



**TECHNICAL AND VOCATIONAL TRAINING
INSTITUTE**

**FACULTY OF ELECTRICAL ELECTRONICS AND
INFORMATION COMMUNICATION TECHNOLOGY**

**MSc. THESIS ON
CLASSIFICATION OF MALT BARLEY SEED USING
ENSEMBLE DEEP LEARNING TECHNIQUE**

By: Dessie Abebe

August -2023

Addis Ababa, Ethiopia



**TECHNICAL AND VOCATIONAL TRAINING
INSTITUTE**

**DEPARTMENT OF INFORMATION TECHNOLOGY
MSc. THESIS ON
CLASSIFICATION OF MALT BARLEY SEED USING
ENSEMBLE DEEP LEARNING TECHNIQUE**

**A Thesis Submitted to the Department of Information
Technology of Technical and Vocational Training Institute
in Partial Fulfilment of the Requirements for the Masters of
Science in Information Technology**

By: Dessie Abebe

Advisor: Habtom G/Egizabher(Ass.prof.)

August -2023

Addis Ababa, Ethiopia

DECLARATION

This is to certify that the thesis entitled “Classification of Malt Barley Seed using Ensemble Deep Learning Technique”, submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology under the faculty of Electrical Electronics and Information Communication Technology, Technical and Vocational Training Institute is a record of original work carried out by me and has never been submitted to this or any other institution to get and other degree or certification. The assistance and help I received during this investigation have been duly acknowledged.

Declared by

Name _____

Signature: _____

Department _____

Date _____

CERTIFICATION

This is to certify that the thesis prepared by Dessie Abebe, entitled “Classification of Malt Barley Seed using Ensemble Deep Learning Technique” and submitted in partial fulfillment of the requirements for the Degree of Masters of **Information Technology** complies with the regulations of the University and meets the accepted standards for originality and quality.

Name of Candidate: _____ Signature: _____ Date: _____.

Name of Advisor: _____ Signature: _____ Date: _____.

Signature of Board of Examiners:

External examiner: _____ Signature: _____ Date: _____.

Internal examiner: _____ Signature: _____ Date: _____.

Dean, SGS: _____ Signature: _____ Date: _____

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God without his assistance nothing would have been successful.

I am also expressing my deepest gratitude to my advisor, Mr. Habtom G/Egizabhire (Ass. Prof) for his commitment, constructive guidance, and suggestion right from the moments of problem formulation to the completion of the work. Many thanks and appreciations go to him for showing important directions, which always made me think that things are possible.

I would also like to thank Prof. Ravendra Babus, for his support, constructive suggestion and encouragement at the very beginning of my research.

I would like to thank the Federal Technical and Vocational Training Institution and all departments of Information Technology instructors and students who have a significant contribution in one way or another during my research work.

I would like to forward my special thanks to the Boortmalt Debre Birhan Factory for supplying samples of malt barley varieties. I would like to extend my appreciation and thanks to laboratory experts and all staff members of the organization for their kind cooperation.

Dessie Abebe

ABSTRACT

The primary step in making beer is selecting the malt barley. Every malt house requires that the types of grain be checked before being purchased. Varietal uniformity is essential for the manufacture of high-quality malt. It is challenging to distinguish between different types of malt barley during inspection since it calls for training and experience. To tackle varietal selection difficulties Malt industries have employed and trained experts; even though, those experts do not work effectively due to tiredness, bias, and other factors. Therefore, many researchers are motivated to the development of classification model based on image processing to support experts across the world. Using digital image processing algorithms based on combined morphological, texture, and color features have been investigated to classify different varieties of Ethiopian malt barley. Sample malt barleys were taken from Debre Berhan Malt Factory which is the only nearby area for taking the varieties. The total number of images taken was containing 13030 malt barley seeds of 4 classes Holker, Misikal, Propino, and Sabini 3257, 3257, 3258, and 3258 respectively. To build the classification models of malt barley varieties, image preprocessing, median filtering, and image augmentation are involved and finally, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and the ensemble of the two techniques are investigated to classify malt barley seed. Based on experimental results, an ensemble model of KNN and SVM outperforms a single model constructed using either KNN or SVM, on combined features of morphology, texture, and color employing the Sequential Forward Feature Selection (SFFS) technique. Quantitatively, the accuracy of KNN is 98.54%, the accuracy of SVM is 98.15% and an ensemble accuracy of 98.58% is achieved for Holker, Propino, Sabini, and Misikal varieties with the combined feature sets of morphology, color, and texture using the ensemble of KNN and SVM.

Keywords: Malt barley seed classification, deep learning, KNN, SVM, and an ensemble of KNN and SVM.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF GRAPHS	xi
LIST OF ABBREVIATIONS.....	xii
CHAPTER ONE.....	1
1. INTRODUCTION	1
1.1. Background	1
1.2 Statement of the Problem	2
1.3 Research Questions	2
1.4 Objectives.....	2
1.4.1 General objective.....	2
1.4.2 Specific objectives.....	3
1.5 Scope and Limitation of the Study.....	3
1.6 Significance of the Study	4
1.7 Organization of the Thesis	4
CHAPTER TWO	6
2. REVIEW OF RELATED LITERATURE	6
2.1 Overview	6
2.2 Malt Barley Selection and Grading Technique.....	6
2.3 Image processing.....	7
2.3.1 Image Acquisition.....	8
2.3.2 Image Preprocessing.....	8

2.3.3 Segmentation	9
2.3.4 Feature Extraction.....	10
2.3.5 Image Identification/Classification.....	11
2.3.5.1 K-Nearest Neighbor (KNN) Classifier	12
2.3.5.2 Support Vector Machine (SVM) Classifier	12
2.3.5.3 Ensemble Approach.....	12
2.4 Related Work.....	13
2.4.1 Barley Varieties Identification.....	13
2.4.2 Application of image processing techniques for malt-barley seed identification	13
2.4.3 Varietal Classification of Barley by convolutional neural networks.....	14
2.5 Gap Analysis	14
2.6 Summary of related works	15
2.7 Summary	16
CHAPTER THREE	17
3. METHODOLOGY	17
3.1 Overview	17
3.1.1 Literature Review	17
3.1.2 Data Set Preparation.....	17
3.1.3 Implementation Procedure and Tools.....	18
3.1.4 Evaluation Procedure.....	18
3.2 Research Methodology Approaches.....	20
3.3 Design of Malt Barley Classification Model.....	20
3.3.1 Image acquisition.....	21
3.3.2 Image Preprocessing.....	22
3.3.2.1 Grayscale Conversion	22

3.3.2.2 Noise Removal/ Denoising	23
3.3.2.2.1 Mean filtering.....	23
3.3.2.2.2 Median Filter	24
3.3.2.2.3 Gaussian Filter.....	24
3.3.2.3 Binarization.....	25
3.3.2.4 Edge detection.....	26
3.3.3 Image Segmentation	27
3.3.3.1 Otsu thresholding.....	27
3.3.3.2 Watershed segmentation	28
3.3.4 Image Augmentation	30
3.3.5 Dataset Splitting	30
3.3.6 Feature Extraction.....	31
3.3.6.1 Morphological Feature Extraction	31
3.3.6.2 Texture Feature Extraction	32
3.3.6.3 Color Feature Extraction.....	32
3.3.7 Classification	32
3.3.7.1 K-Nearest Neighbor (KNN) Classifier	33
3.3.7.2 Support Vector Machine (SVM) Classifier	33
3.3.7.3 Ensemble of K-Nearest Neighbor and Support Vector Machine.....	33
3.4 Summary	35
CHAPTER FOUR.....	36
4. Experimentation.....	36
4.1 Introduction	36
4.2 Dataset Preparation	36
4.3 Building Proposed Model.....	37

4.4 Experimental Setup	38
4.5 Experimental Result and Discussion.....	39
4.5.1 Experiment result before preprocessing	39
4.5.1.1 Confusion matrices for KNN before Preprocessing	39
4.5.1.2 Classification report KNN before Preprocessing.....	40
4.5.1.3 Confusion matrices for SVM before Preprocessing	40
4.5.1.4 Classification report SVM before Preprocessing.....	40
4.5.1.5 Confusion Matrices for Ensemble before Preprocessing.....	41
4.5.1.6 Classification Report Ensemble before Preprocessing	41
4.5.2 Experiment Result after Preprocessing.....	42
4.5.2.1 Confusion matrices for KNN after Preprocessing	43
4.5.2.2 Classification Report for KNN after Preprocessing.....	43
4.5.2.3 Graph of Classification Report for KNN Classifier.....	44
4.5.2.4 Confusion matrices for SVM after Preprocessing	45
4.5.2.5 Classification Report for SVM after Preprocessing.....	45
4.5.2.6 Graph of Classification Report for SVM Classifier.....	46
4.5.2.7 Confusion Matrices for Ensemble after Preprocessing.....	46
4.5.2.8 Classification Report for Ensemble Classifier after Preprocessing	47
4.5.2.9 Graph of Classification Report for Ensemble of KNN and SVM Classifier	47
4.6 Discussion of Results	48
4.7 Research question demonstration.....	48
4.8 Summary	49
CHAPTER FIVE	50
5. CONCLUSION AND RECOMMENDATION.....	50
5.1. Conclusion.....	50

5.2. Recommendation.....	51
REFERENCES	52
APPENDIX.....	I
Appendix I Sample Python Code	I

LIST OF FIGURES

Figure 2.1 image preprocessing steps	7
Figure 3.1 Design science research methodology (Peppers et al., 2007).....	20
Figure 3.2 Architecture of the proposed model for Ethiopian malt barley classification.....	21
Figure 3.3 Sample scanned Holker malt barley seed; (a) RGB image, (b) Gray level image	23
Figure 3.4 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Mean filtering images	24
Figure 3.5 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Median filtering images	24
Figure 3.6 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Gaussian filtering images.....	25
Figure 3.7 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of binary images	26
Figure 3.8 Sample scanned Holker malt barley seed; (a) original RGB image, (b) Result of horizontal Sobel, (c) Result of horizontal sobel (d) Result of edge detected image.....	27
Figure 3.9 Malt barley image segmentation using Otsu thresholding; (a) Original image, (b) Result of forward segmentation using Otsu, (c) Result of reverse segmentation using Otsu.....	28
Figure 3.10 Malt barley image s Watershed segmentation; (a) Original image, (b) Result of erosion segmentation, (c) Result of dilation segmentation, (d) Result of opening operation, (e) Result of the closing operation	30
Figure 3.11 Dataset split distribution.....	31
Figure 4.1 Proposed approaches for MBCM	37
Figure 4.2 Confusion matrixes for KNN before preprocessing	39
Figure 4.3 Precision, recall, and f1-score using KNN classifier before preprocessing	40
Figure 4.4 Confusion matrixes for SVM before preprocessing	40
Figure 4.5 Precision, recall, and f1-score using SVM classifier before preprocessing	41
Figure 4.6 Confusion matrixes for ensemble before preprocessing	41
Figure 4.7 Precision, recall, and f1-score using ensemble classifier before preprocessing.....	42
Figure 4.8 Confusion matrixes for KNN after preprocessing.....	43
Figure 4.9 Precision, recall, and f1-score using KNN classifier after preprocessing	44
Figure 4.10 Confusion matrixes for SVM after preprocessing.....	45

Figure 4.11 Precision, recall, and f1-score using SVM classifier after preprocessing	45
Figure 4.12 Confusion matrixes for ensemble after preprocessing	46
Figure 4.13 Precision, recall, and f1-score using ensemble classifier after preprocessing.....	47

LIST OF TABLES

Table 2.1 Summary of related works	15
Table 3.1 Sample format of n x n confusion matrix	18
Table 4.1 An overview of the malt barley seed image that was taken for the current study	36
Table 4.2 Summary of Experimental Setup	38
Table 4.3 Experiment result before dataset preprocessing	39
Table 4.4 Experiment result for classifiers after preprocessing.....	42
Table 4.5 Compare with different previous studies	48

LIST OF GRAPHS

Graph 4.1 Dataset before augmentation (left) and after augmentation (right)	37
Graph 4.2 Accuracy before preprocessing (left) and after preprocessing (right)	43
Graph 4.3 Graph of precision, recall, and f1-score using KNN classification before preprocessing (left) and after preprocessing (right)	44
Graph 4.4 Graph of precision, recall, and f1-score using SVM classification before preprocessing (left) and after preprocessing (right)	46
Graph 4.5 Graph of precision, recall, and f1-score using Ensemble of KNN and SVM classification before preprocessing (left) and after preprocessing (right)	47

LIST OF ABBREVIATIONS

Abbreviations	Meaning
ANN	Artificial Neural Network
CNN	Conventional Neural Network
CPU	Central Processing Unit
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbors
MBCM	Malt Barley Classification Model
MRI	Magnetic Resonance Image
RAM	Random Access Memory
RBF	Radial Basis Function
RGB	Red, Green, and Blue
SVM	Support Vector Machine
SFFS	Sequential Forward Feature Selection
TN	True Negative
TP	True Positive
WTO	World Trade Organization

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Barley is a member of the grass family and a main cereal grain grown in mild winter climates globally. As early as 10,000 years ago, it was one of the first cereals to be domesticated, especially in Eurasia. Worldwide, 30% of the production of barley is used as beer, and 70% is used as animal feed. Together with barley bread from many cultures, it is used in soups and stews(*Barley - Wikipedia*, n.d.). Barley grains are frequently processed into malt using an old fashioned technique.

Barley is one of the most important agricultural products and is used for reducing the risks of stomach cancer, lowering cholesterol levels, weight loss, reducing inflammatory bowel diseases, increasing strength and energy, and as a source of vitamins and minerals in manufacturing industries(*Barley - Wikipedia*, n.d.).

Barley is used as a food sweetener and as a raw material for the production of alcoholic beverages in manufacturing industries. The majority of alcoholic beverages have been brewed with barley. The various attributes of barley are the parameters that producers are most concerned about, and different malt barley seeds have different quality attributes, which also makes malt barley seed classification an important issue in this research.

It is possible to identify, classify, and grade agricultural products with a human operator but they are usually inconsistent, error-prone, and inefficient; So, I'm inspired towards the replacement of a human operator with automated systems for faster and more accurate analysis.

In this study ensemble deep learning approaches is used. It is a successful approach to train multiple models instead of a single model and to combine K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), to classify from these models(Ganaie et al., 2022).

1.2 Statement of the Problem

The Ethiopian Malt Factory currently uses manual malt barley seed classification methods and subject matter experts try to differentiate the malt barley seeds utilizing some physical and chemical characteristics of barley seeds. This result in time consuming during the classification of malt barley seed.

Even though specialists made an effort to identify each barley seed, the effectiveness of doing so mostly depends on the knowledge and experience of experts in the relevant field. This frequently results in errors, inefficiencies, and inconsistent results.

Hence, the Factories attempt to tackle the problems occurred, during the classification of malt barley seed varieties, by experts through giving, training, and establishing a well-equipped laboratory by spending extra budget. To make the classification of malt barley seeds easier, a model for classifying malt barley seeds is being designed.

So, the goal of this study is to determine whether it is feasible to design a malt barley seed classification model using K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and ensemble K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) of a deep learning algorithm.

1.3 Research Questions

To this end, the following research questions will be formulated:

RQ1. Which filtering technique is best for the classification of malt barely seed?

RQ2. What is the suitable classification algorithm for design a model for malt barley seed classification?

RQ3. What will be done to improve a malt barley seed classification model performance?

1.4 Objectives

1.4.1 General objective

The general objective of this study is to design model that classifies Ethiopian malt barley seed using ensemble deep learning techniques.

1.4.2 Specific objectives

The study has the following specific objectives:

- ❖ Reading previously conducted research on barley classification.
- ❖ Identifying gaps of previously conducted research.
- ❖ Identify a suitable classification algorithm to classify malt barley seed.
- ❖ Applying image preprocessing, segmentation, augmentation and dataset splitting of the augmented dataset to improve model performance.

1.5 Scope and Limitation of the Study

The main purpose of this study is to explore deep learning techniques and approaches to construct a model for malt barley classification. Which play a vital role in malt barley classification, and for constructing a malt barley classification model using K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and the ensemble of K-Nearest Neighbors (KNN), and Support Vector Machine (SVM).

Climate conditions, type of soil, fertilizer, and others also affect the production of barley seeds. Hence, this also causes the barley seeds to deviate from their normal state. Therefore, these factors were the potential causes for the misclassification of malt barley seeds, and setting the dataset preparation environment was one of the most critical and challenging tasks during our experimentation. Since there is no well-organized barley seed image dataset for both training and testing purposes, we can use different techniques like augmentation for datasets for the experiment.

This study focus only on 4 malt barley seed which are released by Boortmalt Debre Birhan malt factory are Holker, Misikal, Propino and Sabini we do not includes other verities seed.

The performance of the system has been assessed via varieties of malted barley that are gathered from the Boortmalt Debre Birhan malt factory. Furthermore, non-barley components were not mixed with the sample data since malt barley seed varieties, which are identified by domain experts using different physical and biological characteristics, are available in malt factories.

The accuracy was acquired below laboratory settings, so it had some limits like the quality of the camera, and factors related to the seed growing environment that may affect the result.

1.6 Significance of the Study

We expect that this research work provides different benefits to the target populations in the domain area identified. After the evaluation of the results found in this research work, the two major stakeholders, malt factories and breweries, have benefited.

The Boortmalt Debre Birhan malt factory uses the models of malt barley seed classification as a framework to assist the experts during malt barley seed classification. At the same time, any brewery industry that uses Boortmalt Debre Birhan malt factory products as input for beverage production get pure malt, which is made from pure malt barley seed, and it also enables the expertise to save time and cost with the support of the classification model.

Moreover, this research can also be used as a cornerstone in the area and referred to by future researchers in the domain area. The researcher also be the beneficiary by getting the necessary research background and fulfilling the requirement of the academic program.

1.7 Organization of the Thesis

This thesis is comprised of the five chapters below:

Chapter One: includes the background of the study, a statement of the problem, the objective of the study, the scope and limitation, and the significance of the research.

Chapter Two: covers the literature review which gives a detailed overview of the study area, a review of related works, and summary (indicating the gap).

Chapter Three: covers methodology overview, general approach and research design/method, data collection method, and procedure, and experimentation approaches and techniques.

Chapter Four: deals with the experimentation activity done to execute the methods and approaches stated in chapter three; the dataset used experimental setup and the results of the experimentations and comparison of previous study are covered.

Chapter Five: provides the conclusion and future research direction for scholars interested in the area.

CHAPTER TWO

2. REVIEW OF RELATED LITERATURE

2.1 Overview

Barley is the basic grain utilized in the manufacturing of malt. Historically, this may have resulted from its greater availability compared with other cereals (Mulatu & Grando, 2011). Classification of barley seed varieties is extremely important to the brewing industry because variation determines the energy of germination, enzymes, course relaxing speed, malt performance, protein content, resistance to microbial contamination, and others. Nowadays, experts are responsible for manually identifying varieties using classification keys and tables as the basis of the variety diagnosis, through which classification is largely dependent on the skills and experience of the evaluator (Kozłowski et al., 2019). Therefore, there should be a means to classify malt barley seed varieties. Designing a classification model for malt barley seed classification is one of the solutions to the problem.

Barley is a significant staple food crop in the highlands of northern Ethiopia. The crop is used to make a variety of traditional meals, including malt, kita, kolo, beso, and enjera (Araya et al., 2011). In the following section, only barleys used for malting purposes are discussed.

2.2 Malt Barley Selection and Grading Technique

In Ethiopia, barley is an important cereal crop that covers 10% of the land under crop cultivation (Mulatu et al., n.d.). Currently, malting barley is regarded as one of the cash crops, and malt factories have expanded their need for it as a result of the country's expanding breweries, rising beer consumption, and increased malt barley processing capacity (*Barley Production in Ethiopia*, n.d.). As long as the barley is cultivated and used, it has lost some of its important agronomic and brewing quality (change in shape, color, texture), due to the effect of environmental factors such as soil type, fertilizer usage, rainfall amount, and others, which has greater impact for the final added value in crop production (Mulatu et al., n.d.).

According to World Trade Organization (WTO), crop grading and selection based standards primarily depend on the origin of the crop, its defects, and its physical characteristics (Araya et al., 2011). In the process of barley selection for malt producers or the brewery industry, the level

of impurities must be minimized, and varieties and the evaluation of the kernel size of barley are regarded as the most important step(Nowakowski et al., 2011).

2.3 Image processing

Image processing is the technique through which images are manipulated in various ways using computer technology; to get clear and enhanced images. Image processing improves the visual appearance of images and simplifies the extraction of useful information for further image analysis(*Image Processing: The Fundamentals - Maria M. P. Petrou, Costas Petrou - Google Books*, n.d.).The basic steps of image processing can be broadly classified into five (Image acquisition; Image preprocessing, Segmentation, Feature extraction, and Image identification).

Image processing starts with image acquisition using a camera or scanner. Sometimes we may receive noisy images due to various effects. So in the preprocessing stage, the raw image noise and distortion are reduced and the image is refined and simplified for the next stages of processing load. After preprocessing we segment objects of interest within the image and then extraction of a set of representative features which are segmented. Finally, based on the extracted features decision is made and each segmented object is identified.

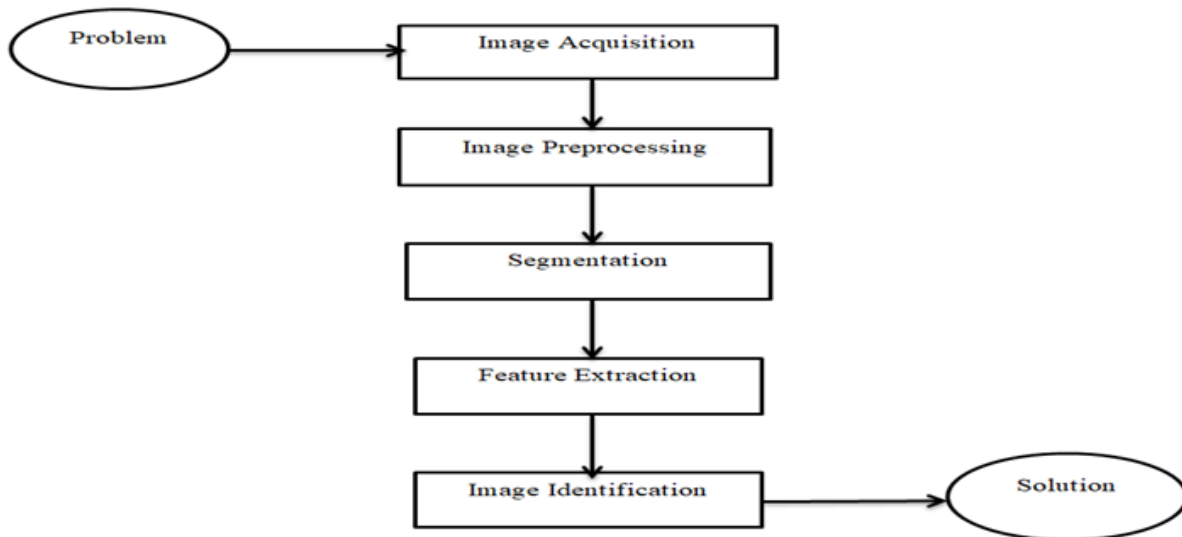


Figure 2.1 image preprocessing steps

2.3.1 Image Acquisition

Image acquisition is the process of obtaining a digital image of an object in the real world using a camera or scanner. In the process of acquiring each image, we may introduce noise which is a random variation in the values of the image's pixels. When acquiring images, pixel values might fluctuate for a variety of reasons, including the deterioration of the environment, camera motion, scanner quality, and the distance between the camera lens and the image (*Image Processing: The Fundamentals - Maria M. P. Petrou, Costas Petrou - Google Books*, n.d.).

2.3.2 Image Preprocessing

The image preprocessing technique is the initial step for enhancing the quality of raw images (*A Beginner's Guide to Image Preprocessing Techniques - Jyotismita Chaki, Nilanjan Dey - Google Books*, n.d.). The preprocessing stage contains a number of operations that are applied for successive changes of the image. These are color image to gray-scale conversion, noise removal, binarization, boundary enhancement, and edge detection.

Converting Color Image to Grayscale: A color image is a red, green, and blue (RGB) image that is saved as an m-by-n-by-3 data array and identifies the color components for each pixel. The range of a pixel values for grayscale images is from 0 to 255. Using a Payton called `rgb2gray()`, a color image is converted to grayscale. This process involves producing a weighted sum of the R, G, and B components, as given in equation 2.1 below (*Convert RGB Image to Grayscale - MATLAB Im2gray*, n.d.).

$$0.2989 * R + 0.5870 * G + 0.1140 * B \dots\dots\dots (2.1)$$

Noise Removal/Filtering: Random changes in the values of the image's pixels cause noise to exist. A variety of noise types can potentially be found in digital images(*Techniques & Methods For Image Filtering & Noise Reduction | Talentica Blog*, n.d.). The most prevalent noise types are multiplicative noise (salt and pepper noise), additive noise (Gaussian noise), and impulsive noise (thermal noise)(*Noise (Electronics) - Wikipedia*, n.d.). While salt and pepper noise is typically the result of a transmission error or an atmospheric disturbance appearing as a black-and-white color on the image, thermal noise is caused by photo-electronic sensors, and Gaussian noise contains intensity variations that are drawn from a Gaussian distribution and are caused by

camera electronics errors. One common technique for decreasing such noise is the employment of median and Gaussian filters. Median filtering techniques are used to lessen noise in photos of malted barley seeds. These methods have the benefit of not losing contrast between steps, keeping boundaries, being less sensitive to extreme values or outliers than the mean, keeping edges while removing noise, and having excellent noise-reduction capabilities for some types of random noise with significantly less blur than the others(*Techniques & Methods For Image Filtering & Noise Reduction* / *Talentica Blog*, n.d.).

Image Binarization: is the process of converting either a color image or a grayscale image, to a bi-level image through which each pixel is categorized either as a foreground or a background pixel. And it is an important preprocessing method in image analysis and boundary detection(Sauvola & Pietikäinen, 2000). Image binarization or thresholding is used to differentiate the content from the background. During the process of binarization, image pixels are classified into two values one and zero, which characterize foreground and background classes respectively. Binarization is done based on the image pixel value; for a threshold value T , if the value of the image pixel is larger than or equal to the threshold then it becomes one otherwise it is zero(Sauvola & Pietikäinen, 2000).

Edge Enhancement/Detection: Edges are the boundaries between segments, and edge detection describes the general problem of determining the location, magnitude, and segment of boundaries in an image(*Edge Enhancement - Wikipedia*, n.d.). Edge detection is the process of locating the edge pixels while edge enhancement is the process of increasing the contrast between the edges and the background(*Edge Detection*, n.d.). In this case, the Sobel edge detection strategy is employed. Sobel edge detection is a 3×3 neighborhood-based gradient operator which uses vertical and horizontal masks. Then, on the input image, apply these two masks independently to produce two gradient components G_x and G_y in horizontal and vertical orientations, respectively.

2.3.3 Segmentation

Segmentation is the process of partitioning an image into a set of homogeneous and meaningful regions with a set of identical properties or attributes. It can also be defined as a subdivision process of an image, in image analysis, into different parts or objects. Segmentation can be done

at both gray level and binary images(*Image Segmentation: Principles, Techniques, and Applications - Tao Lei, Asoke K. Nandi - Google Books, n.d.*).

The image is typically divided during segmentation until the objects of interest are isolated from the background. Two techniques are used to segment values based on gray levels. The first one, which divides a picture based on sharp changes in gray level, is based on the discontinuity of gray-level values; the second, which employs thresholding and region growth, is based on the similarity of gray-level values. The histogram analysis's findings are used to determine the threshold value, which is expected to remain consistent across all photographs taken under identical lighting conditions. As a result, the content of the object may be distinguished from its background using a predetermined threshold value calculated from the histogram of the gray plane.

2.3.4 Feature Extraction

Feature extraction is the process of retrieving meaningful information from an image that is used for the identification /classification of images into different categories(*What Is Feature Extraction - Search, n.d.*). The features of the images and the extraction technique determine the nature and the results of the segmentation and image preprocessing phases. The image features and extraction methods operate on color images, gray level images, or binary images as long as the extracted features' format complies with the classifier's specifications. Therefore, the currently proposed identification approach, which takes into account an evaluation of human visual inspection and physical features of each malt barley seed as starting point, is based on morphology, texture, and color analysis. We employ three Python-based feature extraction techniques morphology, texture, and color to extract features.

Morphology Feature: A wide range of image processing techniques that process images based on shapes. Each pixel in the image is modified during a morphological operation based on the values of the pixels nearby. It is also defined as the geometric properties of images, such as area, height, width, perimeter, maximum diameters, minimum diameters, and others (*Understanding Morphological Image Processing and Its Operations | by Prateek Chhikara | Towards Data Science, n.d.*).

Texture Feature Extraction: is a feature that is used to divide and categorize images into different areas of interest. It provides data regarding an image's color or intensity spatial composition. The texture for a point can't be defined(Sci-Hub | *Texture Feature Extraction Techniques for Image Recognition. SpringerBriefs in Applied Sciences and Technology / 10.1007/978-981-15-0853-0*, n.d.).

Color Feature Extraction: a technique used in image processing to extract features from an image based on its color. It is one of the three major types of image features. The color feature extraction process involves the use of color spaces such as RGB, to extract color features from an image(Zhang, 2019).

2.3.5 Image Identification/Classification

Identification is the process of finding a model that explains and separates data classes in order to forecast the class of objects whose class label (Wu & Chen, 2016).

The two fundamental steps in the classification process are testing and training. In the training step, the classifier learns all extracted features and the relationships between each sample in the labeled sample class. In the testing phase, the trained system is applied to data that it has never seen before to evaluate the classification's effectiveness. The majority of researchers aim to create classifiers with small test errors. Different types of pattern classification techniques have been used in machine learning for the identification of patterns. These classification methods are mainly categorized as supervised learning and unsupervised learning type.

In supervised classification, the classifier is trained using a set of training pattern samples, and the resulting new set can be classified as a member of a known class. In contrast, in unsupervised classification, the system partitions or clusters the entire data set based on some similarity criteria, and each cluster of patterns belongs to a particular class.

It is obvious that no single strategy to classification is ideal in many of the new applications. As a result, mixing different classifiers is now a frequent classification technique. The K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and an ensemble of the two approaches are the most popular classification algorithms.

2.3.5.1 K-Nearest Neighbor (KNN) Classifier

A k-nearest neighbor algorithm predicts the likelihood that a data point belongs to one of two groups based on which group the data points that are closest to it belong to. The distance can be determined using any well accepted method. An illustration of a distance is the Euclidean distance. Following that, we get a certain feature value from all the training sets nearby (Ganaie et al., 2022).

During classification, a test sample is categorized based on the highest number of votes among the K neighbors, with the sample being given to the class most common from its K-nearest neighbors. The positive integer K is determined using the trial and error method which provides the smallest error rate. Then the sample is simply given to the class of its nearest neighbor.

2.3.5.2 Support Vector Machine (SVM) Classifier

SVM is a short form of Support Vector Machine. SVM has demonstrated performance with several classification issues. It tries to identify the optimum hyperplane among the classes by counting the number of points on the edge of the class descriptors (Smola & Schölkopf, 2004). The margin is the distance between classes. The classification accuracy increases as the margin of error increases. SVM can be used to address classification and regression issues. This approach is effective for resolving issues involving both linear and nonlinear datasets. The SVM technique uses a variety of kernel types for classification models, including polynomial, sigmoid, and linear radial basis functions (RBF). The optimal hyperplane for classifying data points into two groups is selected by SVM after searching for characteristics in high-dimensional space. It works with unmanageable datasets of all sizes, small and huge.

2.3.5.3 Ensemble Approach

A single classification model is inefficient in the majority of emerging applications; hence, several approaches must be integrated to produce ensemble models. There are certain limitations to using simple methods to create classification systems that utilize the best individual classifiers (Ganaie et al., 2022).

A collection of individual classifiers plus a combiner can be used to improve the classification performance of a system. The performance of the system can be improved in several ways by using multiple classifiers. Each classifier has the potential to be trained in a unique area of the

feature space and can offer probability estimation and decision making based on the study of individual outcomes. An ensemble is a composite model for classification that combines different classifiers. The individual classifiers vote and the ensemble then predict the class label based on the total number of votes.

The test sample is supplied to the classifier during testing, which causes it to assign the sample to one of the predetermined classes even though it has never seen it before. We utilized a row-by-column matrix known as a confusion matrix to assess each supervised type of classifier's performance on the test dataset. The percentage of accurate and inaccurate predictions generated by the classifier in comparison to the actual classifications in the test data is shown in a confusion matrix. A confusion matrix shows the actual classification numbers in rows and the models predicted classification numbers in columns. The diagonal elements of the matrix show the proportion of correctly categorized samples for each class(Ganaie et al., 2022).

2.4 Related Work

2.4.1 Barley Varieties Identification

According to Nowakowski et al(Nowakowski et al., 2012), The application of image analysis technology aided and facilitated the selection of malt barley. The researchers made an effort to evaluate the efficacy of image analysis in the selection and identification of malt barley using four different types of barley and 1200 images. At various levels of preprocessing, an artificial neural network classifier that used the Radial Basic Function Network (RBF) and features including morphology and color was able to classify objects with an accuracy of roughly 94%.The researcher's also recommended conducting additional research utilizing various feature extraction strategies to obtain the most accurate representative features for use in kernel image analysis.

2.4.2 Application of image processing techniques for malt barley seed identification

According to HAILU, BIRHANU(HAILU, 2020), The primary step in the manufacturing of beer is the selection of malt barley. Every malt house requires that the types of grain be checked before being purchased. The creation of quality malt depends on varietal uniformity. It is challenging to distinguish between different malt barley cultivars during inspection since it takes knowledge and practice to address challenges with varietal selection Malt industries have trained

and employed professionals, yet they are not able to do their duties to the best of their abilities because of fatigue, bias, and other issues. As a result, many researchers are driven to create autonomous prediction models based on image processing in order to assist specialists throughout the world. However, to the researcher's knowledge, no attempts have been made to identify newly developed malt barley types from Ethiopia; hence an effort has been made in this work to do so. A total of 208 pictures were captured, and each one included 5408 malt barley seeds. K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), and the combination of the two techniques are researched to construct identification models for the prediction of malt barley types. K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), and the ensemble of the two techniques are examined to develop the identification models for the prediction of malt barley types. Using the Sequential Forward Feature Selection (SFFS) technique, experimental results show that an ensemble model combining ANN and KNN is better output with accuracy of 95.1% than single model built using either ANN or KNN.

2.4.3 Varietal Classification of Barley by convolutional neural networks

According to Kozowski et al(Kozłowski et al., 2019) assuring high-quality beer is a significant issue in the brewing industry. This can be realized by maintaining control over the brewing process and the quality of the components. The currently used visual inspection of malting barley is time consuming, expensive, and requires specialist knowledge. It has been shown that computer vision algorithms may take the role of human expertise in the identification of barley types and the detection of flawed grains. This study examines the application of convolutional neural network implementations to varietal classification and accuracy of 93%.

2.5 Gap Analysis

Gap analysis is a technique used to identify the gap between the current state of a system and its desired state. Performing a gap analysis of classification malt barley seeds using ensemble deep learning involves evaluating the current state of classification practices and identifying areas where ensemble deep learning techniques can be implemented to improve classification accuracy and efficiency. Here's a step-by-step process for conducting such an analysis:

- ❖ Define the current classification practices: This may include reviewing documentation, interviewing experts, and studying the available literature.

- ✓ Barley Varieties Identification using convolutional neural networks by Nowakowski et al it use CNN model, dataset of 1200, accuracy is 94% and its limitation are Lack of dataset, single model and low accuracy.
 - ✓ Application of image processing techniques for malt barley seed identification by HAILU, BIRHANU it use ensemble of ANN and KNN model, dataset of 5408 ,accuracy is 95.1% and its limitation are Lack of dataset, low Accuracy and Unbalance data.
 - ✓ Varietal classification of malt barley by convolutional neural networks by Kozłowski et al it use CNN model , Dataset for training, valuation and test sets, containing 62 753, 6276 and 6276 images respectively, accuracy is 93% and its limitation are single model and low accuracy.
- ❖ Familiarize with ensemble deep learning: Gain an understanding of ensemble deep learning techniques, such as ensemble models. Explore how these techniques can be applied to classification tasks and their potential benefits over individual models.
- ✓ For the proposed model it use the ensemble of SVM and KNN model it get better accuracy than the individual model which is accuracy of SVM is 98.54%, accuracy of KNN is 98.15 and ensemble of SVM and KNN is 98.58%.
- ❖ Identify the gaps in current classification practices: Compare the capabilities of current classification practices with the advantages offered by ensemble deep learning. Identify the gaps where ensemble techniques can potentially address limitations, such as improving accuracy.
- ✓ For the proposed model it use the ensemble of SVM and KNN model it get better accuracy 98.58% than the previously study it address the accuracy limitation for the previously study.
 - ✓ For the proposed model it use large dataset than the previously study it address the dataset limitation for the previously study.

2.6 Summary of related works

Table 2.1 Summary of related works

Author(s)	Objective	Methodology	Result	Limitation(Gap)
-----------	-----------	-------------	--------	-----------------

Nowakowski et al	Barley Varieties Identification using Artificial neural networks	Model :ANN model Dataset : 1200	94%	<ul style="list-style-type: none"> ✓ Lack of dataset ✓ single model ✓ low accuracy
HAILU, BIRHANU	Application of image processing techniques for malt barley seed identification	Model : ensemble of ANN and KNN Dataset : 5408	95.1%	<ul style="list-style-type: none"> ✓ Lack of dataset ✓ low accuracy ✓ Unbalance data
Kozłowski et al	Varietal classification of malt barley by convolutional neural networks	Model : CNN model Dataset : training, valuation and test sets, containing 62 753, 6276 and 6276 images respectively	93%	<ul style="list-style-type: none"> ✓ single model ✓ low accuracy

2.7 Summary

In this chapter, we have reviewed the literature on malt barley identification, and the theoretical background of image processing such as image acquisition; preprocessing, segmentation, feature extraction, and identification, are discussed.

In this section, we have also reviewed related works on the classification of malt barley seeds that are related to our study using image processing, machine learning, and deep learning techniques. In this thesis, we propose to develop malt barley seed classification using the KNN, SVM, and ensemble of, KNN and SVM deep learning techniques.

CHAPTER THREE

3. METHODOLOGY

3.1 Overview

A research methodology is a group of steps, strategies, resources, and instruments that will assist the researcher in planning, carrying out, and documenting the research. Research design is the setting up of parameters for data collection and analysis in a way that intends to address the research topic. The goal of this research is to design a model for classifying malt barley seeds. The study employs experimental research. Conducting experimental research entails reviewing the literature, creating the dataset, putting it into practice, and evaluating the results(Gero, n.d.).

3.1.1 Literature Review

Various local and international studies have been examined for the successful completion of this study to acquire a clear understanding from the study's inception to its conclusion. For reaching the research objective and addressing research issues, a variety of sources have been examined, including journal articles, conference papers, manuals, corporate reports, books, and the internet.

3.1.2 Data Set Preparation

Malt barley seeds for training and testing the malt barley classification model has been collected from the Boortmalt Debre Birhan malt factory. Sampling is the primary method for gathering samples of each malt barley seed. For the goal of gathering the necessary data regarding malt barley seeds for this research project, we have conducted an interview with the domain experts who were chosen using a purposeful sample technique. Furthermore, other data sources and collection techniques included personal observation and record reviews. Since there is no readymade dataset for this type of research, we have prepared our dataset for training and performance evaluation. To do this, malt barley seeds have been collected from Boortmalt Debre Birhan malt factory. In the factory's laboratory, the domain experts confirm the gathered barley seeds. All sampled seed varieties are the products of the 2022/2023 production year and were taken in the first week of April 2023 the data set was captured using techno spark 5 mobile camera with a resolution 64x 64 pixels and we have used a private dataset of 13030 labeled

images from Boortmalt Debre Birhan malt factory, which contains 4 classes (Holker=3257, Misikal=3257, Propino=3258, Sabini=3258).

3.1.3 Implementation Procedure and Tools

To implement the proposed classification model for the classification of malt barley seed use KNN, SVM, and an ensemble of KNN and SVM deep learning algorithms have been employed and Keras (using Tensor Flow as a backend) have be used. Python programming languages were used for preprocessing, feature extraction, and building a malt barley classification model. The experimentation was conducted using a Computer with an Intel(R) core(TM) i5_3110M CPU of 2.4GHz processor speed and 8.00GB (RAM) 1 TB hard disk capacity with a window platform.

3.1.4 Evaluation Procedure

The performance of a model will be evaluated by comparing its output with the observed data using performance measuring metrics and percentage accuracy measures for each class such as recall, f1-score, and precision values for evaluating diagnostic accuracy. A confusion matrix that shows how many guesses based on known true values that were correct and incorrect will be created. It demonstrates a true positive (TP): the actual value is true, and the model correctly predicted it to be true; true negative (TN): the actual value and the predicted value are both false; False positive (FP): the actual value is false, but the model correctly predicted it to be true; and false negative (FN): the actual value is true, but the model incorrectly predicted it to be false (Khattri et al., 2022a).

It is an $n \times n$ matrix that contains the number of occurrences of actual value and predicted value. Table 2.1 below shows the sample format of the confusion matrix. According to (Ganaie et al., 2022)each metric can be calculated as the formula defined below.

Table 3.1 Sample format of $n \times n$ confusion matrix

Predicted value	Class 1	Class 2	...	Class n
Actual value				
Class 1	X1	X12	X1n

Class 2	X2	X22	...	X2n
.
.
.
Class n	Xn1	Xn2	...	Xnn

Accuracy: A model's accuracy will be determined by how frequently it correctly predicts a value based on the input. However, it doesn't provide specific information about FP and FN. F1-score and recall are crucial in some applications where FP and FN are significant and the formula for accuracy is shown in equation 3.1 below (Kulkarni et al., 2020).

$$\text{Accuracy} = \frac{Tp + TN}{TP + TN + FP + FN} \dots\dots\dots (3.1)$$

Precision: This measurement illustrates the frequency with which a model produces actual positive predictions. The low precision score suggests a high number of false positives and the formula for calculating precision is shown in equation 3.2 below (Khatri et al., 2022a).

$$\text{Precision} = \frac{TP}{TP + FP} \dots\dots\dots (3.2)$$

Recall: this parameter provides information about how often a model predicts false negatives. The low value of recall means the model predicted high false negatives and the formula for calculating recall is presented in equation 3.3 below (Khatri et al., 2022b).

$$\text{Recall} = \frac{TP + TN}{TP + FN} \dots\dots\dots (3.3)$$

F1-score: Precision and recall are combined to determine the F1-score. In other words, a high F1 score suggests that there are few false positives and false negatives, which suggests that the models are accurate in identifying real threats and are unaffected by false alarms the formula for calculating the F1 score is shown in equation 3.4 below (Khatri et al., 2022a).

$$\text{F1 - score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \dots\dots\dots (3.4)$$

3.2 Research Methodology Approaches

To create the artifact for this study, we will employ design science research methodology approaches are used (identify the problem, define the objective, design and development, demonstration, evaluation, and communication)(Peffer et al., 2007). Below is a description of the research's methodological approaches for the proposed thesis.

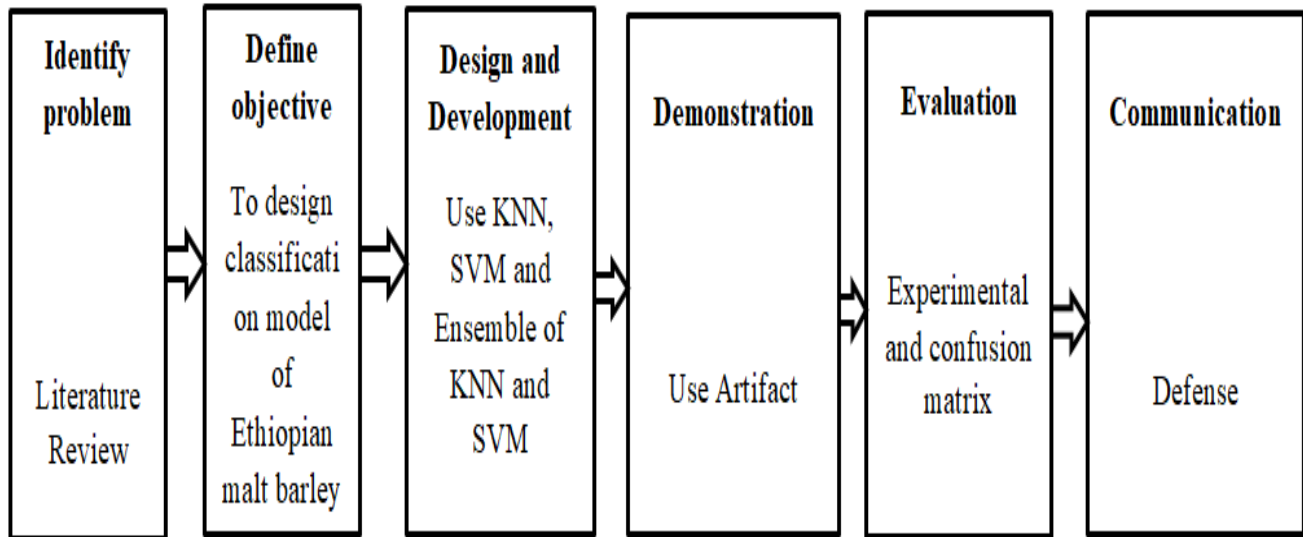


Figure 3.1 Design science research methodology (Peffer et al., 2007)

3.3 Design of Malt Barley Classification Model

The focus of this study is to design a malt barley seed image classification model for Ethiopian malt barley seed. The digitalized barley seed images are first preprocessed to eliminate noise, convert to grayscale, and find edges. After the image is preprocessed, the next step is segmenting the image into a set of homogeneous and meaningful regions with a set of identical properties. After this stage, the segmented images are delivered into the feature extraction stage. Finally, the identified malt barley seeds are aligned to a varietal class that it belongs to. To identify malt barley varieties a series of steps are required. The details of these steps, the main focuses of this study, are depicted in Figure 3.2 below.

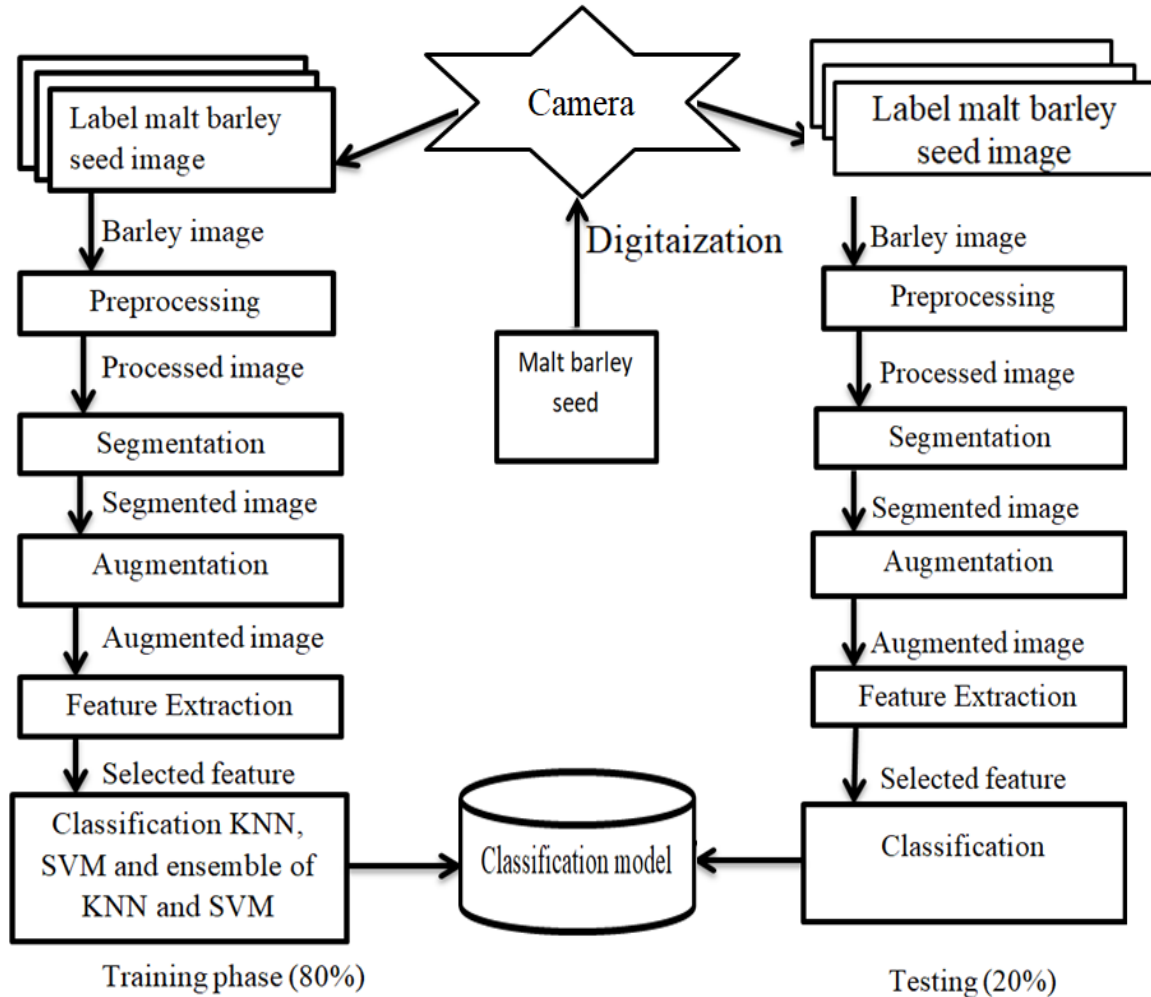


Figure 3.2 Architecture of the proposed model for Ethiopian malt barley classification

3.3.1 Image acquisition

Image acquisition is the first step in digital image processing, which is the major approach used to evaluate the image of malt barley using a computer (*What Is Image Acquisition in Image Processing - Search*, n.d.).

Digitalization makes it possible to create a digital image of the scanned seeds for seed classification in the JPG (Joint Photographers Expert Groups) file format and Techno spark 5 cameras were used to acquire images from malt barley seeds.

For this study, images of four malt barley seed (Holker, Propino, Misikal, and Sabini) from Boortmalt Debre Birhan malt factory, Ethiopia, are considered. The varieties of the sampled

barley seeds were confirmed by specialists at the factory's laboratory upon acquisition. The samples were collected in the first week of April 2023, and every kind of seed examined was a product of the production year 2022/2023.

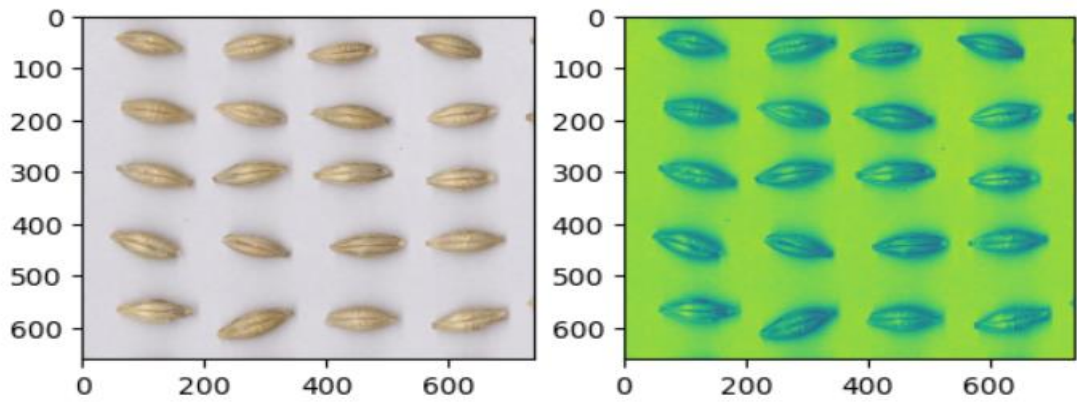
3.3.2 Image Preprocessing

The accuracy of identification models highly depends on the effectiveness of the preprocessing stages. The main goal of preprocessing, from the point of view of this research, is to reduce the noise from the image for further analysis of the next phases. The preprocessing stage is a collection of operations that apply successive transformations on an image to modify and prepare the pixel values of the digitized image for subsequent operations. This includes gray-scale conversion, binarization, edge detection, and noise removal of malt barley seed image, and the details are discussed in the following sub-sections.

3.3.2.1 Grayscale Conversion

At this stage, we convert the color image of the digitized malt barley image into grayscale (an image that has only a one-bit value). A color image is a red, green, and blue (RGB) image that is saved as an m-by-n-by-3 data array and identifies the color components for each pixel. Images in grayscale have pixel values between 0 and 255. The conversion process of a color image to grayscale is done using a Payton which is called `rgb2gray ()` and it works by forming a weighted sum of the R, G, and B components, as given in equation 3.5 below (*Convert RGB Image to Grayscale - MATLAB Im2gray*, n.d.).

$$0.2989 * R + 0.5870 * G + 0.1140 * B \dots\dots\dots (3.5)$$



(a)

(b)

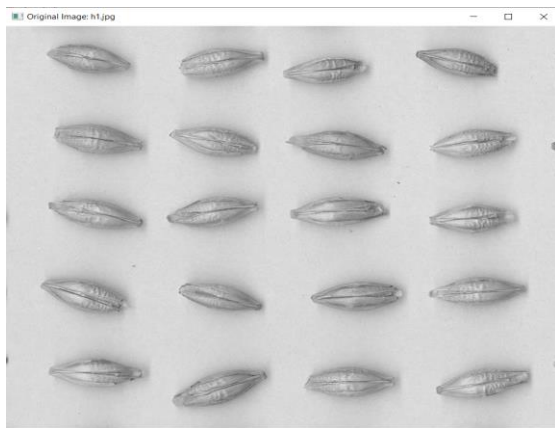
Figure 3.3 Sample scanned Holker malt barley seed; (a) RGB image, (b) Gray level image

3.3.2.2 Noise Removal/ Denoising

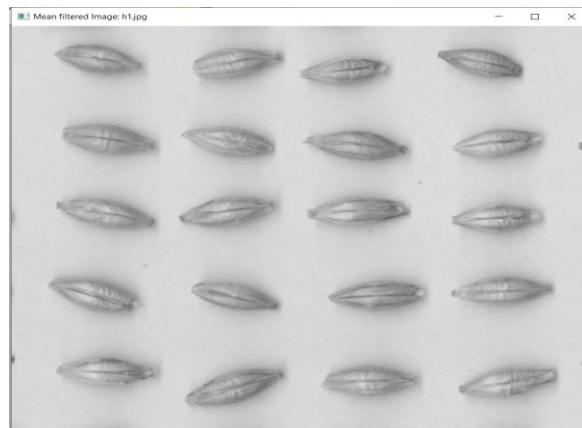
Noise occurs due to the random changes in the values of pixels in the image. In digital images, we may face a variety of noise removal types. Those are the Mean filter; Gaussian filter and Median filter are some of the common methods to reduce such noises (*Techniques & Methods For Image Filtering & Noise Reduction / Talentica Blog*, n.d.). Image denoising is a technique for improving image quality by maintaining as much of the original information as feasible while removing useless information from the signal. In practice, if denoising is not done properly, it can result in severe risk and harm. The size and position of image noise are uncertain, although the image and noise are frequently connected. The camera's electrical signal, for example, is related to noise; the dark area has noise, while the light part has less.

3.3.2.2.1 Mean filtering

Mean filtering is a type of non-linear spatial filter that replaces a pixel's value with the mean of the gray levels in the neighborhood of the pixel, which has the dual benefits of removing random noise and impulse interference while also retaining a significant amount of image edge information. To replace the gray pixel in the middle of the windows with a mean value of a pixel, it chooses a specific type of window, moves it over each point of the image, and then selects another window type(Debayle & Pinoli, 2011).



(a)

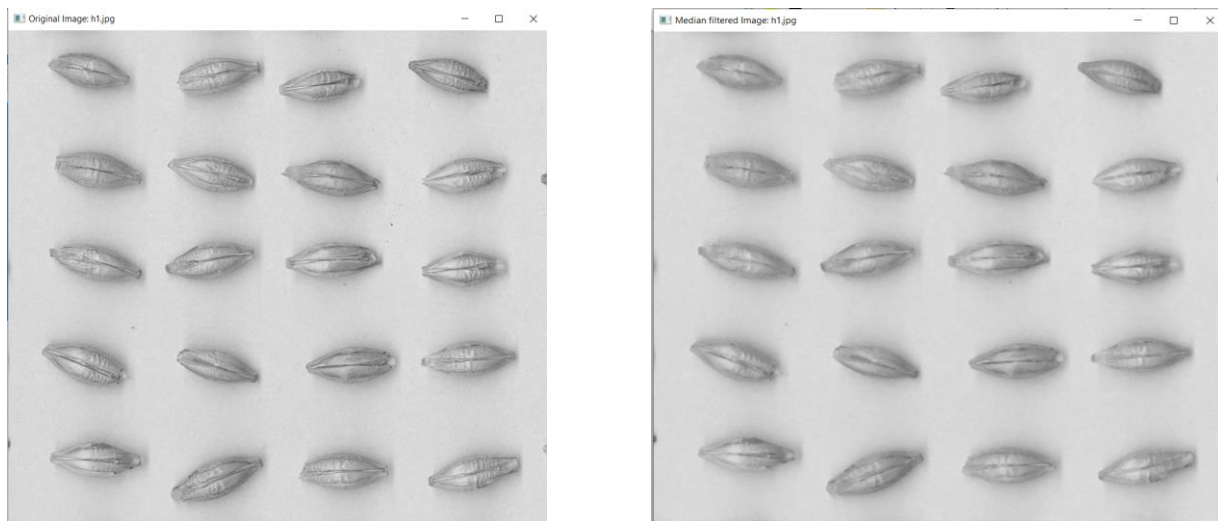


(b)

Figure 3.4 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Mean filtering images

3.3.2.2.2 Median Filter

In image processing, the Median filter is a non-linear technique known as order statistic filtering. It is a popular method for removing noise because it runs through the signal cell by cell, replacing the value of each cell with the neighboring by a median of the intensity levels with mathematical precision. The median value of the corresponding window is substituted for the center pixel of an M neighbor. Because the technique is applied across the image, it tends to transform both noisy and denoised pixels in the image. Because of the median filter's tendency, good pixel cells are replaced by corrupted ones. As a result, denoising frequently results in the removal of fine details from an image because it is done at the expense of the distorted and blurred features that this filtering technique possesses (Debayle & Pinoli, 2011).



(a)

(b)

Figure 3.5 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Median filtering images

3.3.2.2.3 Gaussian Filter

The Gaussian filter has a non-uniform low pass. The Gaussian filter is used to remove Speckle Noise from ultrasound and MRI images. The average value of the surrounding pixel or

neighboring pixels replaces the noisy pixel present in the image, which is based on a Gaussian distribution (Debayle & Pinoli, 2011).



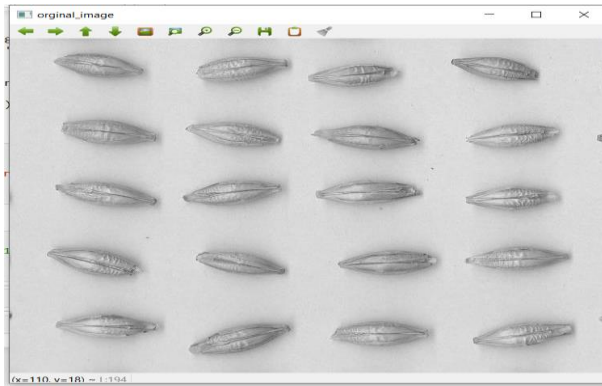
(a)

(b)

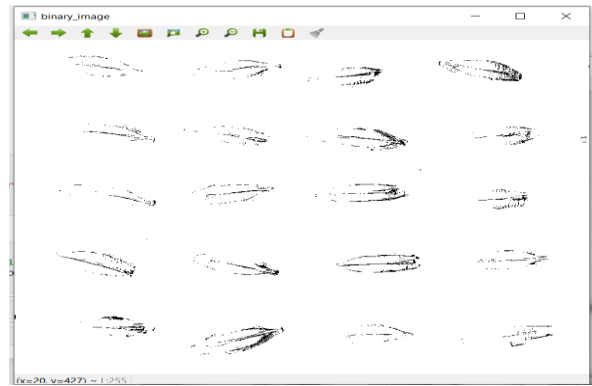
Figure 3.6 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of Gaussian filtering images

3.3.2.3 Binarization

We converted the filtered grayscale malt barley seed image into binary (0s and 1s) after using a filtering algorithm to reduce noise and improve the image quality. Binarization is the process of converting either a color image or a grayscale image, to a bi-level image through which each pixel is categorized either as a foreground or a background pixel and it is an important preprocessing method in image analysis and boundary detection (Sauvola & Pietikäinen, 2000). Image binarization or thresholding is used to distinguish the content from the background. During the process of binarization, image pixels are classified into two values one and zero, which represent foreground and background classes respectively (Sauvola & Pietikäinen, 2000).



(a)

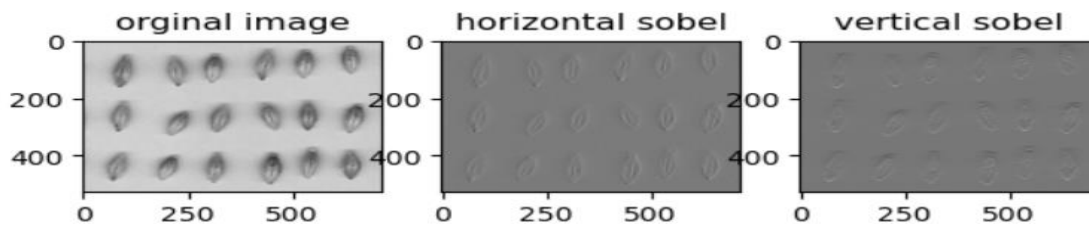


(b)

Figure 3.7 Sample scanned Holker malt barley seed; (a) Original RGB image, (b) Result of binary images

3.3.2.4 Edge detection

Edges are the boundaries between segments. Edge detection refers to the overall problem of locating, sizing, and segmenting borders in a picture (*Edge Enhancement - Wikipedia*, n.d.). Edge detection involves finding the edge pixels, whereas edge enhancement involves making the edges and background more contrastive (*Edge Detection*, n.d.). The Sobel edge detection method is applied in this situation. A 3 x 3 neighborhood-based gradient operator called Sobel edge detection employs both vertical and horizontal masks. Then, by using these two masks independently on the input image, we may produce two gradient components, G_x and G_y , for the horizontal and vertical axes.



(a)

(b)

(c)



(d)

Figure 3.8 Sample scanned Holker malt barley seed; (a) original RGB image, (b) Result of horizontal Sobel, (c) Result of horizontal sobel (d) Result of edge detected image

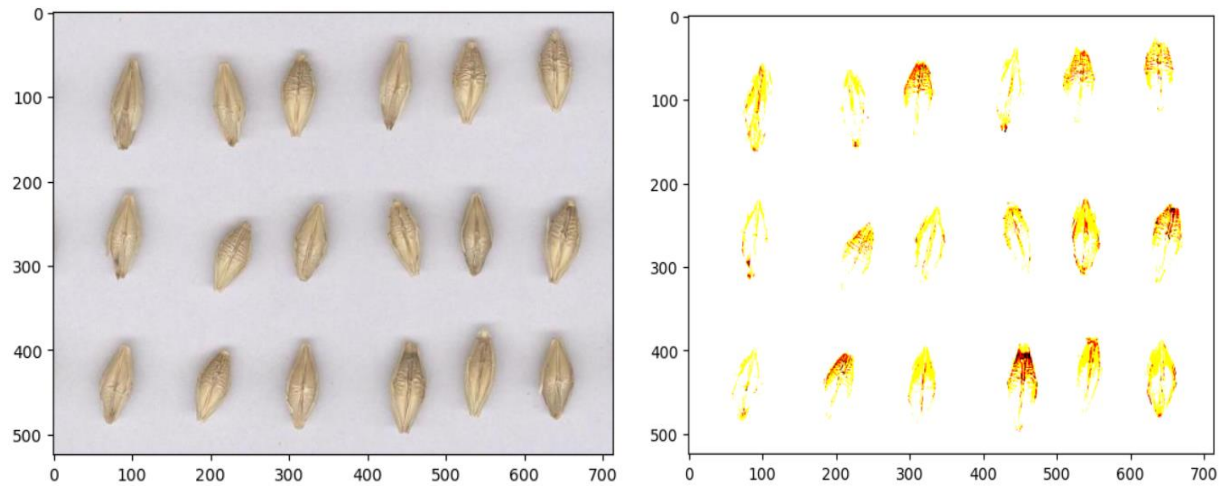
3.3.3 Image Segmentation

Image segmentation is the technique of dividing a digital image into various image segments in digital image processing. Image segmentation seeks to simplify or transform an image's representation into something more significant and understandable (*Image Segmentation - Search*, n.d.). Watershed segmentation and Otsu thresholding segmentation are two methods for segmenting images. Thresholding method is the simplest method of image segmentation which is based on a clip level (or a threshold value) to change a gray-scale image into a binary image.

For segmenting images of malted barley, we suggested Otsu thresholding and watershed segmentation approaches in this work. We originally used the Otsu thresholding approach for that purpose before moving on to the watershed segmentation method.

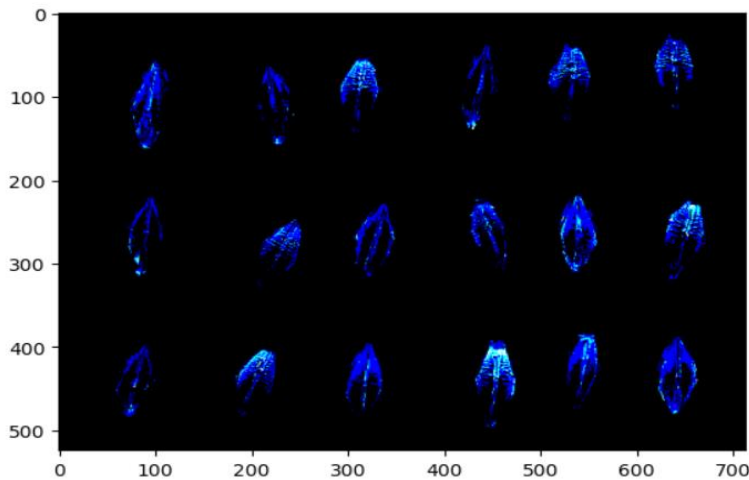
3.3.3.1 Otsu thresholding

Using this technique, the original color image can be automatically thresholded using clustering to create a binary or grayscale version. By checking each threshold value between 0 and 255, it determines the ideal threshold value. To create a binary image, it assumes that the input image contains two classes of pixels, known as background pixels and foreground pixels, and determines the best threshold value for each class by minimizing intra class intensity and maximizing inter class variance. (*Otsu Thresholding—Image Binarization | by HBY Coding Academic | Medium*, n.d.).



(a)

(b)



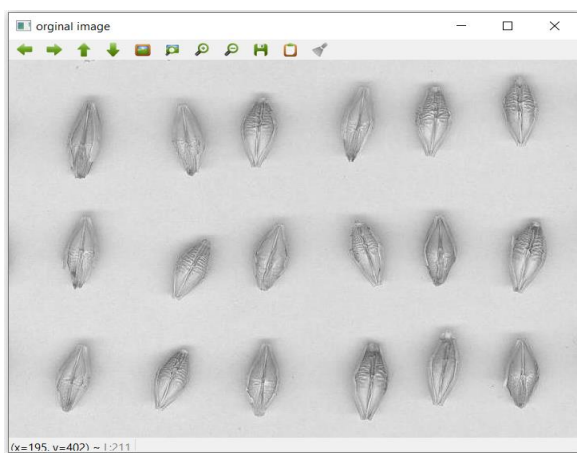
(c)

Figure 3.9 Malt barley image segmentation using Otsu thresholding; (a) Original image, (b) Result of forward segmentation using Otsu, (c) Result of reverse segmentation using Otsu.

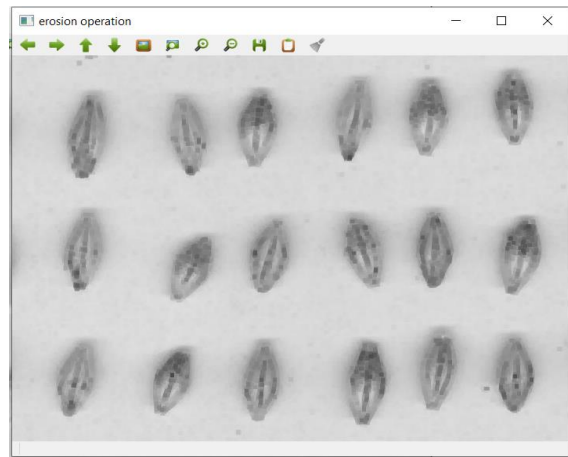
3.3.3.2 Watershed segmentation

In image acquisition, the barley is randomly distributed which leads to overlapping between two or more seeds. In Otsu segmentation, we are unable to isolate each seed from the image when the seeds are touched or overlapped with each other. To overcome this problem, we have used a classical algorithm called watershed segmentation after Otsu segmentation to extract touched or overlapped barley in the image. Using this algorithm, we build barriers in the locations where

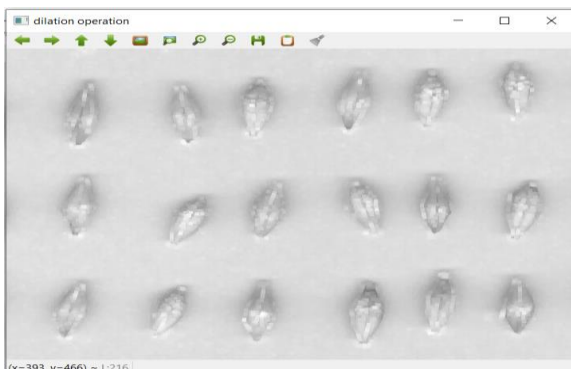
barley is merged. Before applying watershed segmentation to the image, we use morphological opening and morphological closing operations in the image. To implement morphological opening and morphological closing operation in the image before we can implement erosion and dilation operation. During Otsu segmentation, there may be small noises in the image, in this case, the morphological opening operation is used to remove small white noise in the image, and the morphological closing operation is used to remove small holes in the object(*Watershed Segmentation — Skimage 0.21.0 Documentation*, n.d.).



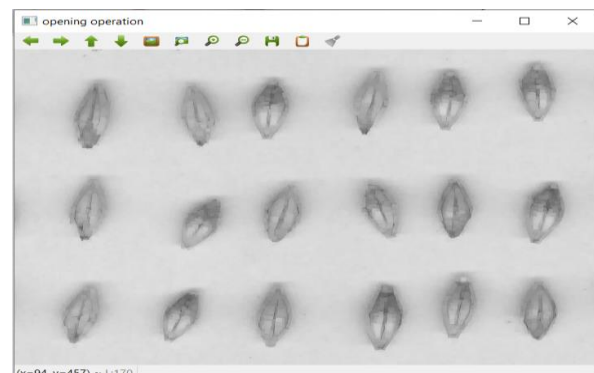
(a)



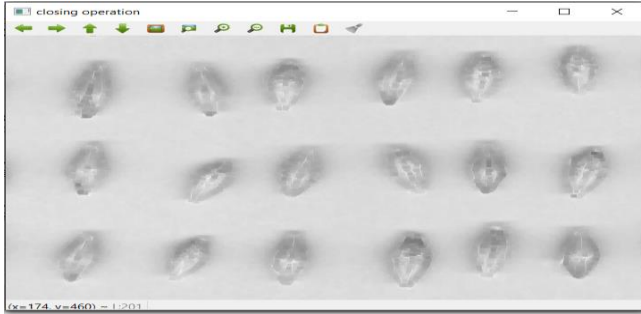
(b)



(c)



(d)



(e)

Figure 3.10 Malt barley image s Watershed segmentation; (a) Original image, (b) Result of erosion segmentation, (c) Result of dilation segmentation, (d) Result of opening operation, (e) Result of the closing operation

3.3.4 Image Augmentation

We use data offline augmentation to increase the training dataset and reduce over fitting due to a lack of training data. Because it is a deep learning classification, a large dataset is required to achieve the best performance of the model. The issue of dataset scarcity in this paper by using, Image Data Generator of rotation, shifting, add brightness, flip and zoom augmentation methods to solve a shortage of dataset(*What Is Image Augmentation - Alumentations Documentation*, n.d.).

3.3.5 Dataset Splitting

We split our data into 80:20 ratios using a hold-out technique, first, we cut and use 20% for testing then we use the remaining 80 % for the training set second, and we can split the training dataset into 80% for training and 20% for validation. The total number of dataset is 13030 and the number of dataset for training and testing it use is 10424 and 2606 respectively.

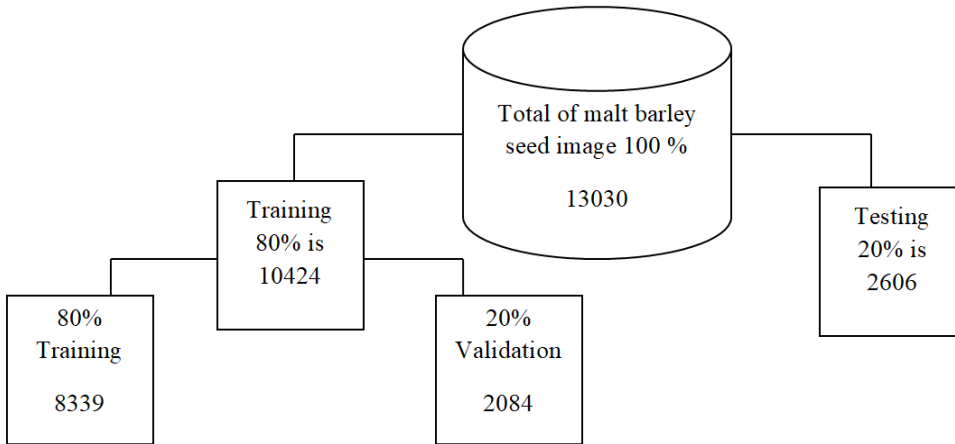


Figure 3.11 Dataset split distribution

3.3.6 Feature Extraction

Feature extraction is the process of retrieving meaningful information from an image that is used for the identification /classification of images into different categories(*What Is Feature Extraction - Search*, n.d.). The feature of images and the extraction method decide the nature and the output of the image-preprocessing and segmentation steps. As far as the format of the extracted features matches the requirements of the classifier, the image features and extraction methods work on color images, gray level, or binary images. Therefore, the currently proposed identification approach, which takes into account an evaluation of human visual inspection and physical features of each malt barley seed as starting point, is based on morphology, texture, and color analysis. We employ three Python-based feature extraction techniques morphology, texture, and color to extract features.

3.3.6.1 Morphological Feature Extraction

Morphology features a wide range of image processing operations that analyze images based on shapes. Each pixel in the image is modified during a morphological operation based on the values of the pixels nearby. Morphology features are the geometric properties of images, such as area, height, width, perimeter, maximum diameters, minimum diameters, and others (*Understanding Morphological Image Processing and Its Operations | by Prateek Chhikara | Towards Data Science*, n.d.).

3.3.6.2 Texture Feature Extraction

Texture feature is a basic attribute of an image is its texture and determined by measurements of its smoothness, coarseness, and regularity of pixel structure. Energy, entropy, contrast, and homogeneity are among the useful data that can be obtained by image segmentation, classification, and interpretation(HAILU, 2020).

3.3.6.3 Color Feature Extraction

Color feature extraction is a method for extracting features from an image based on its color. It is one of the three main categories of image features. To extract color features from an image, the color feature extraction technique uses color spaces like RGB(Zhang, 2019).

3.3.7 Classification

The two fundamental steps in the classification process are testing and training. In the training step, the classifier learns all extracted features and the relationships between each sample in the labeled sample class. In the testing phase, the trained system is applied to data that it has never seen before to evaluate the classification's effectiveness. The majority of researchers aim to create classifiers with small test errors. Different types of pattern classification techniques have been used in machine learning for the classification of patterns. These classification methods are mainly categorized as supervised learning and unsupervised learning type.

In supervised classification, the classifier is trained using a set of training pattern samples, and the resulting new set can be classified as a member of a known class. In contrast, in unsupervised classification, the system partitions or clusters the entire data set based on some similarity criteria, and each cluster of patterns belongs to a particular class(*Supervised vs. Unsupervised Learning: What's the Difference? | Indeed.Com*, n.d.).

It is obvious that no single strategy to classification is ideal in many of the new applications. As a result, mixing different classifiers is now a good classification technique. The K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and an ensemble of the two approaches are the proposed classification algorithms used in the thesis.

3.3.7.1 K-Nearest Neighbor (KNN) Classifier

A data categorization technique known as the k-nearest-neighbor predicts the likelihood that a data point will belong to one of two groups based on which group the data points that are closest to it belong to (Ganaie et al., 2022).

The test sample is categorized according to the class that has received the most support from the sample's closest K neighbors, with the sample being put into that class. The trial-and-error method, which has the lowest error rate, is used to determine the positive integer K. The sample is then only given the class of its closest neighbor.

So, Classification model is built using KNN, test results show that after the digital image preprocessing step is involved the accuracy of KNN is 98.54%.

3.3.7.2 Support Vector Machine (SVM) Classifier

SVM has proven to be effective for a variety of classification problems. It tries to find the best hyperplane among classes by counting the number of points on the edge of the class descriptors. The margin is the distance between classes. When there is a larger margin of error, the classification accuracy improves. The data points along the border are referred to as support vectors. SVM can be used to solve both regression and classification problems. This method works well for solving problems with linear and nonlinear datasets. For a classification model, the SVM algorithm employs various kernel types such as linear radial basis function (RBF), Polynomial, and Sigmoid. SVM searches for features in high-dimensional space and chooses the best hyperplane for categorizing data points into two classes. It is effective for both small and large datasets that cannot be handled (Smola & Schölkopf, 2004).

So, Classification model is built using SVM test results show that after the digital image preprocessing step is involved the accuracy of SVM is 98.15%.

3.3.7.3 Ensemble of K-Nearest Neighbor and Support Vector Machine

Ensemble methods are learning algorithms that are constructed from a set of classifiers. A single classification model, in most of the emerging applications, doesn't behave efficiently due to this it is difficult to build a machine learning device, one which would give the best possible answer every time. A classification system that aims at utilizing the best individual classifier has some

drawbacks and some classifiers work better in some kinds of applications in the presence of specific sorts of noise. Hence, multiple classification algorithms have to be combined giving results to ensemble models which have relatively better classification accuracy(Ganaie et al., 2022). It is very difficult to identify the best classifier unless doing deep experimentation on different data sets or having prior know-how about it. Using different feature sets and different classification techniques, classification systems have different performances. To enhance the classification performance of a system, one can use a set of individual classifiers and combiner to make the final decision. Multiple classifiers can be used in several ways to enhance the system's performance. Each classifier can be trained in a different region of feature space and can provide probability estimation and decision-based on the analysis of individual results. There exist numerous methods for model combination; the linear combiner, the product combiner, and the voting combiner are the most commonly used methods. The classification performance of multiple classifiers not only depends on the combination strategy, but also relies on the competency, complementariness, and the number of base classifiers. By analogy, when designing an ensemble is the same as when establishing a committee of people; a number of committee members may differ according to the duty that they composed for and each member of the committee should be as competent as possible, but the members should be complementary to one another. If they always agree, then the committee is unnecessary, any single member is sufficient. If the members are complementary, then when one or a few members make an error, the probability is high that the remaining members can correct this error. Classification of pixels in an image and identifying their relevant class mainly depends on the feature extraction and classifier selection process. The K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and an ensemble of K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) classifier models are built to identify the type of malt barley seed images. During ensemble learning, instead of generating single models, the ensemble passes the training set to each of its multiple base models, obtains their classification, and then combines them in some appropriate manner(Ganaie et al., 2022).

So, Classification model is built using Ensemble of KNN and SVM, test results show that after the digital image preprocessing step is 98.58%.

3.4 Summary

In this chapter, we have focus on research methodology, research methodology approach, and the detail of malt barley classification model.

CHAPTER FOUR

4. Experimentation

4.1 Introduction

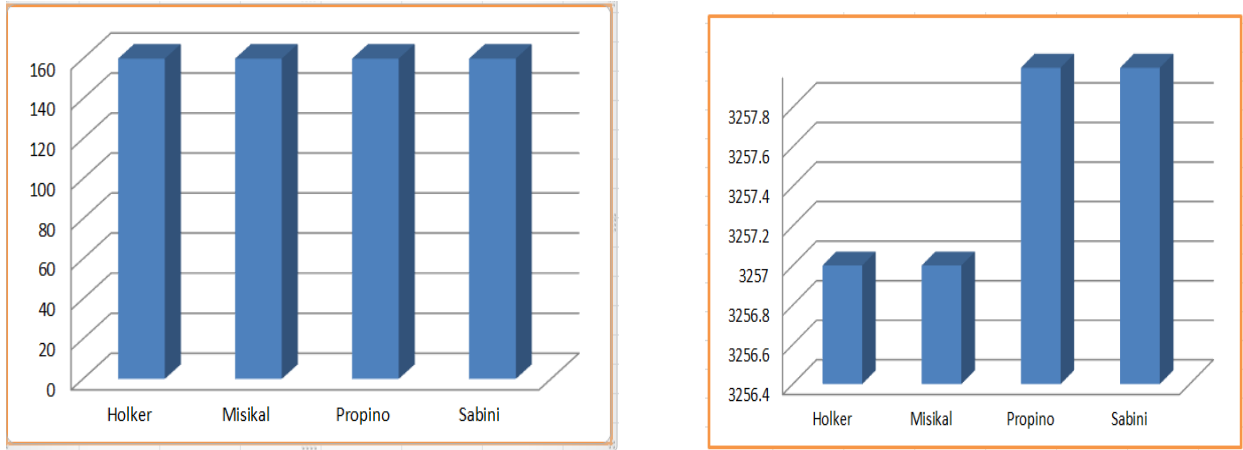
The complete experimental process is covered in this chapter, from beginning to end. The experimental setup, dataset preparation, preprocessing experiment, model-building experiment, and result evaluation during the training and test phase will all be explained in great depth.

4.2 Dataset Preparation

Since there is no readymade dataset for this type of research, we have prepared our dataset for training and performance evaluation. To do this, malt barley seeds have been collected from Boortmalt Debre Birhan malt factory. In the factory's laboratory, the domain experts confirm the gathered barley seeds. All sampled seed varieties are the products of the 2022/2023 production year and were taken in the first week of April 2023 the data set was captured using techno spark 5 mobile camera with a resolution 64x 64 pixels and we have used a private dataset of 13030 labeled images from Boortmalt Debre Birhan malt factory, which contains 4 classes (Holker=3257, Misikal=3257, Propino=3258, Sabini=3258). To evaluate the performance of the model, 80 percent of the dataset is used to training (80% train and 20% validate) the model, and the rest 20 percent of the dataset is categorized randomly as testing. The details quantities of sample malt barley seeds used in our experimentation are summarized in Table 4.1 below.

Table 4.1 An overview of the malt barley seed image that was taken for the current study

No	Variety Name	Data source	Special Resolution	Image format	Number of malt barely seed
1	Holker	Boortmalt Debre Birhan factory	64x 64	JPG	3257
2	Misikal	Boortmalt Debre Birhan factory	64x 64	JPG	3257
3	Propino	Boortmalt Debre Birhan factory	64x 64	JPG	3258
4	Sabini	Boortmalt Debre Birhan factory	64x 64	JPG	3258



Graph 4.1 Dataset before augmentation (left) and after augmentation (right)

4.3 Building Proposed Model

As described in Figure 4.6 below, our Classification model is constructed by feeding the training data set which was partitioned in the data set partitioning phase. To build the model we have used K-Nearest Neighbors (KNN), Support Vector Machine, and the ensembles of these two classification algorithms.

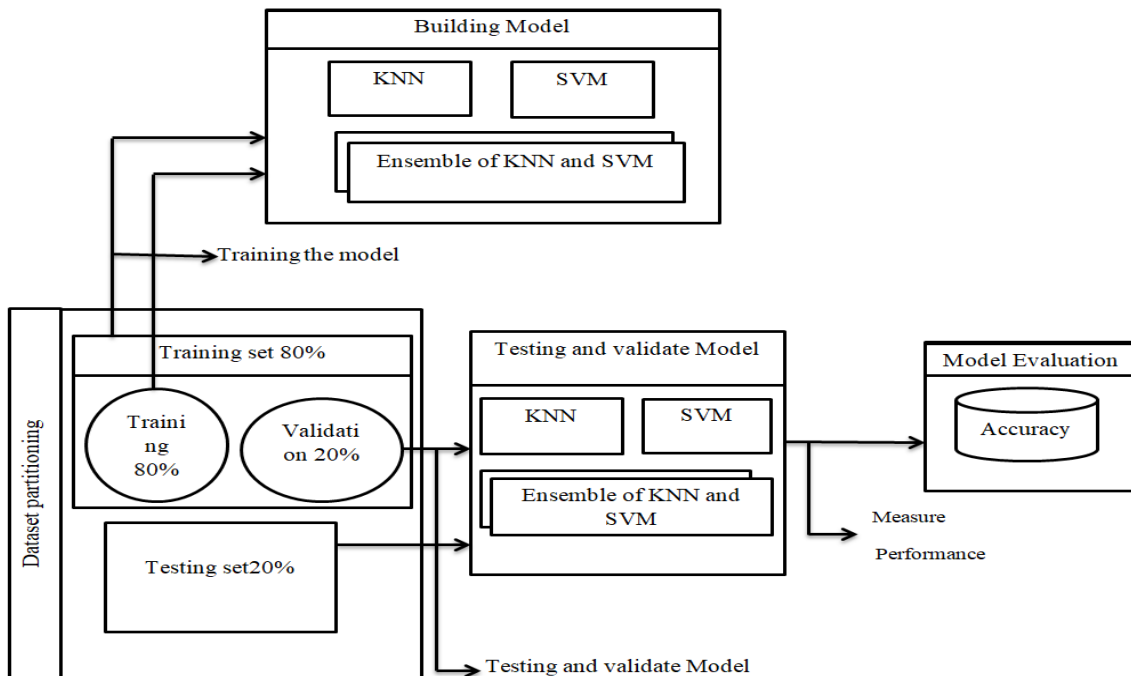


Figure 4.1 Proposed approaches for MBCM

4.4 Experimental Setup

The model was developed using Python language on a Lenovo laptop with 8 GB of RAM, 1 TB hard drive, 2.40 GHz Intel Core i5 processor, and 10th generation, with TensorFlow and the Kears API at the backend. We used 64×64 -pixel size images of 13030 images with 4 label classes for our experiment. The dataset was split 80/20. The performance of our proposed model is evaluated using the accuracy, precision, recall, f1-score, confusion matrices, training, validation, and testing values.

Table 4.2 Summary of Experimental Setup

No	Name	Description
1	Development tools	Language-python Backend -TensorFlow Api-keras Os-windows 10 Notebook-jupyter Pc=8GB RAM 1TB HHD 2.4Ghz, core=i5 and 10 th generation
2	Dataset	Total dataset=13030 Format=jpg Size=64x64 Split: 80:20 Number of classes=4
3	Digital image preprocessing	Image Preprocessing=gray scale, noise removal, binarization, and edge detection Filter technique=Gaussian, mean, median. Segmentation=otsu threshold and Watershed segmentation
4	Classifier	KNN,SVM and ensemble of KNN and SVM
5	Performance metrics	accuracy, precision, recall, f1-score, confusion matrices

4.5 Experimental Result and Discussion

4.5.1 Experiment result before preprocessing

Before applying the preprocessing technique, we conducted experiments utilizing a classifier that used our proposed models, namely the KNN, SVM, and ensemble of KNN and SVM. When we run our experiment, 64x64 pixels size image was used for the training, validation, and test dataset.

Table 4.3 Experiment result before dataset preprocessing

Name	Classifier	Testing Accuracy %
	KNN	47.61%
Experiment result before preprocessing	SVM	54.76%
	Ensemble of KNN and SVM	57.14%

4.5.1.1 Confusion matrices for KNN before Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.2 represent the result of a confusion matrix of the KNN classifier before a preprocessing step is performed(*What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com*, n.d.).

```
Confusion_matrix for KNN before preprocessing is:  
[[8 4 1 4]  
 [0 5 1 2]  
 [2 0 5 4]  
 [1 3 0 2]]
```

Figure 4.2 Confusion matrixes for KNN before preprocessing

4.5.1.2 Classification report KNN before Preprocessing

Below figure 4.3 are a result of KNN before a preprocessing step is involved and represents metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.73	0.47	0.57	17
Misikal	0.42	0.62	0.50	8
Propino	0.71	0.45	0.56	11
Sabini	0.17	0.33	0.22	6
accuracy			0.48	42
macro avg	0.51	0.47	0.46	42
weighted avg	0.58	0.48	0.50	42

Figure 4.3 Precision, recall, and f1-score using KNN classifier before preprocessing

4.5.1.3 Confusion matrices for SVM before Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.4 represent the result of a confusion matrix of the SVM classifier before a preprocessing step is performed(*What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com*, n.d.).

```
Confusion_matrix for SVM before preprocessing is:  
[[10  0  1  3]  
 [ 1  6  1  0]  
 [ 6  0  4  0]  
 [ 2  5  0  3]]
```

Figure 4.4 Confusion matrixes for SVM before preprocessing

4.5.1.4 Classification report SVM before Preprocessing

Below figure 4.5 are the result of SVM before a preprocessing step is involved and represents metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.53	0.71	0.61	14
Misikal	0.55	0.75	0.63	8
Propino	0.67	0.40	0.50	10
Sabini	0.50	0.30	0.37	10
accuracy			0.55	42
macro avg	0.56	0.54	0.53	42
weighted avg	0.56	0.55	0.53	42

Figure 4.5 Precision, recall, and f1-score using SVM classifier before preprocessing

4.5.1.5 Confusion Matrices for Ensemble before Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.6 represent the result of the confusion matrix of the Ensemble classifier before a preprocessing step is performed(*What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com*, n.d.).

```
Confusion_matrix for Ensemble before preprocessing is is:
[[14  3  1  2]
 [ 1  0  0  1]
 [ 3  3  3  1]
 [ 1  2  0  7]]
```

Figure 4.6 Confusion matrixes for ensemble before preprocessing

4.5.1.6 Classification Report Ensemble before Preprocessing

The below figure 4.7 are a result of SVM before the preprocessing step is involved and represents metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.74	0.70	0.72	20
Misikal	0.00	0.00	0.00	2
Propino	0.75	0.30	0.43	10
Sabini	0.64	0.70	0.67	10
accuracy			0.57	42
macro avg	0.53	0.42	0.45	42
weighted avg	0.68	0.57	0.60	42

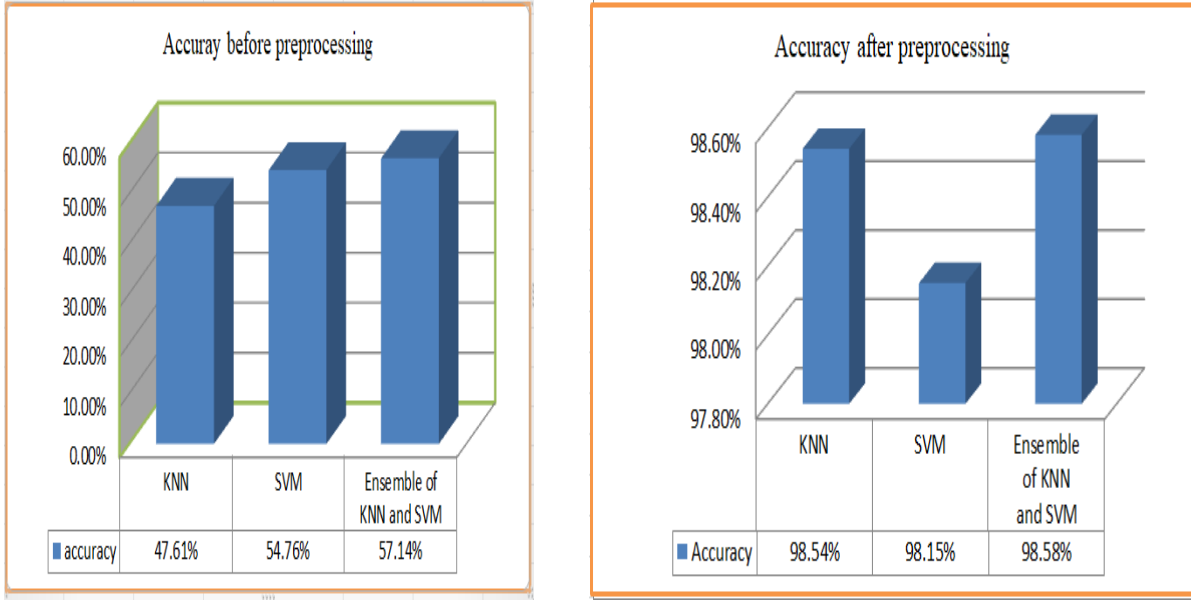
Figure 4.7 Precision, recall, and f1-score using ensemble classifier before preprocessing

4.5.2 Experiment Result after Preprocessing

After preprocessing and applying a classifier, we augment the images and classify them using KNN, SVM, and an Ensemble of KNN and SVM and the results are in Table 4.4 below.

Table 4.4 Experiment result for classifiers after preprocessing

Name	Classifier	Testing Accuracy %
	KNN	98.54%
Experiment result after preprocessing	SVM	98.15%
	Ensemble of KNN and SVM	98.58%



Graph 4.2 Accuracy before preprocessing (left) and after preprocessing (right)

4.5.2.1 Confusion matrices for KNN after Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.8 represent the result of a confusion matrix of the KNN classifier after a preprocessing step is performed(What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com, n.d.).

```
Confusion_matrix for KNN learning is:
[[741  6  8  1]
 [  0 665  0  0]
 [  0  4 681  0]
 [  2  8  5 485]]
```

Figure 4.8 Confusion matrixes for KNN after preprocessing

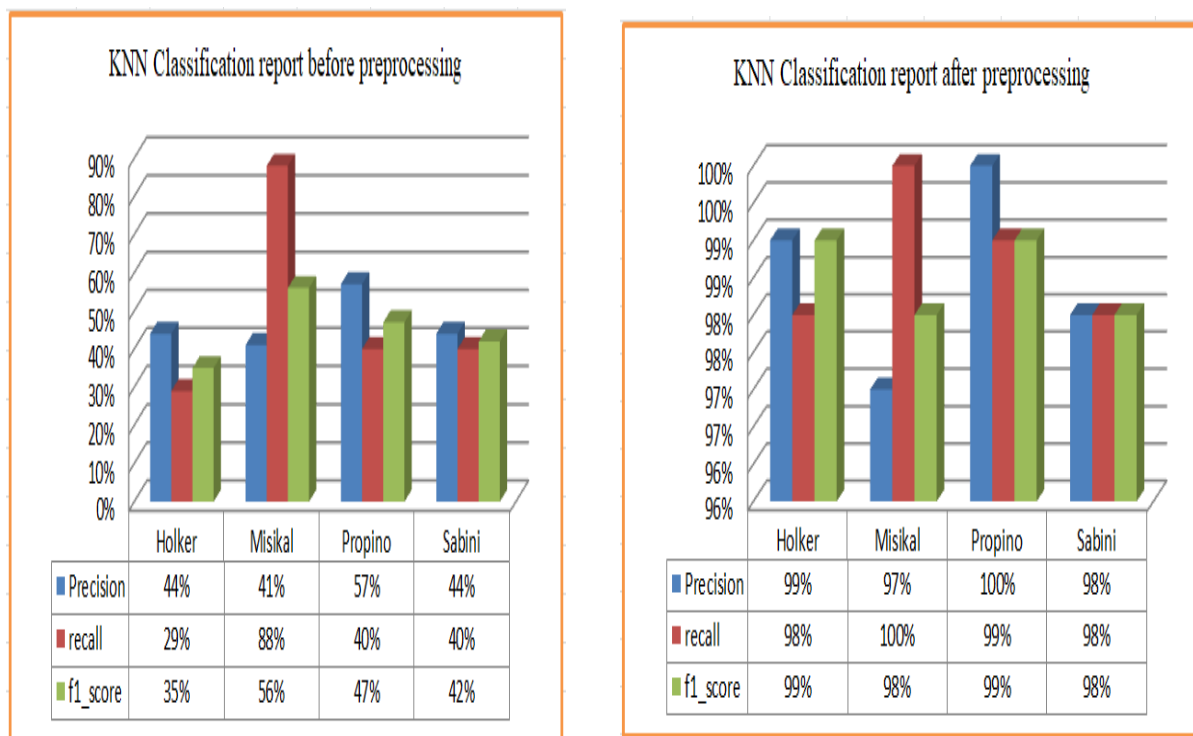
4.5.2.2 Classification Report for KNN after Preprocessing

The below figure 4.9 are the result of KNN after preprocessing, segmentation, and augmentation step are involved and represent metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.99	0.98	0.99	784
Misikal	0.97	1.00	0.98	649
Propino	1.00	0.99	0.99	666
Sabini	0.98	0.98	0.98	507
accuracy			0.99	2606
macro avg	0.99	0.99	0.99	2606
weighted avg	0.99	0.99	0.99	2606

Figure 4.9 Precision, recall, and f1-score using KNN classifier after preprocessing

4.5.2.3 Graph of Classification Report for KNN Classifier



Graph 4.3 Graph of precision, recall, and f1-score using KNN classification before preprocessing (left) and after preprocessing (right)

4.5.2.4 Confusion matrices for SVM after Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.10 represent the result of a confusion matrix of the SVM classifier after a preprocessing step is performed(*What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com*, n.d.).

Confusion_matrix for SVM learning is:

```
[[735 11  7  3]
 [ 6 653  0  6]
 [ 1  0 683  1]
 [ 6  5  2 487]]
```

Figure 4.10 Confusion matrixes for SVM after preprocessing

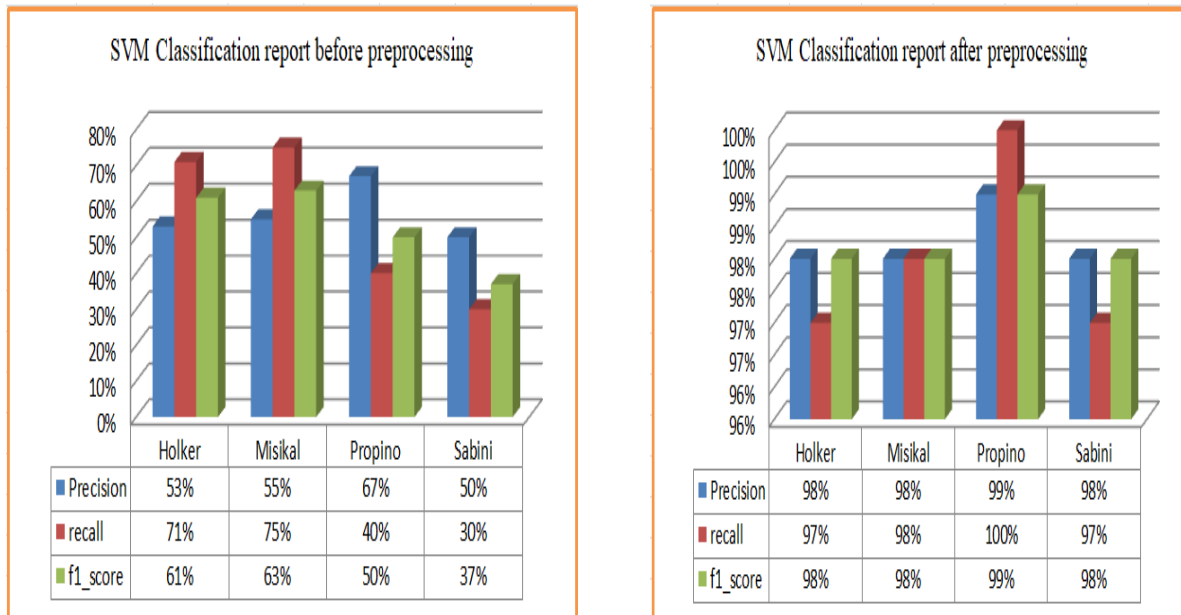
4.5.2.5 Classification Report for SVM after Preprocessing

Below figure 4.11 are the result of SVM after preprocessing, segmentation, and augmentation step are involved and represent metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.98	0.97	0.98	756
Misikal	0.98	0.98	0.98	665
Propino	0.99	1.00	0.99	685
Sabini	0.98	0.97	0.98	500
accuracy			0.98	2606
macro avg	0.98	0.98	0.98	2606
weighted avg	0.98	0.98	0.98	2606

Figure 4.11 Precision, recall, and f1-score using SVM classifier after preprocessing

4.5.2.6 Graph of Classification Report for SVM Classifier



Graph 4.4 Graph of precision, recall, and f1-score using SVM classification before preprocessing (left) and after preprocessing (right)

4.5.2.7 Confusion Matrices for Ensemble after Preprocessing

The confusion matrix is a technique for summarizing the performance of a classification algorithm and the below figure 4.12 represent the result of a confusion matrix of the Ensemble classifier after the preprocessing step is performed(*What Is a Confusion Matrix in Machine Learning - MachineLearningMastery.Com, n.d.*).

Confusion_matrix for Ensemble learning is:

```
[[749  4  3  0]
 [ 6 659  0  0]
 [ 1  4 680  0]
 [ 8 11  7 474]]
```

Figure 4.12 Confusion matrixes for ensemble after preprocessing

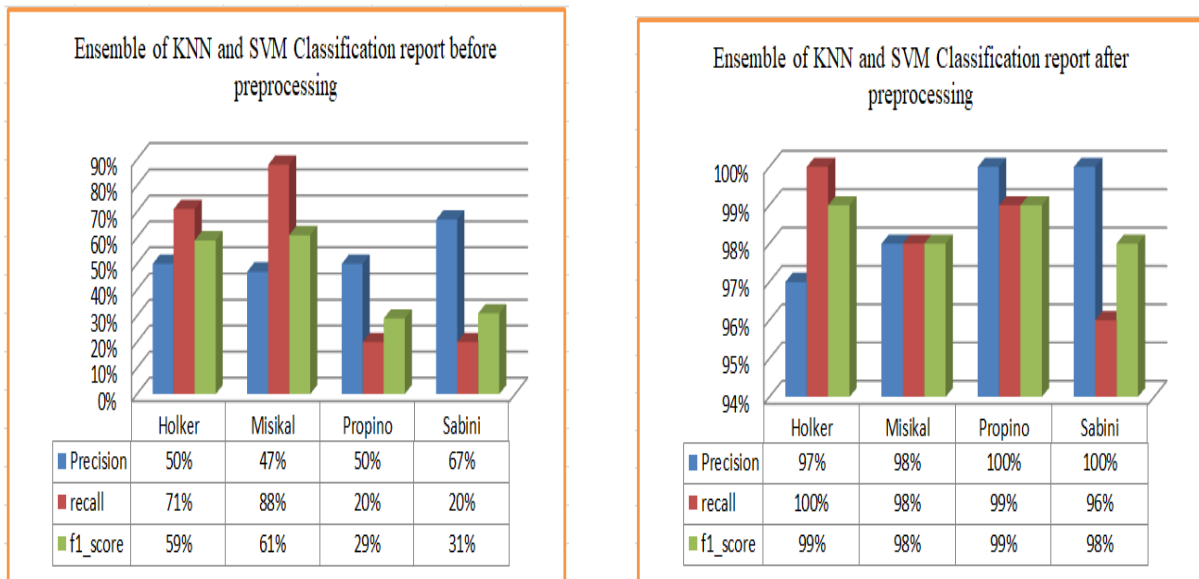
4.5.2.8 Classification Report for Ensemble Classifier after Preprocessing

The below figure 4.13 are a result of Ensemble after preprocessing, segmentation, and augmentation step are involved and represent metrics that are used to evaluate the performance of a model.

	precision	recall	f1-score	support
Holker	0.97	1.00	0.99	787
Misikal	0.98	0.98	0.98	633
Propino	1.00	0.99	0.99	713
Sabini	1.00	0.96	0.98	473
accuracy			0.99	2606
macro avg	0.99	0.98	0.99	2606
weighted avg	0.99	0.99	0.99	2606

Figure 4.13 Precision, recall, and f1-score using ensemble classifier after preprocessing

4.5.2.9 Graph of Classification Report for Ensemble of KNN and SVM Classifier



Graph 4.5 Graph of precision, recall, and f1-score using Ensemble of KNN and SVM classification before preprocessing (left) and after preprocessing (right)

4.6 Discussion of Results

We show that our proposed ensemble models fared better in classifying malt barley seed than the model previously proposed by scholars. In this study, during experimentation, ensemble of KNN and SVM models have been found to work very well in the classification of Ethiopian malt barley seeds. The results of the current classification model and the previous research are depicted in Table 4.5.

Table 4.5 Compare with different previous studies

Name	Classifier	Performance
Barley Varieties Identification	CNN	94%
Application of image processing techniques for malt barley seed identification	Ensemble of ANN and KNN	95.1%
Varietal classification of malt barley	CNN	93%
Our proposed ensemble model	Ensemble of KNN and SVM	98.58%

As can be seen in Table 4.5, the performance of malt barley seed classification model of the current study shows promising result using ensemble of KNN and SVM is 98.58% and the other are related work to malt barley seed classification and identification.

4.7 Research question demonstration

RQ1. Which filtering technique is best for the classification of malt barely seed?

Answer: Median filtering technique is the best filtering technique for our dataset and gives a high classification score. Gaussian filtering is also a good filtering technique.

RQ2. What is the suitable classification algorithm for constructing a model for malt barley seed classification?

Answer: Ensemble of KNN and SVM classifier is the better classifier for our proposed model.

RQ3. What will be done to improve a malt barley seed classification model performance?

Answer: Apply image preprocessing step, Filtering is a very vital step in our model to increase the performance of our model, and data augmentation also contributes to increasing model performance.

4.8 Summary

In this chapter, we have focus on dataset preparation, building a model, experimental setup, experiment result before and after preprocessing, discussion of result and research question and their answer are covered in this chapter.

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATION

5.1. Conclusion

Malt barley is a cash crop that is mostly utilized to make local alcoholic beverages and beer. Malt barleys have different biological and physical characteristics which play a major role in the production of beer. Barleys with those characteristics used in beer production are selected by well-equipped laboratories and highly skilled experts; sometimes the laboratory may not be well-equip and even if the expertise is highly skilled they may face tiredness and be exposed to bias.

The development of a malt barley classification model can aid in the selection and marketing of malt barley, which is a crucial step in the early stages of brewing beer. In this study, an attempt has been made to construct an optimal model for the classification of Ethiopian malt barley seed. In order to achieve this, the study used experimental research, which entails the development of data sets for training and assessing the malt barley seed classification model. Images of Ethiopian malt barley varieties (Holker, Propino, Sabini, and Misikal) are captured from Boortmalt Debre Birhan malt factory.

Classification model is built using KNN, SVM, and an Ensemble of KNN and SVM. The test results show that after the digital image preprocessing step is involved the accuracy of KNN, SVM, and ensemble of KNN and SVM are 98.54%, 98.15%, and 98.58% respectively.

The proposed model achieves 98.58% promising results using an ensemble of KNN and SVM towards the classification of malt barley seed.

5.2. Recommendation

The following recommendations are made for future and further research projects based on the investigation and study results:

1. In removing noises and detecting edges from barley seed image median filtering and Sobel operator were used respectively and it works well to clean noise. However, these noise removal and edge detection techniques cut if seeds are overlapped and have different sizes. Therefore, to improve the performance of the model, future works need to integrate advanced noise removal techniques that separate noises from seeds in noise, overlapped and non-uniform size barleys.
2. Using forward feature selection performs best when the optimal subset has a small number of features and it is unable to remove features that become obsolete after the addition of other features. Hence, future works should adopt feature selection techniques such as information, and principal component analysis to get the best representative features.
3. In this study properly arranged and scanned seed images have been used. The arrangement of seeds affects the process of segmentation. As a result future work should investigate randomly distributed and overlapped barley seeds.
4. In this study, we consider the malt barley seeds for classification; as a result, future work should consider the malt from which barley varieties it is produced.

REFERENCES

- A Beginner's Guide to Image Preprocessing Techniques - Jyotismita Chaki, Nilanjan Dey - Google Books.* (n.d.). Retrieved March 4, 2023, from https://books.google.nl/books?id=Dfp0DwAAQBAJ&printsec=frontcover&dq=what+is+Image+Preprocessing&hl=en&sa=X&redir_esc=y#v=onepage&q=what+is+Image+Preprocessing&f=false
- Araya, A., Habtu, S., Haile, M., Sisay, F., & Dejene, T. (2011). Determination of Local Barley (*Hordeum Vulgare*) Crop Coefficient and Comparative Assessment of Water Productivity for Crops Grown Under the Present Pond Water in Tigray, Northern Ethiopia. *Momona Ethiopian Journal of Science*, 3(1), 65–79. <https://doi.org/10.4314/mejs.v3i1.63686>
- Barley - Wikipedia.* (n.d.). Retrieved December 24, 2022, from <https://en.wikipedia.org/wiki/Barley>
- Barley Production in Ethiopia.* (n.d.). Retrieved January 26, 2023, from <https://www.nationmaster.com/nmx/timeseries/ethiopia-barley-production-fao>
- Convert RGB image to grayscale - MATLAB im2gray.* (n.d.). Retrieved May 26, 2023, from <https://www.mathworks.com/help/matlab/ref/im2gray.html>
- Debayle, J., & Pinoli, J. C. (2011). Theory and applications of general adaptive neighborhood image processing. *Advances in Imaging and Electron Physics*, 167, 121–183. <https://doi.org/10.1016/B978-0-12-385985-3.00002-X>
- Edge detection.* (n.d.).
- Edge enhancement - Wikipedia.* (n.d.). Retrieved March 31, 2023, from https://en.wikipedia.org/wiki/Edge_enhancement
- Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115. <https://doi.org/10.1016/j.engappai.2022.105151>
- Gero, J. S. (n.d.). *Research Methods for Design Science Research: Computational and Cognitive Approaches 1.*

HAILU, B. (2020). *APPLICATION OF IMAGE PROCESSING TECHNIQUES FOR MALT-BARLEY SEED IDENTIFICATION*. <http://ir.bdu.edu.et/handle/123456789/11090>

Image Processing: The Fundamentals - Maria M. P. Petrou, Costas Petrou - Google Books.

(n.d.). Retrieved March 4, 2023, from

https://books.google.nl/books?hl=en&lr=&id=w3BpSIxN9ZYC&oi=fnd&pg=PR23&dq=what+is+image+processing&ots=3MexdB0K3B&sig=YyWdhYe-CwcSOB8tOedyPDwQmoY&redir_esc=y#v=onepage&q=what+is+image+processing&f=false

Image Segmentation: Principles, Techniques, and Applications - Tao Lei, Asoke K. Nandi -

Google Books. (n.d.). Retrieved March 4, 2023, from

https://books.google.nl/books?id=J0aMEAAAQBAJ&printsec=frontcover&dq=what+is+Image+segmentation&hl=en&sa=X&redir_esc=y#v=onepage&q=what+is+Image+segmentation&f=false

Image Segmentation - Search. (n.d.). Retrieved April 27, 2023, from

<https://www.bing.com/search?q=Image+Segmentation+%0D%0A&qs=n&form=QBRE&sp=-1&lq=1&pq=image+segmentation+&sc=1-20&sk=&cvid=CAD30C9AFDE043CB8720AD84E3EF56BF&ghsh=0&ghacc=0&ghpl=>

Khatrı, A., Agrawal, S., & Chatterjee, J. M. (2022a). *Wheat Seed Classification: Utilizing Ensemble Machine Learning Approach*. <https://doi.org/10.1155/2022/2626868>

Khatrı, A., Agrawal, S., & Chatterjee, J. M. (2022b). *Wheat Seed Classification: Utilizing Ensemble Machine Learning Approach*. *Scientific Programming*, 2022. <https://doi.org/10.1155/2022/2626868>

Kozłowski, M., Górecki, P., & Szczypiński, P. M. (2019). Varietal classification of barley by convolutional neural networks. *Biosystems Engineering*, 184, 155–165. <https://doi.org/10.1016/J.BIOSYSTEMSENG.2019.06.012>

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*, 83–106. <https://doi.org/10.1016/B978-0-12->

818366-3.00005-8

- Mulatu, B., Ethiopia, B. L.-B. research and development in, & 2011, undefined. (n.d.). Barley research and development in Ethiopia—an overview. *Researchgate.Net*. Retrieved March 4, 2023, from https://www.researchgate.net/profile/Tesfay-Belay/publication/281897910_Biotypic_and_genetic_variability_in_the_Russian_Wheat_Aphid_Diuraphis_noxiaMordvilko_Homoptera_Aphididae/links/55fcff3708ae07629e1988cf/Biotypic-and-genetic-variability-in-the-Russian-Wheat-Aphid-Diuraphis-noxiaMordvilko-Homoptera-Aphididae.pdf#page=16
- Mulatu, B., & Grando, S. (2011). Barley Research and Development in Ethiopia. *Barley Research and Development in Ethiopia. Proceedings of the 2nd National Barley Research and Development Review Workshop, April*, 1–16.
<http://localhost:8080/xmlui/handle/123456789/198>
- Noise (electronics) - Wikipedia*. (n.d.). Retrieved March 30, 2023, from [https://en.wikipedia.org/wiki/Noise_\(electronics\)](https://en.wikipedia.org/wiki/Noise_(electronics))
- Nowakowski, K., Boniecki, P., Tomczak, R. J., Kujawa, S., & Raba, B. (2012). Identification of malting barley varieties using computer image analysis and artificial neural networks. *Https://Doi.Org/10.1117/12.954155*, 8334, 430–434. <https://doi.org/10.1117/12.954155>
- Nowakowski, K., Boniecki, P., Tomczak, R. J., & Raba, B. (2011). Identification process of corn and barley kernel damages using neural image analysis. *Https://Doi.Org/10.1117/12.896664*, 8009, 75–79. <https://doi.org/10.1117/12.896664>
- Otsu thresholding—image binarization | by HBY coding academic | Medium*. (n.d.). Retrieved June 5, 2023, from <https://hbyacademic.medium.com/otsu-thresholding-4337710dc519>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Sauvola, J., & Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern Recognition*, 33(2), 225–236. [https://doi.org/10.1016/S0031-3203\(99\)00055-2](https://doi.org/10.1016/S0031-3203(99)00055-2)

Sci-Hub | Texture Feature Extraction Techniques for Image Recognition. SpringerBriefs in Applied Sciences and Technology | 10.1007/978-981-15-0853-0. (n.d.). Retrieved May 23, 2023, from <https://sci-hub.st/https://link.springer.com/book/10.1007/978-981-15-0853-0>

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>

Supervised vs. Unsupervised Learning: What's the Difference? | Indeed.com. (n.d.). Retrieved June 9, 2023, from <https://www.indeed.com/career-advice/career-development/supervised-vs-unsupervised-learning>

Techniques & Methods For Image Filtering & Noise reduction | Talentica Blog. (n.d.). Retrieved March 30, 2023, from <https://www.talentica.com/blogs/image-filtering-and-noise-reduction/>

Understanding Morphological Image Processing and Its Operations | by Prateek Chhikara | Towards Data Science. (n.d.). Retrieved May 23, 2023, from <https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756>

Watershed segmentation — skimage 0.21.0 documentation. (n.d.). Retrieved June 5, 2023, from https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_watershed.html

What is a Confusion Matrix in Machine Learning - MachineLearningMastery.com. (n.d.). Retrieved May 26, 2023, from <https://machinelearningmastery.com/confusion-matrix-machine-learning/>

what is feature extraction - Search. (n.d.). Retrieved April 28, 2023, from <https://www.bing.com/search?pglt=43&q=what+is+feature+extraction&aqs=edge..69i57.83499j0j1&FORM=ANCMIS9&PC=U531>

what is image acquisition in image processing - Search. (n.d.). Retrieved April 27, 2023, from <https://www.bing.com/search?q=what+is+image+acquisition+in+image+processing&form=ANNTH1&refig=8ec8396401b44b9e85c755fc66f834a1&sp=2&lq=0&q=SC&pq=what+is+image+acque&sk=SC1&sc=10-19&cvid=8ec8396401b44b9e85c755fc66f834a1>

What is image augmentation - Alumentations Documentation. (n.d.). Retrieved June 5, 2023,

from https://alumentations.ai/docs/introduction/image_augmentation/

Wu, M., & Chen, L. (2016). Image recognition based on deep learning. *Proceedings - 2015 Chinese Automation Congress, CAC 2015*, 542–546.

<https://doi.org/10.1109/CAC.2015.7382560>

Zhang, D. (2019). *Color Feature Extraction*. 49–80. https://doi.org/10.1007/978-3-030-17989-2_4

APPENDIX

Appendix I Sample Python Code

```
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
datagen = ImageDataGenerator(
    rotation_range = 10,
    shear_range = 0.2,
    zoom_range = 0.2,

    #brightness_range = (0.5, 1.5)
)
import numpy as np
import os
from PIL import Image
image_directory = 'C:/Users/desu/Downloads/Image_dataset/Image_dataset/holker_submitted/'
h =64
w=64
dataset = []
my_images = os.listdir(image_directory)
for i, image_name in enumerate(my_images):
    if (image_name.split('.')[1] == 'jpg'):
        image = io.imread(image_directory + image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((w,h))
        dataset.append(np.array(image))
x = np.array(dataset)
i = 0
for batch in datagen.flow(x, batch_size=16,
                           save_to_dir= 'C:/augdataset/holker_submitted',
                           save_prefix='n',
                           save_format='jpg'):
    i += 1
    if i > 280:
        break
```

Table I Sample Python code for dataset augmentation

```

# Ignore the warnings
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
#The directory of the data
Dir = 'C:/data/'
Classes = ['holker_submitted', 'Misikal_submitted', 'propino_submitted', 'sabini_submitted']

#The image size
img_size = 64
for category in Classes:
    path = os.path.join(Dir, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path, img))
        plt.imshow(img_array)
        plt.show()
        break
    break

```

```

#Training data
training_data = []
def creat_traingin_data():
    for category in Classes:
        path = os.path.join(Dir, category)
        class_num = Classes.index(category)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img))
                new_arr = cv2.resize(img_arr, (img_size, img_size))
                training_data.append([new_arr, class_num])
            except Exception as e:
                pass
creat_traingin_data()
#Converting the image to array, because we want the model to train it
the_len = len(training_data)
X = []
y = []
for categories, label in training_data:
    X.append(categories)
    y.append(label)
X = np.array(X).reshape(the_len, -1)
y = np.array(y)
#Flattening the array
out = np.concatenate(X).ravel()
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=3)
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, train_size=0.8, random_state=3)
#The KNN
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
#The grid search

```

```

#The grid search
parameters = {'n_neighbors':[10, 100, 1000]}
grid = GridSearchCV(knn, parameters)
grid.fit(X_train, y_train)
grid_pred = grid.predict(X_test)
#Accuracy of KNN learning training:
print(knn.score(X_train,y_train))
#Validation
print(knn.score(X_validation,y_validation))
#Accuracy of KNN learning:
print(knn.score(X_test, y_test))
#The Confusion_matrix of the KNN model
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test, knn_pred)
print("Confusion_matrix for KNN learning is:")
print(cm)
rep = classification_report(y_test, knn_pred)
print("Classification for KNN learning report is:")
print(rep)

```

Table II Sample Python code for KNN classifier

```

# Ignore the warnings
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
#The directory of the data
Dir = 'C:/data/'
Classes = ['holker_submitted', 'Misikal_submitted', 'propino_submitted', 'sabini_submitted']

#The image size
img_size = 64
for category in Classes:
    path = os.path.join(Dir, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img))
        plt.imshow(img_array)
        plt.show()
        break
    break

```

```

#Training data
training_data = []
def creat_traingin_data():
    for category in Classes:
        path = os.path.join(Dir, category)
        class_num = Classes.index(category)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img))
                new_arr = cv2.resize(img_arr, (img_size, img_size))
                training_data.append([new_arr, class_num])
            except Exception as e:
                pass
creat_traingin_data()
#Converting the image to array, because we want the model to train it
the_len = len(training_data)
X = []
y = []
for categories, label in training_data:
    X.append(categories)
    y.append(label)
X = np.array(X).reshape(the_len, -1)
y = np.array(y)
#Flattening the array
out = np.concatenate(X).ravel()
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y, test_size=0.2,random_state=3)
X_train, X_validation, y_train, y_validation=train_test_split(X_train, y_train, train_size=0.8, random_state=3)
#The SVM
svm = SVC(kernel='poly', C=0.1, gamma=1)
svm.fit(X_train, y_train)
svm_pred = svm.predict(X_test)

```

```

#The grid search
parameters = {'C':[0.1,10], 'gamma':[1, 0.1, 0.01]}
grid = GridSearchCV(svm, parameters)
grid.fit(X_train, y_train)
grid_pred = grid.predict(X_test)
#Accuracy of SVM learning training:
print(svm.score(X_train,y_train))
#Validation
print(svm.score(X_validation,y_validation))
#Accuracy of SVM learning:
print(svm.score(X_test, y_test))
#The Confusion matrix of the SVM model
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test,svm_pred)
print("Confusion matrix for SVM learning is:")
print(cm)
rep = classification_report(y_test, svm_pred)
print("Classification for SVM learning report is:")
print(rep)

```

Table III Sample Python code for SVM classifier

```

# Ignore the warnings
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
#The directory of the data
Dir = 'C:/data/'
Classes = ['holker_submited', 'Misikal_submited', 'propino_submitted', 'sabini_submited']

#The image size
img_size = 64
for category in Classes:
    path = os.path.join(Dir, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path, img))
        plt.imshow(img_array)
        plt.show()
        break
break

```

```

#Training data
training_data = []
def creat_traingin_data():
    for category in Classes:
        path = os.path.join(Dir, category)
        class_num = Classes.index(category)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img))
                new_arr = cv2.resize(img_arr, (img_size, img_size))
                training_data.append([new_arr, class_num])
            except Exception as e:
                pass
creat_traingin_data()
#Converting the image to array, because we want the model to train it
the_len = len(training_data)
X = []
y = []
for categories, label in training_data:
    X.append(categories)
    y.append(label)
X = np.array(X).reshape(the_len, -1)
y = np.array(y)
#Flattting the array
out = np.concatenate(X).ravel()
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y, test_size=0.2,random_state=3)
X_train, X_validation, y_train, y_validation=train_test_split(X_train, y_train, train_size=0.8, random_state=3)
knn = KNeighborsClassifier(n_neighbors=1)
svm = SVC(kernel='poly', C=0.1, gamma=1)
#Then we use Voting Classifier for ensemble Learning:
from sklearn.ensemble import VotingClassifier
Ens = VotingClassifier( estimators= [('KNN',knn),('SVM',svm)], voting = 'hard')

```

```

#Training the Ensemble Learning:
Ens= Ens.fit(X_train , y_train)
ens_pred = Ens.predict(X_test)
#Accuracy of Ensemble Learning training:
print(Ens.score(X_train,y_train))
#Validation
print(Ens.score(X_validation,y_validation))
#Accuracy of Ensemble Learning:
print(Ens.score(X_test, y_test))
#The quality of the KNN model
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test, ens_pred)
print("Confusion_matrix for Ensemble before preprocessing is is:")
print(cm)
rep = classification_report(y_test, ens_pred)
print("Classification for Ensemble before preprocessing report is:")
print(rep)

```

Table IV Sample Python code for Ensemble of KNN and SVM classifier

```

#import libraries
import scipy
import numpy as np
import random
import glob
import time
import cv2
import os
from os import listdir,makedirs
from os.path import isfile,join
import numpy as np
from PIL import Image
from scipy import signal
import matplotlib.pyplot as plt
#median_filter
path='C:/Users/desu/Downloads/Image_dataset/Image_dataset/sabini_submitted/'
midpath='C:/filter/medians/'
IMAGE_NAMES = os.listdir(path)
BALANCE_ALPHA = 0.2
def output_image(display_name, save_name, image):
    cv2.imshow(display_name, image)
    cv2.imwrite(midpath + save_name, image)
    #print("Image %s%s is saved." % (midpath, save_name))
|
def get_kernel():
    return np.ones((3, 3), np.float32) / 9

def get_median(filter_area):
    res = np.median(filter_area)
    return res

```

```

def median_filter(image, height, width):
    for row in range(1, height + 1):
        for column in range(1, width + 1):
            filter_area = image[row - 1:row + 2, column - 1:column + 2]
            image[row][column] = get_median(filter_area)

    return image

def filter_image(image, image_name, filter_name, filtering_function):
    # Get the image size for the kernel looping.
    height, width = image.shape[:2]

    # Add 1px reflected padding to allow kernels to work properly.
    image = cv2.copyMakeBorder(image, 1, 1, 1, 1, cv2.BORDER_REFLECT)

    # print("Calculating %s for %s" % (filter_name, image_name))
    start_time = time.time()
    res = filtering_function(image, height, width)
    # print("Successfully calculated %s for %s in %s seconds." % (filter_name, image_name, str(time.time() - start_time)))

    return res

def main():
    if not os.path.exists(midpath):
        os.makedirs(midpath)

    for image_name in IMAGE_NAMES:
        # Read and print the original image.
        image = cv2.imread(path + image_name, 0)
        cv2.imshow('Original Image: %s' % image_name, image)

        # Calculate the median filtered version and print the resulting image.
        filtered_image = filter_image(image, image_name, 'median filter', median_filter)
        output_image('Median filtered Image: %s' % image_name, '%s_mdf.jpg' % image_name, filtered_image)

    print("Completed all of the images.")

    # Destroy all the images on any key press.
    cv2.waitKey(0)
    cv2.destroyAllWindows()
if __name__ == "__main__":
    main()

```

Table V Sample Python code for Median filter