D. kaur, "Various Image Segmentation Techniques," *international journal of computer science and mobile computing*, may 2014.
- [79] S. M. a. P. Prabhu, "Digital Image Processing Techniques – A Survey," *International Multidisciplinary Research Journal*, vol. 5, no. 11, May 2016.

## **Appendix A:**

### **Interview Questions**

#### **Dear Respondents;**

First of all, I would like to thank you for your willingness to present for the interview and to share the knowledge. My name is Gebreyes Gebeyehu and I'm studying master of science in information systems in Debre Berhan University. I'm conducting my research work on developing image based coffee bean classification based on their growing region. To build the above model successful, as a researcher I should have to get the answer for the following questions provided below.

1. How many coffee varieties found in Ethiopia?
2. What are the most common coffee variety produced in Ethiopia?
3. How do you differentiate coffee varieties?
4. What are the most determinant features to identify the coffee?
5. What is the colour of each coffee variety?
6. Is any material (Apparatus) used for differentiating?
7. How you can measure the size and moisture of the coffee bean?
8. What is the acceptable value for the size of the coffee?
9. The value of moisture content must be?

## Appendix B:

# -\*- coding: utf-8 -\*-

.....

Created on Mon Nov 30 03:56:44 2020

@author: UNKNOWN

'''

**###Importing Different Libraries**

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import os
import itertools
import keras
os.environ['KERAS_BACKEND'] = 'tensorflow'
```

**###Reading Images**

```
data_path = 'A:/Y/Dataset/normal/'
dataset = []
label = []
gujji = os.listdir(data_path + 'G_filtered/')
for i, image in enumerate(gujji):
    input_img = cv2.imread(data_path + 'G_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(0)
img_data = np.array(dataset)
img_data = img_data.astype('float32')
img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
jimma = os.listdir(data_path + 'J_filtered/')
for i, image in enumerate(jimma):
    input_img = cv2.imread(data_path + 'J_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(1)
img_data = np.array(dataset)
img_data = img_data.astype('float32')
```



```

img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
#img_data = img_data.astype('float32')
#img_data/=255
kaffa = os.listdir(data_path + 'K_filtered/')
for i, image in enumerate(kaffa):
    input_img = cv2.imread(data_path + 'K_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(2)
img_data = np.array(dataset)
img_data = img_data.astype('float32')
img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
nekempti = os.listdir(data_path + 'N_filtered/')
for i, image in enumerate(nekempti):
    input_img = cv2.imread(data_path + 'N_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(3)
img_data = np.array(dataset)
img_data = img_data.astype('float32')
img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
sidamo = os.listdir(data_path + 'S_filtered/')
for i, image in enumerate(sidamo):
    input_img = cv2.imread(data_path + 'S_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(4)
img_data = np.array(dataset)
img_data = img_data.astype('float32')
img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
yirga = os.listdir(data_path + 'Y_filtered/')
for i, image in enumerate(yirga):
    input_img = cv2.imread(data_path + 'Y_filtered/' + image)
    #input_img = cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
    input_img = cv2.resize(input_img, (224,224))
    dataset.append(input_img)
    label.append(5)
### Changing into numpy array and Normalizing the input data

```

```
img_data = np.array(dataset)
img_data = img_data.astype('float32')
img_data/=255
img_data = img_data.reshape((-1, 224, 224, 1))
```

### ###Importing different keras module

```
from keras.models import Model, Sequential
from keras.layers import Conv2D,Activation,MaxPool2D,Dense,Flatten,Dropout
from keras.optimizers import Adam
from keras.layers.normalization import BatchNormalization
from sklearn.preprocessing import label_binarize
from sklearn.metrics import confusion_matrix, classification_report
```

```
yes = Sequential()
input_shape = (224, 224, 1)
num_classes = 6
```

```
#input_img = (224,224,1)
yes.add(Conv2D(32,(3,3), padding = 'valid', input_shape = input_shape))
yes.add(Activation('relu'))
yes.add(MaxPool2D(pool_size=(2,2), strides=2))
yes.add(Dropout(rate = 0.2))
```

```
yes.add(Conv2D(32,(3,3), padding = 'valid'))
yes.add(Activation('relu'))
yes.add(MaxPool2D(pool_size=(2,2), strides=2))
yes.add(Dropout(rate = 0.25))
```

```
yes.add(Conv2D(64,(3,3), padding = 'valid'))
yes.add(Activation('relu'))
yes.add(MaxPool2D(pool_size=(2,2), strides =2)) ###strides =2
yes.add(Dropout(rate = 0.25))
```

```
yes.add(Conv2D(64,(3,3), padding = 'valid'))
yes.add(Activation('relu'))
yes.add(MaxPool2D(pool_size=(2,2), strides=2))
yes.add(Dropout(rate = 0.5))
```

```
yes.add(Flatten())
yes.add(Dense(128, activation = 'relu'))
```

```
yes.add(Dropout(rate = 0.5))
yes.add(Dense(num_classes, activation='softmax'))
print(yes.summary())
```

### ###Compiling the model

```
yes.compile(optimizer = 'Adam' , loss = "categorical_crossentropy", metrics=["accuracy"])
# Set a learning rate annealer
from keras.callbacks import ReduceLRonPlateau, ModelCheckpoint, CSVLogger, EarlyStopping
learning_rate_reduction = ReduceLRonPlateau(monitor='val_accuracy',patience=25, verbose=1,
factor=0.5, min_lr=0.00001)
#filepath = "saved-model/weights-improvement.hf5"
checkpoint = ModelCheckpoint('saved-model_weights-improvement_dropout.h5', monitor =
"val_accuracy", verbose = 1, save_best_only=True, mode = 'max' )
early = EarlyStopping(monitor="val_loss", patience=25, verbose=1)
csv = CSVLogger("without_seg_flow_all_dropout.csv", separator=',', append=False)
callbacks_list = [checkpoint, learning_rate_reduction, early, csv]
```

### ### Splitting the Image dataset into training and test

```
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
X_train, X_test, Y_train, Y_test = train_test_split(img_data,
to_categorical(np.array(label)),test_size=0.2,random_state=0)
x_train, x_validate, y_train, y_validate = train_test_split(X_train, Y_train, test_size = 0.1,
random_state = 0)
```

### ###Data Transformation

```
from keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    brightness_range=[0.2,1.2],
    fill_mode='nearest')
```

### ###Training the model

```

epochs = 50
batch_size = 32
history = yes.fit_generator(datagen.flow(x_train,y_train, batch_size=batch_size),
                             epochs = epochs, validation_data = (x_validate, y_validate),
                             verbose = 1, steps_per_epoch=x_train.shape[0] // batch_size
                             , callbacks=callbacks_list)
loss, accuracy = yes.evaluate(X_test, Y_test, verbose=1)
loss_v, accuracy_v = yes.evaluate(x_validate, y_validate, verbose=1)
print("Validation: accuracy = %f ; loss_v = %f" % (accuracy_v, loss_v))
print("Test: accuracy = %f ; loss = %f" % (accuracy, loss))
#yes.save("model_without_seg_flow.h5")

```

#####Sample screenshot of Model training result

```

Epoch 00030: val_accuracy improved from 0.98718 to 0.99145, saving model to saved-model_weights-improvement.h5
Epoch 31/50
2184/2184 [=====] - 74s 34ms/step - loss: 0.0200 - accuracy: 0.9922 - val_loss: 0.0882 -
val_accuracy: 0.9744

Epoch 00031: val_accuracy did not improve from 0.99145
Epoch 32/50
2184/2184 [=====] - 73s 33ms/step - loss: 0.0234 - accuracy: 0.9927 - val_loss: 0.0412 -
val_accuracy: 0.9882

Epoch 00032: val_accuracy did not improve from 0.99145
Epoch 33/50
2184/2184 [=====] - 71s 32ms/step - loss: 0.0375 - accuracy: 0.9867 - val_loss: 0.0730 -
val_accuracy: 0.9765

Epoch 00033: val_accuracy did not improve from 0.99145
Epoch 34/50
2184/2184 [=====] - 70s 32ms/step - loss: 0.0219 - accuracy: 0.9922 - val_loss: 0.0677 -
val_accuracy: 0.9829

Epoch 00034: val_accuracy did not improve from 0.99145
Epoch 35/50
2184/2184 [=====] - 70s 32ms/step - loss: 0.0306 - accuracy: 0.9904 - val_loss: 0.0433 -
val_accuracy: 0.9893

Epoch 00035: val_accuracy did not improve from 0.99145
Epoch 36/50
2184/2184 [=====] - 73s 33ms/step - loss: 0.0175 - accuracy: 0.9950 - val loss: 0.0533 -

```

IPython console History

LSP Python: ready conda: python 37 (Python 3.7.9) Line 35, Col 29 UTF-8 CRLF RW