DEVELOPMENT OF IN-FIELD DATA ACQUISITION SYSTEMS AND MACHINE LEARNING-BASED DATA PROCESSING AND ANALYSIS APPROACHES FOR TURFGRASS QUALITY RATING AND PEANUT FLOWER DETECTION

By

PEYMAN NEMATZADEH

Bachelor Science in Agricultural Engineering
University of Tabriz
Tabriz, Iran
2012

Master of Science in Mechanical Engineering of
Biosystems
University of Tehran
Tehran, Iran
2015

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2022

DEVELOPMENT OF IN-FIELD DATA ACQUISITION SYSTEMS AND MACHINE LEARNING-BASED DATA PROCESSING AND ANALYSIS APPROACHES FOR TURFGRASS QUALITY RATING AND PEANUT FLOWER DETECTION

Dissertation Approved:

Dr. Ning Wang

Dissertation Adviser

Dr. Paul Weckler

Dr. John Long

Dr. Phillip Alderman

Dr. Yanqi Wu

ACKNOWLEDGEMENTS

I would like to express my gratitude to my research advisor, Dr. Ning Wang, for providing me with the opportunity to pursue a doctoral program under her guidance and assistance during my Ph.D. studies. I would like to express my sincere gratitude to my advisory committee members, Dr. Paul Weckler, Dr. John Long, Dr. Phillip Alderman, and Dr. Yanqi Wu, for their precious support and advice on my research work. Their advice has important implications for this dissertation.

I would like to express my profound gratitude to the Associate Vice President of the Division of Agricultural Sciences & Natural Resources, Dr. Scott Senseman, for his support and kind help during the most challenging time of my doctoral studies. I would like to express my sincere gratitude to the Department Head of Biosystems and Agricultural Engineering, Dr. Mari Chinn, for her support and guidance I would like to convey my sincere gratitude and special thanks to Dr. Scott Frazier and Dr. Saleh Taghvaeian for their moral support and advice.

Great appreciation is expressed to the Department of Biosystems and Agricultural Engineering for full financial support. I am grateful to all my friends for their social support, friendship, and for our gatherings. To mention a few, I would like to thank my friends, Pouya Mokhtari Farivar, Hussein Gharakhani, Pouya Ahadi, Saeed Manouchehri, Diako Mahmodi, and Jose Audberto Torres Riasros.

And last but not least, I would like to express my deepest gratitude to my family for their everlasting love and encouragement throughout these three and half years. I miss you so much. My appreciation also goes to my fiancée for her friendship, support, and motivation during the most challenging time. Thank you so much.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: PEYMAN NEMATZADEH

Date of Degree: JULY, 2022

Title of Study: DEVELOPMENT OF IN-FIELD DATA ACQUISITION SYSTEMS AND MACHINE LEARNING-BASED DATA PROCESSING AND ANALYSIS APPROACHES FOR TURFGRASS QUALITY RATING AND PEANUT FLOWER DETECTION

Major Field: BIOSYSTEMS ENGINEERING

Abstract: Digital image processing and machine vision techniques provide scientists with an objective measure of crop quality that adds to the validity of study results without burdening the evaluation process. This dissertation aimed to develop in-field data acquisition systems and supervised machine learning-based data processing and analysis approaches for turfgrass quality classification and peanut flower detection. The new 3D Scanner App for Apple iPhone 12 Pro's camera with a LiDAR sensor provided high resolution of rendered turfgrass images. The battery life lasted for the entire time of data acquisition for an experimental field (49 m × 15 m size) that had 252 warm-season turfgrass plots. The utilized smartphone as an image acquisition tool at the same time achieved a similar outcome to the traditional image acquisition methods described in other studies. Experiments were carried out on turfgrass quality classification grouped into two classes (“Poor”, “Acceptable”) and four classes (“Very poor,” “Poor,” “Acceptable,” “High”) using supervised machine learning techniques. Gray-level Co-occurrence Matrix (GLCM) feature extractor with Random Forest classifier achieved the highest accuracy rate (81%) for the testing dataset for two classes. For four classes, Gabor filter was the best feature extractor and performed the best with Support Vector Machine (SVM) and XGBoost classifiers achieving 82% accuracy rates. The presented method will further assist researchers to develop a smartphone application for turfgrass quality rating. The study also applied deep learning-based features to feed machine learning classifiers. ResNet-101 deep feature extractor with SVM classifier achieved accuracy rate of 91% for two classes. ResNet-152 deep feature extractor with the SVM classifier achieved 86% accuracy rate for four classes. YOLOX-L and YOLOX-X models were compared with different data augmentation configurations to find the best YOLOX object detector for peanut flower detection. Peanut flowers were detected from images collected from a research field. YOLOX-X with weak data augmentation configurations achieved the highest mean average precision result at the Intersection over Union threshold of 50%. The presented method will further assist researchers in developing a counting method on flowers in images. The presented detection technique with required minor modifications can be implemented for other crops or flowers.

Keywords: Turfgrass Quality Classification, Supervised Machine Learning, Deep Feature Extraction, Peanut Flower Detection, Deep Learning

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF EQUATIONS

LIST OF ABBREVIATIONS AND ACRONYMS

AANN	Auto-Associative Neural Network
ANN	Artificial Neural Network
AP	Average Precision
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
DCCN	Deep Convolutional Neural Network
DGCI	Dark Green Color Index
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
GLCM	Gray Level Co-Occurrence Matrix
GPS	Global Positioning System
HSB	Hue, Saturation, Brightness
HSV	Hue, Saturation, Value
IoU	Intersection over Union
mAP	mean Average Precision
NTEP	National Turfgrass Evaluation Program
PCA	Principal Component Analysis
R-CNN	Region Based Convolutional Neural Network
RGB	Red, Green, Blue
ROI	Region Of Interest
RPN	Region Proposal Network
RTK	Real-Time Kinetic
SLAM	Simultaneous Localization And Mapping
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
YOLO	You Only Look Once

CHAPTER I

INTRODUCTION

1.1 Precision Agriculture

Precision agriculture is a management strategy utilized to collect, process, and analyze temporal and spatial data. It combines the data with other information to support management decisions. The decisions depend on the predicted variability to improve resource use and pursue an efficient, productive, profitable, sustainable, and better quality of agriculture production (Cassman, 1999). Precision agriculture applies technologies to meet the before-mentioned objectives. Integration of new and specific technologies can assess and manage variabilities at precise and detailed levels, which cannot be achieved manually (Pierce & Nowak, 1999). Data acquisition can be carried out with a photographic sampling of the crop field. The collected images can be analyzed with an image processing system (Burgos-Artizzu et al., 2010).

1.2 Field Data Acquisition Technologies for Precision Agriculture

There are many different ways to collect visual information in agricultural fields depending on the available technology (Thorp & Tian, 2004). For instance, images can be collected by cameras mounted on aircraft, satellites, ground-based systems, including tractors, agricultural robots, and remote-controlled vehicles, or just by hand (Thorp & Tian, 2004). Image acquisition is a non-destructive method for collecting information about plants and crop quality (Kamilaris & Prenafeta-Boldú, 2018). Machine vision system incorporates sensing units and supporting computer algorithms (Mathanker et al., 2011). Computer vision techniques are started to be utilized in precision agricultural applications in the 1990s (Burgos-Artizzu et al., 2010). Image analysis is a method to quantify and classify plants and crops (Shi et al., 2015).

Digital image processing has been implemented in different agricultural fields to provide information about soil cover, plant health, plant species identification, plant size, and density with readily accessible equipment in the last few decades (Hemming & Rath, 2001; Stier et al., 2013). Features can be extracted from the images and then fed to classifiers to evaluate agricultural product quality (Mathanker et al., 2011). Digital image analysis provides an objective measure of the plant and crop quality that adds to the validity of study results without burdening the evaluation process (Stier et al., 2013). Analyzing images can be done with different techniques, such as machine learning, linear polarizations, wavelet-based filtering, vegetation indices, regression analysis, and deep learning (Kamilaris & Prenafeta-Boldú, 2018).

1.3 Machine Learning-Based Image Processing

In machine learning, a model is fed many training patterns (sets of inputs). Then the model learns and utilizes the training patterns to compute new patterns (O'Mahony et al., 2020). Traditional computer vision techniques are required to manually select object features for the classification algorithms. Classification algorithms are utilized in image classification-related problems (Li et al., 2018). Computer vision algorithms are well-established, transparent, and optimized for their performance and power efficiency (O'Mahony et al., 2020). Machine learning approaches are limited to artificial description abilities, while deep learning techniques are limited to the training data (Li et al., 2018).

Deep learning is a subset of machine learning (O'Mahony et al., 2020). "Deep learning methods use multiple processing layers to discover patterns and structure in an extensive dataset. Each layer learns a concept from the data that subsequent layers build on, the higher the level, the more abstract the learned concepts. Deep learning does not depend on prior data processing and automatically extracts features." [(Rusk, 2016) page 35]. Deep learning can achieve greater accuracy in image classification, semantic segmentation, object detection, and Simultaneous Localization and Mapping (SLAM). It offers excellent accuracy and versatility compared to machine learning techniques when many computing resources are provided (O'Mahony et al., 2020).

The process of visual quality rating and evaluating process in crop production is subjective, time-consuming, and labor-intensive. It is hard to analyze a large number of collected data. Image analysis techniques are utilized for crop identification, classification, or anomaly detection in many agricultural applications. Machine and deep learning models are helping farmers and crop producers to make decisions on plants growth, yield production, and quality in various scenarios (Kamilaris & Prenafeta-Boldú, 2018). Deep convolutional neural networks reliably identify and classify plant species and crops in agricultural research (Lee et al., 2017; Mohanty et al., 2016; Yu et al., 2019). Deep convolutional neural network systems for plant recognition are based on plant leaf features and morphological patterns (Yu et al., 2019).

1.4 Current Challenges in Texture Analysis and Object Detection

There is no formal definition for texture in image processing. This descriptor provides smoothness, coarseness, and regularity (Gonzalez & Woods, 2018). Texture identification and object categorization are some of the most challenging issues in computer vision. State-of-the-art results in texture and object recognition can be taken with local features. The local features are computed at a sparse set of scale- or affine-invariant key point locations found by specialized interest operators (Zhang et al., 2007). Textural images in image processing and machine vision indicate a specific pattern of distribution of the intensity of the pixel illumination replicated sequentially throughout the image. Texture classification, segmentation, synthesis, and shape are among the main issues that texture analysis deals with (Armi & Fekri-Ershad, 2019).

The majority of texture analysis methods in texture classification are categorized under statistical and transform-based methods or a combination of them. Texture image analysis constantly changes due to noise, rotation, scale, illumination, and viewpoint. New methods are rising to overcome some of the before-mentioned challenges (Armi & Fekri-Ershad, 2019). Digital image analysis evaluation for turfgrass color is also affected by ambient light conditions, for instance. The comparison of the digital image analysis data between turfgrass plots derived from various locations and times could be only valid if the images were obtained under similar lighting conditions. Images can be obtained either in an embedded lighting system or at night using standard artificial light sources to overcome the challenges from sunlight. Digital image analysis demonstrated reproducibility over subjective visual ratings of turfgrass coverage despite the challenges (Stier et al., 2013).

Plant species can be identified by characters derived from their leaves, flower shape, or branching structure. Plants like flowers and leaves are flexible objects that cause various deformations. Many flowers and leaves have a three-dimensional nature that enhances the challenge of producing good quality leaf images and loses helpful structure information (Cope et al., 2012). Support Vector Machine (SVM) classifier demonstrated potential for visual classification tasks in agricultural fields (Zhang et al., 2007). Ferreira et al. (2017) achieved a high precision rate in classifying weeds into the grass and broadleaf weeds, respectively, using Support Vector Machines and traditional Neural Networks classifiers with Gabor filter and Scale-Invariant Feature Transform feature extractors (dos Santos Ferreira et al., 2017).

It is possible to identify objects (like flowers or crops) in images with a machine learning system. Machine learning techniques are still limited in processing biological data. Developing a pattern-recognition or machine learning system required rigorous engineering and significant domain expertise to design a feature extractor. The feature extractor transformed the raw data into a suitable internal representation or feature vector. Then the learning subsystem, which is often a classifier, could detect or classify patterns on the input data (LeCun et al., 2015). Objects recognition is a significant challenge in computer vision at considerably different scales (Lin et al., 2017).

Machine vision systems utilizing various sensors and image processing techniques emerged in agricultural fields to achieve a more accurate and less labor-intensive technique for estimating bloom intensity (Dias et al., 2018). An automated, efficient, and precise method to count the number of flowers is greatly needed for early yield prediction (Chen et al., 2019). The problem of classifying objects, which belong to the same category, such as grass, requires an expert or domain-specific knowledge (Angelova & Zhu, 2013). Individual flower detection and accurate location determination by a vision-based system can help farmers obtain yield estimation and mapping. Knowledge of the accurate number of flowers also assists farmers in making better decisions in advance on cultivation practices and the size of the harvest (Lin & Chen, 2018).

Faster R-CNN, a state-of-art deep learning detector, demonstrated the capability to detect strawberry flowers under poor conditions, including illumination, location, and size (Lin & Chen, 2018). Faster R-CNN effectively combined color and morphological information for apple flower detection, leading to significantly better performances than other existing deep learning models (Dias et al., 2018). Faster R-CNN with ResNet-50 network demonstrated improvement for detecting coconuts in their two crucial maturation stages. The model could distinguish tender from mature coconuts and locate them simultaneously (Parvathi & Tamil Selvi, 2021).

Another deep learning network structure is You Only Look Once (YOLO), the most representative work with real-time speed. YOLO divides the image into sparse grids and makes multi-class and multi-scale predictions per grid cell (Ge et al., 2021). The later versions of YOLO, YOLOv2 and YOLOv3 demonstrated a better performance than YOLO (Fu et al., 2021). Fu et al. (2020) tested the YOLOv3-tiny model for a robotic harvesting and fruit picking technology multi-arm operation. They compared it to other deep learning models such as Faster R-CNN and YOLOv2. The average precision of kiwi fruit detection with YOLOv3 was the highest while maintaining a 34 ms detection time per image (Fu et al., 2021). Janowski et al. (2021) detected apples in the field using YOLOv3. YOLO was chosen in this study for its efficiency and possibility of implementing on mobile devices. The accuracy rate was between 80% and 96% (Janowski et al., 2021).

Deep learning is an essential branch of machine learning. Deep learning algorithms have many advantages over traditional machine learning algorithms for image classification and object detection and recognition (Hasan et al., 2021). Deep learning techniques have played a crucial role and potential for excellent image classification, regression, and segmentation in precision agricultural fields (Yang & Xu, 2021).

1.5 Originality and Need of the Research

Variations between classes are essential for deep learning models to differentiate features and characteristics and have an accurate classification performance. Accuracy depends on class variations, which requires rich data variations. Deep learning models outperform other intelligent approaches by accurately predicting crop qualities and classes (Kamilaris & Prenafeta-Boldú, 2018). Digital image processing has been used in different agricultural fields for plant classification (Seeland et al., 2017) and flower detection (Dias et al., 2018). Traditional computer vision techniques can address image classification, segmentation, and detection problems by changing color space by converting RGB (Red, Green, Blue) images to HSV (Hue, Saturation, Value) and morphological operations. However, traditional computer vision techniques on digital images are facing challenges due to the requirements for constant monitoring, manual manipulation, consistent color values and intensity, and the camera angle (viewpoint) for data acquisition. Advantages of machine and deep learning over traditional computer vision techniques are the better performance compared to the traditional methods and processing of complex data such as in-field acquired images, 3D models, video processing, and more (Armi & Fekri-Ershad, 2019; Stier et al., 2013; O'Mahony et al., 2019).

Most of the turfgrass quality studies (Ding et al., 2016; Karcher & Richardson, 2003; Parra et al., 2020; Richardson et al., 2001) using image processing focused on the general percentage of turfgrass cover, color evaluation, and weed detection. To the best of the author's knowledge, no turfgrass image classification study has been done to distinguish turfgrass quality classes such as poor, acceptable, and excellent. Differentiating "acceptable" class against "poor" class quality is challenging for human evaluators but is a crucial step to do. The critical question is to investigate the possibility of distinguishing acceptable class from poor turfgrass quality using digital image processing techniques.

To the best of the author's knowledge, no study has been done on peanut flower detection based on digital images collected either in-field or under lab conditions. Neither the latest version of YOLO developed in 2021 or YOLOX models have been implemented in the agricultural field before. Overall, data acquisition systems for peanut flower detection and turfgrass quality rating based entirely on Python's high-level programming language are unavailable and have not been reported yet to the best of the author's knowledge. This dissertation aimed to develop data acquisition and evaluation systems for turfgrass quality classification and peanut flower detection on in-field images. The study also aimed to generate image datasets for turfgrass quality rating and peanut flower detection, which were not available in any online database. The importance of accessibility of an online dataset is that it can reduce the time spent on in-field data acquisition, which is time-consuming, limited to seasons, costly, and provides uncertain quality of images. Access to an online database can help researchers develop new methods and algorithms for data processing with tested and good-quality data.

1.6 Research Tasks

The objective of this research was to develop in-field data acquisition systems and machine learning-based data processing and analysis approaches for bermudagrass (*Cynodon* spp.) quality classification and peanut flower (*Arachis hypogaea*) detection. The specific tasks included:

Task 1: to select feature extractors to feed machine learning classifiers and eventually find the best classifier for different turfgrass quality ratings.

Task 2: to identify deep learning-based features to feed machine learning classifiers used in Task 1 and compare accuracy rates with those achieved in Task 1 for turfgrass' quality ratings.

Task 3: to utilize deep learning algorithms to identify peanut flowers for Spanish varieties using the new high-performance YOLOX detection models. YOLOX-L and YOLOX-X models were compared with different data augmentation configurations to find the best YOLOX object detector for peanut flower detection.

Task 4: to perform experiments to validate the performance of the developed algorithms for turfgrass quality classification, and peanut flower detection, respectively.

1.7 Research Hypotheses

The first hypothesis is that the traditional machine learning classification algorithms; Support Vector Machine (SVM), and Random forest classifiers can be used effectively for turfgrass (*Cynodon* species) quality rating.

The second hypothesis is that deep learning-based feature extraction will outperform the feature extractors (Gabor filter and GLCM) utilized in machine learning from Task 1.

The third hypothesis is that the latest YOLOX object detector models will demonstrate the feasibility of detecting peanut flowers from in-field acquired RGB (Red, Green, Blue) images. YOLOX-X will achieve higher mean average precision (mAP) than YOLOX-L model.

CHAPTER II

LITERATURE REVIEW

Precision agriculture has been practiced since 1990s. It generally assists in achieving better farm management practices such as fertilizers and herbicides applications. Large farm fields can be apportioned into management zones. Each field receives customized management inputs based on soil types, landscape position, and management history. Managing farmlands requires intensive data acquisition and processing at the right time and locations. Precision agriculture uses the technological advances of computer processing, yield monitoring, remote sensing, and sensor design to process extensive data. It is expected that Agro-businesses in the United States will have a high demand for information management services and technological advances. These technological advances include global positioning system (GPS) location guidance with Real-Time Kinetic (RTK) technology, robotics, and real-time decision-making based on sensor networks (Mulla, 2013).

2.1 Turfgrass Quality Rating

Bermudagrass (*Cynodon* spp.) is a commonly used turfgrass species in an urban environment (Yu et al., 2019). It has been in demand as a high-quality turfgrass over the past 30 years (Stier et al., 2013). The intensity of turfgrass maintenance on golf courses, parks, sports fields, residential and institutional areas increase each year in the United States to meet the demand. Turfgrass has recreational and aesthetic advantages and environmental benefits by covering the bare ground (Balogh & Walker, 1992). Turfgrass quality is defined by several features, including genetic color, leaf texture, density, uniformity, living ground cover, plant health, and drought resistance (Stier et al., 2013; Morris, 2021). The National Turfgrass Evaluation Program (NTEP) published guidelines for turfgrass breeders, researchers, and extension specialists in turfgrass quality evaluation (Morris & Shearmen, 2021).

The data provided by NTEP is the standard for the US turfgrass industry. Unlike other crops, turfgrass quality is a measure of aesthetics and has functional utilization. A well-trained observer or turfgrass evaluator can efficiently distinguish minor differences between turfgrass varieties using a visual rating system. The visual rating requires consistency to guarantee quality. This rating system is based on a 1 to 9 rating scale, where 1 indicates the poorest quality and 9 indicates the highest. A rating of six and above often indicates acceptable quality. Turfgrass quality is based not only on color but also on leaf texture, density, living ground cover, disease or insect damage, uniformity, and environmental stress like drought resistance. Each of these characters has a rating system, between 1 and 9 (Morris, 2021).

Evaluating turfgrass quality has its difficulties due to the complexity of quality ratings methods (Morris & Shearmen, 2021). Plant experts visually carry out traditional quality ratings; however, the rating process is labor-intensive, time-consuming, subjective, and often biased (Ding et al., 2016). Digital image processing and machine vision techniques have been implemented in many agricultural applications in the last few decades to provide information about soil cover, plant health, species identification, size, and density (Hemming & Rath, 2001). Digital image analysis allows scientists to classify several turfgrass quality components using different digital cameras (Stier et al., 2013). The elements of digital cameras include lens system, filters, color filter array, image sensor, and digital image processor (Lanh et al., 2007).

The necessity for observational data acquisition using in-field, aircraft, or satellite remote sensing techniques has increased in farm management. Many crop monitoring applications require image acquisition and processing due to the benefit of temporal frequency and product delivery time (Herwitz et al., 2004). Digital image processing can be used for several tasks, such as identifying plant species (Gebhardt et al., 2006) and estimating canopy cover in grasslands (Bonesmo et al., 2004).

Gebhardt et al. (2006) used digital image processing to identify plant species for weed control in a grassland. Object-oriented image classification was applied. It is an application of pattern recognition in machine vision. The steps required to carry out are image acquisition, pre-processing, image segmentation, feature extraction, and classification. The objects were classified based on their mathematical patterns, including a vector of information containing geometry, color, and texture. A pixel-based image classification utilized spectral information in one or more spectral bands and classifies each pixel based only on this information (Gebhardt et al., 2006).

A vision system was installed on a commercial fruit sorter with four independent inspection lines. The images of fruits were taken with a multispectral camera, which simultaneously captures four bands. These four bands are the three conventional color bands (RGB) and 750 nm (near-infrared). The fruits rotated while moving beneath the camera to capture different angles of the fruits. The designed vision system utilized only near-infrared information to select the fruits and estimate their size and shape. RGB bands were applied for the fruit color estimation and defect detection. The inspection tasks were divided between two digital signal processors (Aleixos et al., 2002).

Two image analysis procedures were performed through the two digital signal processors running in parallel in a primary/secondary architecture. The master processor calculated the geometrical and morphological features of the fruit utilizing only the near-infrared band. The secondary processor estimated the fruit color and detected the skin defects using the four RGBI bands. After the image processing, the primary processor obtained the information from the secondary and sent the result to a control computer (Aleixos et al., 2002). The performance of a developed method always needed to be tested since the performance allows to verify the method viability for automatic tasks in agriculture based on image processing (Guijarro et al., 2011).

2.2 Digital Image Analysis for Turfgrass Quality Rating

Digital image analysis provided an alternative method to measure turfgrass parameters more accurately and efficiently than the traditional visual ratings for estimating turfgrass cover. Digital image analysis was an effective way of determining turfgrass coverage, producing both accurate and reproducible data. The technique effectively removed the internal error and evaluator bias commonly correlated with subjective ratings (Richardson et al., 2001). Karcher and Richardson (2003) accurately determined the HSB (hue, saturation, brightness) levels of Munsell Plant Tissue color chips and quantified the color differences between zoysiagrass (*Zoysia japonica* Steud.) and creeping bentgrass (*Agrostis palustris* Huds.) (Karcher & Richardson, 2003)

Digital image analysis was applied in several turfgrass studies (Aitkenhead et al., 2003; Karcher & Richardson, 2003; Richardson et al., 2001) to evaluate turfgrass research plots. These studies focused on the general percentage of cover and color evaluations. Digital image analysis provided objective data and an evaluation method that included less reliance on highly skilled evaluators and cataloging of research images. Digital image analysis also had some limitations. It was often difficult to differentiate between green weed species and turfgrass with existing image analysis software. It is limited to the evaluation of turfgrass plots with weeds present. It was also difficult to evaluate turfgrass stress response when more than one stressor was present, like drought and disease (Stier et al., 2013).

Digital image analysis also assisted in detecting and enhancing patterns and classifying objects. Crimmins and Crimmins (2008) studied the efficacy of repeat digital photography for monitoring phenologic events in plants. The image processing toolbox (v 5.4) for MatLab version 7.4.0 for Windows (MathWorks, Natick, MA) was implemented to quantitatively capture the number of flowers in bloom for cinch weed (*Pectis papposa* Harvey and Gray) and the trailing windmills (*Allionia incarnata* L.). The automated counting algorithm managed to underpredict trailing windmill bloom counts. The algorithm underpredicted the flower counts by approximately 30% for cinch weed (Crimmins & Crimmins, 2008).

Rath and Kawollek (2009) studied the robotic harvesting of Gerbera ornamentals (*Gerbera jamesonii*) based on the detection and three-dimensional modeling of cut flower pedicels. The detection rates decreased when the number of pedicels per plant increased. The image processing algorithm had a 94% detection rate when one pedicel was present; then, it decreased to 44% when the number of pedicels was three (Rath & Kawollek, 2009). Nisar, Yang, and Ho (2016) predicted the yield of Daisy flowers using digital images analysis. The average predicted yield had an error of 5.52% compared to the reference yield (Nisar et al., 2016). Frey, Robertson, and Bukoski (2007) utilized vector analysis for quantifying rotational symmetry on digital images of *Geranium robertianum* flowers. The utilized method accurately quantified the asymmetry of radial structures and allowed comparisons among individuals and species (Frey et al., 2007).

2.3 Machine Learning Technology

Machine learning and big data technologies, and high-performance computing have created new opportunities to quantify and understand data-intensive processes in the agricultural fields. Machine learning techniques have a learning process to learn from training data to perform a task. Data consists of a set of examples. An example can be a set of features or variables. A feature can be nominal, binary, ordinal, or numeric. The performance of a machine learning model is measured by a performance metric, which can be improved with experience over time. Several statistical and mathematical models are used to calculate the performance of the machine learning models and algorithms. The trained model can be used to classify, predict, or cluster new testing data using the experience obtained during the training process after the end of the learning process. Machine learning tasks are classified depending on the learning type, models, or models applied to execute the selected task. Learning types can be supervised or unsupervised. Learning models can be regression, clustering, and dimensionality reduction (Liakos et al., 2018).

Machine learning is an emerging technology that can assist in finding rules and patterns in large sets of data. It allows better decision-making and informed actions in real-world scenarios without or with minimal human intervention. Machine learning provides a robust and flexible framework for data-driven decision-making and incorporating expert knowledge into the system. These are critical aspects of the machine learning techniques, making them broadly adopted and highly applicable to precision agriculture. Machine learning systems in precision agricultural applications can assist in better decisions on management across space and time (Chlingaryan et al., 2018).

Machine learning techniques provide valuable sources for fruit detection and counting (Bargoti & Underwood, 2017), plant stress identification (Ghosal et al., 2018), early detection of plant diseases (Sinha & Singh Shekhawat, 2020), prediction of irrigation requirements (Goap et al., 2018), classification of plants (Naeem et al., 2021), and so much more. In the following sections, machine and deep learning methods will be introduced through precision agricultural examples for extracting image texture features and flower detection and counting purposes.

2.4 Image Color Indices

A digital image is made up of a finite number of elements. Each element has a specific value and location (Gonzalez & Woods, 2018). Pixels indicate the smallest controllable element of a digital image represented on the screen (Foley & Van Dam, 1983). A digital image consists of rows and columns of pixels; for instance, an image with four columns and three rows has 12 pixels (Eck, 2018). Each pixel contains independent color information (Stier et al., 2013). The colors are made up of a combination of red, green, and blue light. Different colors are built up by varying the intensity of red, green, and blue light. Three numbers giving the intensity of red, green, and blue can specify the color. These three numbers are given between 0 and 1. Zero is the minimum intensity, and one is the maximum. RGB (red, green, blue) color model is the method for specifying the color. Other color models exist besides the RGB color model. One model is called HSV (hue, saturation, value, often called HSB; hue, saturation, brightness), which stands for hue, saturation, and value. HSV color model is considered more natural to describe colors than RGB (Eck, 2018).

Converting RGB levels to HSV parameters is crucial in turfgrass image processing. This step is essential in the data analysis since the HSV color scale represents how the human eye converts color. Conversion from RGB color to HSV can be calculated with Equation 1 (Stier et al., 2013):

$$\text{If } max_{RGB} = R, \text{ then } Hue = 60 \times \left[\frac{G-B}{max_{RGB}-min_{RGB}}\right] \text{ (1)}$$

Dark Green Color Index (DGCI) can correct the errors in images caused by different light conditions and cameras. DGCI combines HSV values into a single measure of dark green color. The Dark Green Color Index provides a more consistent green color measure than the individual HSV values. DGCI can be calculated with Equation 2 (Rorie et al., 2011):

$$DGCI\ value = \frac{\left[\frac{Hue-60}{60}+(1-Saturation)+(1-brightness)\right]}{3} \text{ (2)}$$

Digital image analysis was first implemented as a research tool across agricultural disciplines. It needed acquiring a quality images with a good color representation and minimum glare of shadowing. It is also needed to select appropriate portions of the image to quantify one or more parameters of interest from the selected parts. Digital image analysis requires a high-quality camera with good color rendering properties and specialized image analysis software. An enclosed lightning system was also necessary to eliminate the effects of the ambient light (Stier et al., 2013).

Meyer and Neto (2008) used an improved color vegetation index with an automatic threshold to determine the accuracy of the index on plant-soil-residue images. The collected images were taken of soybean *(Glycine max* (L.) Merrill), sunflower (*Helianthus pumilus*), redroot pigweed (*Amaranthus retroflexus*), and velvetleaf (*Abutilon theophrasti* Medicus) under natural sunlight at noon. The vegetative index images were calculated using MATLAB for Hindman and Shelton's images (The MathWorks, Inc, Natick, MA). Thresholding was performed utilizing the method of Otsu (Otsu, 1979) to obtain excess binary green and normalized difference index images. The Otsu method with excess green index achieved an 88% accuracy rate (Meyer & Neto, 2008).

Aitkenhead et al. (2003) developed and compared two methods to distinguish a specific crop (Autumn King carrot) from weeds (ryegrass (*Lolium perenne*) and Fat Hen (*Chenopodium album*)). The first step was image processing, where the selected plants were highlighted from their background soil with a variety of transformations on RGB colors. The transformation required several steps. The first one was converting the image into a raster pixel array. Then the next step was to determine the proportion of the total RGB of each pixel, which was green, and convert the image into greyscale. The third step was to obtain thresholding, followed by noise reduction from the image. The last step was to split the image into grids; then, the grid was retained for training and testing purposes. The selected grid from the image was then translated onto an input layer of the same size, and training was carried out. This adapted technique is Auto-Associative Neural Network (AANN). The obtained accuracy rate in the study (Aitkenhead et al., 2003) was 65.7% with the AANN technique.

2.5 Image Texture Indices

Feature extraction is an essential step in digital image analysis next to classification. Three features can be extracted from RGB images: color, texture, and shape (Septiarini et al., 2021). Texture analysis has been the interest of many researchers for a long time. Texture analysis algorithms have a comprehensive range of methods, from random field models to multiresolution filtering techniques. Researchers consider texture features for pattern retrieval. As a feature extractor, the Gabor filter is optimal in minimizing the joint two-dimensional uncertainty in space and frequency. It is a linear filter utilized for texture analysis in image processing (Daugman, 1985). The statistics of its microfeatures (such as orientation, scale tunable edge, and bar detectors) in a given region are utilized to characterize the underlying texture information (Manjunath & Ma, 1996). Gabor filter is a linear filter utilized for texture analysis in image processing (Daugman, 1985).

The two-dimensional Gabor functions $g(x,y)$ (Equation 3) and its Fourier transform $G(u,v)$ (Equation 4) is as follows (Manjunath & Ma, 1996):

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] (3)$$

$$G(u,v) = exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} (4)$$

where the 2D Gaussian principal is described by its center locations (x,y), the widths σ_x, $2\pi jWx$ described the wavelength or spatial frequency, phase j, a wave-vector W along the y principle axes, and σ_y $\sigma_u = \frac{1}{2}\pi\sigma_x$ and $\sigma_v = \frac{1}{2}\pi\sigma_y$ (Manjunath & Ma, 1996).

Each image was convolved with a multiple spatial resolution, a multiple orientation set of Gabor filters. The two-dimensional Gabor transform, shown in Equation 3, is a complex value (Lyons et al., 1998).

$$G_{\vec{k},+}(\vec{r}) = \frac{k^2}{\sigma^2} e^{-k^2\|\vec{r}-\vec{r}_0\|^2/2\sigma^2} \cos\left(\vec{k} \times (\vec{r} - \vec{r}_0)\right) - e^{-\sigma^2/2} \text{ (5)}$$

where $G_{\vec{k},+}$ indicates for the filter of even phase and $\vec{k}$ is the filter wave-vector.

$$G_{\vec{k},-}(\vec{r}) = \frac{k^2}{\sigma^2} e^{-k^2\|\vec{r}-\vec{r}_0\|^2/2\sigma^2} \sin\left(\vec{k} \times (\vec{r} - \vec{r}_0)\right) \text{ (6)}$$

where $G_{\vec{k},-}$ indicates for the filter of odd phase. The components of the Gabor filter ($R_{\vec{k}}$) are defined (Equation 7) as the amplitude of the combined even ($R^2_{\vec{k},+}$) and odd filter ($R^2_{\vec{k},-}$) responses (Lyons et al., 1998):

$$R_{\vec{k}} = \sqrt{R^2_{\vec{k},+} + R^2_{\vec{k},-}} \text{ (7)}$$

The response amplitude is less sensitive to position changes than are the linear filter responses (Lyons et al., 1998).

Gray Level Co-Occurrence Matrix (GLCM) is another feature extractor for texture analysis in image processing. GLCM is utilized to retrieve texture properties from the images. GLCM can develop fourteen feature metrics. Specific feature metrics are measured in-depth and repeatedly introduced (Prakash & Saradha, 2021). The commonly selected features for GLCM are energy, inertia, entropy, inverse difference, correlation, and contrast. Energy (Equation 8) can be determined as (Naeem et al., 2021; Prakash & Saradha, 2021):

$$\xi = \sum_u \sum_v (\rho_{uv})^2 \text{ (8)}$$

where u and v are the spatial coordinates and ρ_{uv} is gray level values. Energy is the total of the grey stage's matrix squares of every element.

Inertia (Equation 9) can be obtained:

$$\tau = \sum_u \sum_v (u - v)^2 \rho_{uv} \text{ (9)}$$

Entropy (Equation 10) is the calculation of the volume of knowledge on the pixel intensities that represents the extent of standardized variations in details on the image texture. It can be described as follows:

$$\psi = -\sum_u \sum_v \rho_{uv} \log_2 \rho_{uv} \text{ (10)}$$

Inverse difference (IDE) represents (Equation 11) the identified image texture and can quantify the volume of shift in various part of the frame. It can be determined as:

$$IDE = \sum_u \sum_v \frac{\rho_{uv}}{|u-v|} \text{ (11)}$$

Correlation (Equation 12) is the measurement of column and row similarity values:

$$\varphi = \frac{1}{\sigma_u \sigma_v} \sum_u \sum_v (u - \mu_u)(v - \mu_v) \rho_{uv} \text{ (12)}$$

Contrast (Ge et al., 2021) (Equation 13) represents the difference in the displayed or image signal intensity between two areas of interest:

$$CON = \sum_u \sum_v (u - v)^2 \rho_{uv} \text{ (13)}$$

Feature selection is an essential step in data processing for machine learning-based classification. Feature selection assists in selecting the most beneficial feature. It also removes the features with no importance in the classification process. Feature selection identifies the minimum number of significant features in building the classifier model. It is time-consuming when many features are applied to the classifier to give the result. The accuracy rate can increase, and the time can decrease if a feature selection method is used before running the model (Naeem et al., 2021). One of the feature selection techniques is based on the Principal Component Analysis (PCA) (Jin et al., 2006). PCA is a linear unsupervised feature extraction method for reducing the dimensionality of the data. This reduction aims to obtain lower-dimensional data while preserving as much of the variation of the data as possible (Halko et al., 2009).

Septiarini et al. (2021) proposed a maturity classification method for oil palm (*Elaeis guineensis*) fresh fruit bunch based on a machine vision approach. The maturity level was divided into raw, ripe, and half-ripe classes. The proposed method applied color and texture features for feature selection and classification. Principal component analysis was applied for feature selection to select the most notable features. An artificial neural network (ANN) with a back-propagation algorithm was applied in the classification process to receive the prediction class. The proposed technique distinguished the three classes, achieving a 93% accuracy score (Septiarini et al., 2021).

Sinha and Shekawat (2020) classified two disease types (Northern corn leaf blight and grape leaf spot diseases) that affected four different plant species. Texture features were calculated with the GLCM feature extractor. K-means segmentation, Delta-E color segmentation, and color space histogram thresholding were used and tested to segment the regions of interest (ROI). K-means segmentation achieved the best isolation of the infected area from leaves. Overall, a 74% accuracy rate was achieved in the best-case scenario, using a threshold range of 1200 – 1600 on classifier values. The best precision was obtained when the optimized threshold value was 1250 on the value of the classifier. The best results were also received, where the infection patterns were visible (Sinha & Singh Shekhawat, 2020).

Tavakoli et al. (2021) proposed an algorithm to classify species and cultivars from the same and different species. The analyzed datasets involved different cultivars of white beans, red beans, and pinto beans leaves. The base of the network architecture was VGG16 convolutional neural network architecture and fine-tuned a model trained on the ImageNet dataset. The Stochastic Gradient Descent algorithm optimized the CNN (Convolutional Neural Network) model. The experiments were implemented utilizing Tensorflow 2. The maximum mean classification accuracy rates were 95.86%, 91.37%, and 86.87%, obtained at the species, cultivars from the same species, and cultivars from different species levels, respectively (Tavakoli et al., 2021).

Bohler, Schaepman, and Kneubuhler (2018) used a high-resolution dataset collected with two uncalibrated consumer-grade cameras, carried by a UAV, for crop classification. The data of the two cameras were combined into a dataset. A novel method was proposed to combine spectral and textural information to classify crops in a small structured arable landscape. Random Forest classifier and VHR data from uncalibrated consumer-grade cameras were used. Two textual features (first-order statistics and mathematical morphology) were calculated to incorporate spatial information into the classification chain. Mean, standard deviation, range, and entropy statistical characteristics were employed. Morphological operations comprised dilatation/erosion, opening/closing, opening/closing top hat, opening/closing by reconstruction, and opening/closing by reconstruction top hat. Features were calculated based on the structuring element. The overall accuracy for the complete set of crop classes based on a pixel-based classification was 66.7% (Böhler et al., 2018).

Ding et al. (2016) developed a method for automated evaluation of turfgrass quality from aerial images. Turfgrass quality differed in each plot and was assessed visually by a turfgrass expert. The objective of the study was to utilize aerial images to generate quality evaluations for optimizing irrigation. A neural network was trained to extract appropriate features, such as RGB images' intensity, color, and texture of turfgrass. The Gabor filter and GLCM calculated characteristics such as color percentage, greenness, and texture features. Each vector data had an output label between 0 (worst quality) and 9 (best quality). Both linear and non-linear models were used in the study. The linear models were Lasso, Ridge, Bayesian regression, and SVM as a non-linear model. The received accuracy rates for classification were between 68% (for Lasso regression) and 87% (for SVM) (Ding et al., 2016).

2.6 Image-Based Flower Detection and Counting

Identification of flowers, plants, or crops is critical in precision agriculture since it assists in automatic harvesting, phenotyping, and yield prediction. Image analysis and computer vision provide a fast and non-destructive technique for yield prediction, phenotyping, and pest or disease identification (Afonso et al., 2019). Flower detection is challenging because it involves significant pre-processing steps, such as separating the flowers from the background. Classification is another challenge to represent images of flowers mathematically (Lodh & Parekh, 2017). Another challenging issue with flower detection and counting from images is the heavily overlapped objects by leaves or another flower or fruit (Gao et al., 2020).

Afonso et al. (2019) detected tomato flowers in a greenhouse setting. An Intel RealSense D435 camera (Intel RealSense Technology, California), mounted on a moving platform, was used to capture the yellow flowers. One hundred thirty-two flowers were annotated using the LabelMe tool. The segmentation method was developed using Halcon software (MVTec Software GmbH, Germany) on a Linux Mint 18.2 system. The evaluation was carried out in MATLAB 2017 (The MathWorks, Natick MA, USA) for the annotated ground truth. The precision rate was 0.79 and 0.77 recall (Afonso et al., 2019). Yang et al. (2017) aimed to identify pumpkin flowers and localize them from RGB images taken in a field to predict the pumpkin yield. The results demonstrated that yellow flowers could be detected in HSV color space. The yellow flowers were correctly detected both in daytime and evening; however, the authors also detected flowers on images where the leaf color was yellow. Most errors were caused by withered flowers and occurred when the color was distorted from yellow to white (Yang et al., 2017).

Lin et al. (2017) used a pre-trained CNN model with a VGG-16 feature extractor as an object-level model to improve flower classification. Multiple models were compared. The high-level model classified flowers from an overall point of view and could serve as a mid-level model. The mid-level model could identify local discriminate patterns. Large datasets work better for CNN-based methods (Lin et al., 2017). Seeland et al. (2017) studied classification to comparatively evaluate method combinations towards their classification accuracy in flower image-based plant classification on three different datasets. All classifications were performed using a linear Support Vector Machine (SVM), achieving 94.8% classification accuracies as the best accuracy rate (Seeland et al., 2017).

Lodh and Parekh (2017) proposed a segmentation method that extracts statistical values from an HSV image to determine a threshold. The defined threshold was later used to create a binary mask, separating the foreground from the background. The developed classification model was based on a SVM. During the pre-processing, the background was separated, and only the foreground was segmented out, which demonstrated the flowers. Color and GIST features were then extracted and combined. The combined feature descriptors represented the flowers. The whole feature database was split into two parts: one for training and one for testing. The training features were fed into a classifier. The classifier outputs a predictive model used to test the accuracy of the approach. SVM was chosen since it operates well with high dimensional feature descriptors. The utilized method achieved an 85.93% accuracy rate (Lodh & Parekh, 2017).

Dias et al. (2018) developed a technique for flower identification, which was automated, robust to clutter and changes in illumination, and generalizable to multiple species in orchards. A fully convolutional network was pre-trained on a sizeable multi-class dataset. The model evaluated high-resolution images covering each tree in less than 50 seconds, unlike human workers, who required an average of up to 50 minutes to count the number of flowers per tree. The proposed method for fruit flower segmentation had three principal operations. First, a high-resolution image was divided into smaller patches in a sliding window. Second, each patch was evaluated using a fine-tuned CNN, and lastly, the refinement algorithm on the obtained score maps was applied to compute the final segmentation mask. Four datasets were created and made publicly available for two apples (A and B), a peach, and pear flowers to evaluate the method (Dias et al., 2018).

Region growing refinement algorithm was used, which utilized the score maps available from the CNN to divide the image into regions of high confidence background, high confidence object, an uncertainty region. Using DeepLab, an open source semantic segmentation model, with the RGR algorithm improved prediction and recall rates for validating the apple A set by 15%. Apple A set received 79.4% accuracy, apple B 67.1%, peach 86.8%, and pear 94.1% using DeepLab with RGR. The model trained using only images of apple flowers succeeded in generalizing peach and pear flowers, which are noticeably different in color and morphology, without any supervised fine-tuning or image pre-processing (Dias et al., 2018).

Hiary et al. (2018) used deep learning methods such as CNN besides existing flower datasets to address the flower classification task. The developed automatic method detected the region around the flower in an RGB image. Then the cropped images were used to learn a robust CNN classifier to distinguish different flower classes. Localization was performed by segmenting the flower region using a fully convolutional network (FCN) method. A robust CNN classifier accurately classified different flower classes during the learning process. The segmentation FCN was initialized by the VGG-16 model, while the classification CNN was initialized by segmentation FCN. Detection was performed though finding the minimum bounding box around an automatically segmented flower. Segmentation was achieved as a binary classification task within a fully convolutional network framework. The developed method was assessed on different known flower datasets. The results demonstrated that the proposed technique achieved at least 97% classification accuracy on all datasets (Hiary et al., 2018).

Most of the studies (Afonso et al., 2019; Dias et al., 2018; Lin et al., 2017; Lodh & Parekh, 2017; Seeland et al., 2017; Yang et al., 2017) focused on flower identification, detection, and classification tasks, while another challenging issue counting flowers from images has not been widely studied yet. Farjon et al. (2020) built an automated vision-based system to estimate apple tree blooming intensity and support peak day determination with high accuracy. A field test was conducted to evaluate the developed method. The method for blooming intensity estimation was based on the CNN flower detector, and performance was tested. The performance of the developed automated vision-based system was compared to human expert judgments under field conditions. A Faster R-CNN detector was utilized for flower detection with the VGG-16 network implementation, pre-trained on the ImageNet dataset. The detection task had two stages executed as two different but connected network modules. The first network module was the Region Proposal Network (RPN), which was responsible for finding areas in the image that were likely to contain an object, such as the Regions of Interest (ROI). The selected ROIs were moved into the second network module. The second network module classified the ROIs into M classes + background. A finer box-regressor further refined the suggested bounding boxes and was trained to minimize an object-specific loss function (Farjon et al., 2020).

An on-sight estimator estimated the tree blooming intensity by counting the flowers based on a single image with no past information about the tree. The number of flowers was the output of the detection phase with a detection confidence threshold set to 0.83. The average flower size was selected based on the assumption that larger and more mature flowers indicated an advanced stage of blooming, whereas smaller flowers indicated an earlier stage. The number of flowers for additional detection thresholds selected were 0.7, 0.75, 0.8, 0.92, and 0.99. A sequence-based estimator considered time-series features describing the tree in previous days. The developed CNN-based detector detected flowers reliably despite confounding conditions, including flower viewpoint, illumination variance, flower clustering formation, and flower- occlusion. According to Farjon et al. (2020), blooming intensity estimation was a statistical task since its success does not depend on accurately detecting all flowers but on a rough estimation of their number (Farjon et al., 2020).

Gao et al. (2020) proposed a deep learning-based multi-class fruit detection method to identify apples as multiple classes. VGG16 model of the Faster R-CNN was adapted and implemented for the multi-class fruit detection purpose. Images were collected from the field and taken with front- and backlighting to receive variation. Four classes were defined in this study. The first class referred to fruits that are occluded by leaves. Branches or wires occluded fruits or leaves, or other fruits occluded the fruits included the second-class. The third class was non-occluded fruits, and the last one was those fruits that overlapped. The input images were evaluated VGG16 and ZFNet (Zeiler and Fergus Network). ZFNet is another commonly used network for object detection, which was developed by Zeiler and Fergus (2013) and has a high detection rate and fast detection speed (Zeiler & Fergus, 2013). VGG16 outperformed ZFNet, and the two architectures received minimum average precision rates of 87.9% and 79.3%, respectively. The first class received 89.9% with VGG16 and 81.3% with ZFNet. The second class received 85.8% with VGG16 and 71.3% with ZFNet. The third class received 90.9% with VGG16 and 90.2% with ZFNet, and the fourth class received 84.8% with VGG16 and 74.3% with ZFNet (Gao et al., 2020).

CHAPTER III

SUPERVISED MACHINE LEARNING CLASSIFICATION OF TURFGRASS QUALITIES FROM DATA COLLECTED BY A SMARTPHONE

ABSTRACT: Evaluators rate turfgrass qualities visually, which is challenging because of the complexity of in-field quality evaluation. The evaluation system has a 1 (“poorest”) to 9 (“highest”) scale. Six and above are acceptable turfgrass qualities. Distinguishing poor quality (rate 5) from acceptable (rate 6) is essential for turfgrass breeders. Digital image processing and analysis have been implemented to classify turfgrass quality components (color or density) to assist in the evaluation. No turfgrass image classification study has been done using machine learning techniques to distinguish turfgrass qualities. This study aimed to identify highly significant features to feed machine learning classifiers and eventually find the best classifiers for different turfgrass quality evaluation. 3D scans of turfgrass plots were acquired from a bermudagrass and zoysiagrass research field, using the novel technology of the 3D Scanner App installed on an Apple iPhone 12 Pro with a built-in LiDAR sensor. The 3D scans were rendered to high-quality images for further analysis. Experiments were carried out with two (“Poor”, “Acceptable”) and four (“Very poor,” “Poor,” “Acceptable,” and “High”) classes using machine learning classifiers: Support Vector Machine (SVM), Random forest, XGboost, and Logistic Regression. Gray-Level Co-Occurrence Matrix feature extractor with Random forest classifier achieved the highest classification accuracy (81%) for the testing dataset when two classes were analyzed. Gabor filter worked better for four classes and performed the best with SVM and XGBoost classifiers, achieving an 82% accuracy rate. In the future, deep learning techniques may further increase classification performance metrics.

Keywords*:* *Smartphone, Machine Learning, Dataset Balancing, Image Feature Extraction, Image Classification, Turfgrass Quality Rating*

3. 1. Introduction

Turfgrass breeders have been developing high-quality turfgrass varieties over the past 30 years in the United States (Stier et al., 2013), which are suitable for intense maintenance practices (Gopinath et al., 2021). Unlike other crops, turfgrass quality is a measure of aesthetics and functional utilization (Morris, 2021). The National Turfgrass Evaluation Program (NTEP) standardizes turfgrass qualities (Morris & Shearmen, 2021) to assist traditional visual rating. The rating system is based on a 1 to 9 rating scale, where 1 indicates the poorest quality and 9 indicates the highest. A rating of 6 and above is often used for acceptable turfgrass quality (Morris, 2021). In-field visual turfgrass evaluation has its difficulties due to the complexity of the quality ratings (Morris & Shearmen, 2021). Visual rating is also labor-intensive, time-consuming, subjective, and biased (Ding et al., 2016). Digital image processing and analysis have been implemented to classify turfgrass quality components, such as color (Karcher & Richardson, 2003) or leaf texture (Parra et al., 2020). Various image sensors (Stier et al., 2013) and robotic platforms were used in turfgrass studies for data acquisition (Ding et al., 2016; Gebhardt et al., 2006; Kazemi et al., 2020).

The Apple iPhone Pro 12 smartphone with a novel built-in LiDAR sensor was launched in October 2020. The LiDAR sensor not only can enhance photo and video effects but also creates unique 3D mapping in minutes and optimize the augmented reality (AR) application utilizing state-of-the-art photogrammetry algorithms. The built-in wide-angle iPhone cameras captures additional images to add texture to the 3D scan (Apple, 2022). Tavani et al. (2022) tested and evaluated the video-photographic capabilities of the iPhone 12 Pro for documenting field observations. The 6-core CPU (Central Processing Unit) in the iPhone 12 Pro outperformed all smartphone competitors for its computation capacity, including image processing, at the time of their test. The iPhone Pro 12 is also a powerful device for capturing macro photographs (Tavani et al, 2022). Two studies reported that the Apple iPhone Pro 12 LiDAR was capable of modeling a coastal cliff realistically using Structure from Motion Multi-View Stereo (SfM MVS) photogrammetry (Luetzenburg et al., 2021), and detecting trees and measuring their diameter at breast height (Mokroš et al., 2021). Using the Apple iPhone Pro 12 with built-on LiDAR sensor can provide a cost-effective, user-friendly, convenient, and time-efficient technique for image acquisition.

Most of the turfgrass quality studies (Ding et al., 2016; Karcher & Richardson, 2003; Parra et al., 2020; Richardson et al., 2001) using image processing focused on the general percentage of turfgrass cover, color evaluation, and weed detection. To the best of the author's knowledge, no turfgrass image classification study has been done to distinguish turfgrass quality classes such as poor, acceptable, and excellent. Differentiating acceptable class from poor is challenging for human evaluators but is a crucial step. The critical question is to investigate the possibility of distinguishing acceptable class from poor turfgrass quality using digital image processing techniques. Machine learning provides a robust and flexible framework for data-driven decision-making and incorporating expert knowledge into the system. It allows better decision-making and informed actions in real-world scenarios without or with minimal human intervention (Chlingaryan et al., 2018). Bird et al. (2022) distinguished healthy fruits from undesirable ones in images using machine learning techniques (Bird et al., 2022). The combination of digital image processing and machine learning approaches provided the opportunity to identify objects (like flowers, fruits, or crops) in images (LeCun et al., 2015). Karcher and Richardson (2003) quantified the color differences between zoysiagrass (*Zoysia japonica* Steud.) and creeping bentgrass (*Agrostis palustris* Huds.) using machine learning techniques (Karcher & Richardson, 2003).

This study classified different turfgrass qualities into four classes ("Very poor," "Poor," "Acceptable," "High") from images using supervised machine learning techniques. The study aimed to identify highly significant features to feed classifiers and eventually find the best classifiers. The study also proposed a straightforward in-field data collection method, which was time and cost-effective, and convenient for a research field. The utilized small-field image acquisition tool at the same time achieved a similar outcome to the traditional image acquisition methods described in other studies. This is the first study to use a novel in-field data acquisition tool for turfgrass quality evaluation and the first attempt to fully computerize turfgrass quality evaluation with as little human interaction involved as possible.

3. 2. Material and Methods

The overall developed method consisted of five major steps, including a) a rapid, in-field image acquisition, b) image pre-processing, c) feature extraction, d) feature selection, and e) turfgrass quality classifications.

3.2.1 Data Acquisition

This study used a new image acquisition method for turfgrass classification based on a novel technology provided by the 3D Scanner App on an Apple iPhone 12 Pro with a built-in camera integrated with a LiDAR (Light Detection And Ranging) sensor. There is a LiDAR-based app for iPhone, the 3D Scanner App developed by Laan Labs (New York), which was tested by Tavani et al. (2022) for the accuracy of the LiDAR, and the accuracy of the 3D model measurements taking with the 3D Scanner App. The 3D Scanner App directly can measure distances on the screen with an accuracy of ~ 1 cm. In the test of Tavani et al. (2022), the iPhone 12 Pro LiDAR represented an excellent, highly portable tool for rapid documentation of 3D outcrop pattern, geometry, and texture. The iPhone 12 Pro's LiDAR sensor together with SfM MVS (Structure from Motion Multi-View Stereo) photogrammetric reconstruction techniques generates digital models. SfM MVS models requires post acquisition registrations to properly orient, scale and locate the 3D reconstructed scene with the respect to a local or global coordinate frame (Luetzenburg et al., 2021).

Tavani et al. (2022) also tested in their study the location of selected points in the 3D digital model and compared the locations with their actual location determined by a Stonex S900 GNSS receiver in Real Time Kinematic (RTK) mode. Tavani et al. (2022) found very low angular deviations around the X and Y axes, confirming the capacity of the Apple ARKit framework to recognize the world frame's vertical axis using the iPhone 12 Pro’s built-in gyroscope sensor (Tavani et al, 2022). The Apple iPhone 12 Pro is a convenient device for in-field data collection. In this study, the 3D Scanner App provided a high resolution of rendered turfgrass images, which was essential to extract highly significant features for the supervised machine learning classifications for turfgrass quality, which were developed in this study.

Turfgrass 3D scans were acquired using an Apple iPhone Pro 12 in a replicated bermudagrass and zoysiagrass research nursery field (36°07'03.9"N 97°06'11.4"W) at the Oklahoma State University Turfgrass Research Center in Stillwater, Oklahoma. The experimental field had 252 warm-season turfgrass plots arranged in nine columns and 28 rows. An experienced evaluator visually rated each plot regularly between one and seven (seven was the highest rating in this experimental field during the data collection time). The ratings provided by the evaluator were then used as a ground truth dataset. The turfgrass 3D scans were collected ten times between August 17 and November 2, 2021, using the 3D Scanner App version 1.9.5 (Laan Labs, New York) installed on an Apple iPhone 12 Pro. The turfgrass field was scanned column by column by holding the iPhone in hand, moving slowly, and panning the camera around slowly while taking the 3D scans.

The exact process was repeated until all the nine columns were scanned (Figure 1). The 3D Scanner App then generated three files: texture.jpg, mtl (wavefront material template library), and .obj, for every 3D scan. A complete raw dataset for the experimental field included nine columns × three files (27 data files). The original image resolution that has been used to create texture to align with the 3D scan had a size of 1920 × 1440 pixels. The raw data were exported later to an open-source software, CloudCompare (2021). A summary of the data acquisition can be found on Table 1.

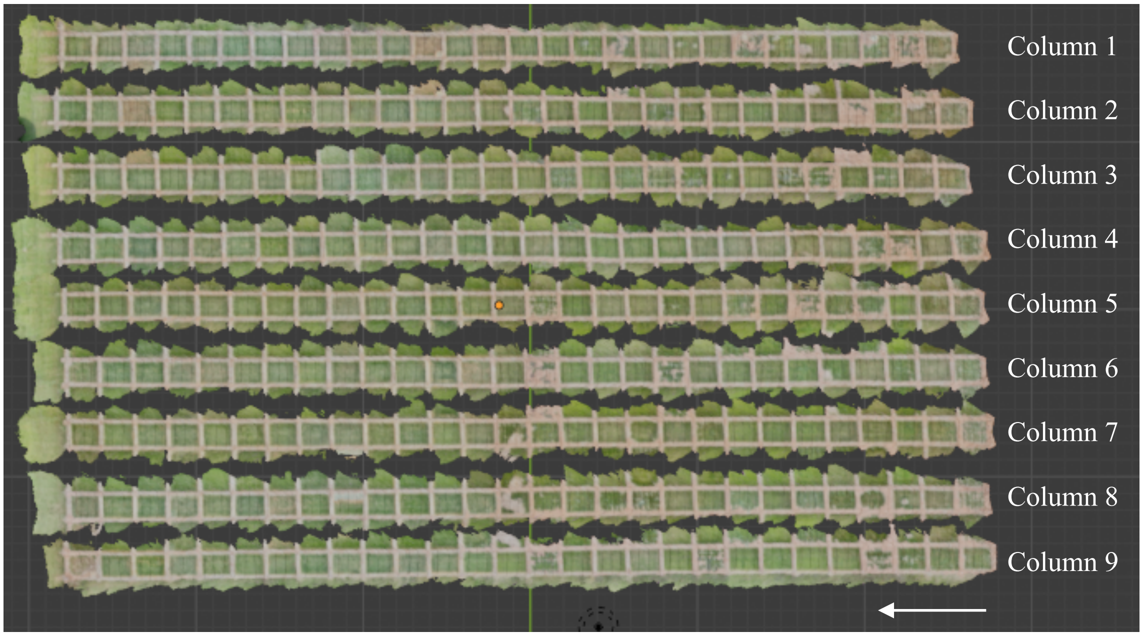


Figure 1 Turfgrass images collected on August 17, 2021, using an Apple iPhone 12 Pro with a built-in camera integrated with LiDAR sensor. The arrow shows the moving direction during data collection.

Table 1 Summary of the data acquisition

Data acquisition	
Smartphone	Apple iPhone Pro 12
Sensor	Built-in camera integrated with LiDAR sensor
Smartphone App	3D Scanner App version 1.9.5 (Laan Labs, NY)
Generated files	texture.jpg, mtl, and .obj
Original image size at the time of the data acquisition	1920 × 1440 pixels
Raw data exported to	Open-source software, Cloud Compare

3.2.2 Image Pre-Processing

Generating Turfgrass Images

The open-source software, CloudCompare (2021), was used to render the collected data files. Each 3D scan data included 28 turfgrass plots. Since every column was scanned individually, the turfgrass plots were separated after rendering. After the 3D scans were rendered, RGB (Red Green Blue) images were generated. An algorithm was developed in Python 3.9.8 to crop each column image automatically into 28 separated plot images. Each separated turfgrass plot image from the entire column included turfgrass and bare soil. The images were saved in .bmp format. Figure 2 demonstrates the image pre-processing steps.

Region of Interest (ROI)

In turfgrass rating, the turfgrass evaluator rates the grass plots based on the quality of the central area of each plot. In this study, each image's region of interest (ROI) was determined by identifying the center of each individual turfgrass plot (separated from the entire column in the before-mentioned paragraph) and cropping the image with a size of 312 × 312 pixels for further analysis (Figure 2c and 2d). The image size of 312 × 312 pixels represented a physical size of 70 cm × 70 cm.

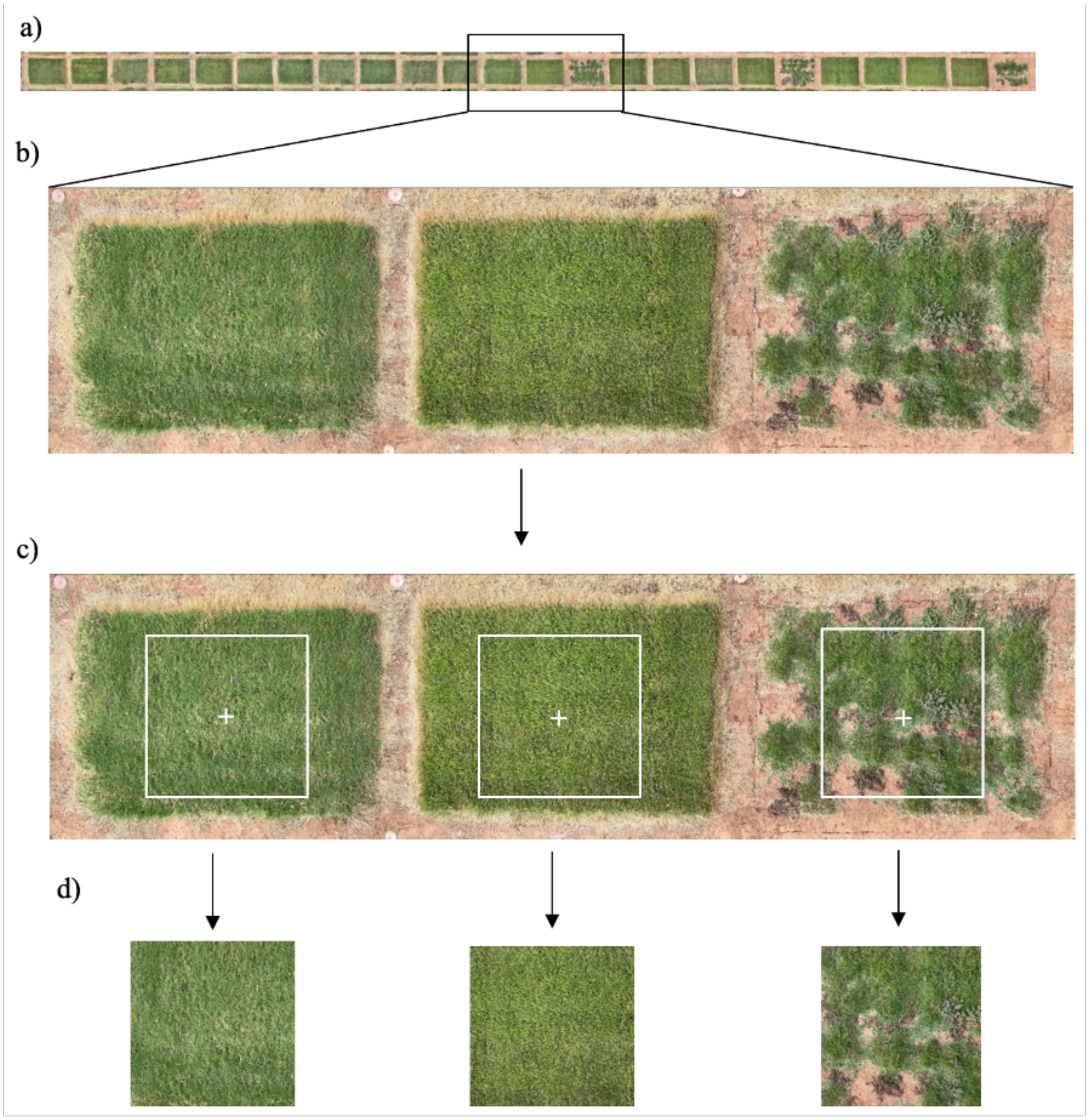


Figure 2 Image pre-processing steps: a) Obtaining a rendered column image from the collected data files; b) Three examples of turfgrass plots; c) Identifying the center of the turfgrass plot (+) and the Region of Interest (ROI) (white square box) d) Cropped ROI images saved in .bmp format.

Image Enhancement and De-Noising

Contrast limited adaptive histogram equalization (CLAHE) is an image enhancement method that was initially developed by Pizer et al. (1987) and extensively used for medical images (Zuiderveld, 1994). CLAHE is based on adaptive histogram equalization (Ding et al.) and designed to broadly apply to low-quality images (Pizer et al., 1987) to avoid noise and reduce the edge-shadowing effect (Reza, 2004). CLAHE algorithm supplies a better contrast, preserves image details (Qiao et al., 2017), and provides maximum entropy compared to other histogram equalization algorithms (Kuran & Kuran, 2021). CLAHE adopted a regional scheme that divides an image into multiple sections and applies individual histograms for each section to prevent over-contrast. CLAHE generates relatively uniform levels of intensities of an image regardless of the illumination of the original image. CLAHE avoids amplifying the noise by limiting sudden increases of slopes in the cumulative probability density function of the histogram (Choi et al., 2016).

CLAHE separates an image into a number of local tiles ($M \times N$). A histogram is then computed individually for each tile. First, the average number of pixels per region is calculated (Pizer et al., 1987) in Equation 14:

$$N_a = \frac{N_X \times N_Y}{N_G} \quad (14)$$

where N_a is the average number of the pixels, N_x is the number of pixels in the X dimension, N_Y is the number of pixels in the Y dimension, and N_G *is* the number of grey levels. Second, the histogram will be clipped using Equation 15 (Pizer et al., 1987):

$$N_{cl} = N_a \times N_{ncl} \qquad (15)$$

where N_{cl} is the clip limit and N_{ncl} is the normalized clip limit between 0 and 1. Third, the clip limit is applied for the height of histogram (Equation 16) (Pizer et al., 1987):

$$H_i = \begin{cases} N_{cl} & if \; N_i \geq N_{cl} \\ N_i & else \end{cases} \; i = 1, 2, \ldots, L - 1 \qquad (16)$$

where *H* is the height of the histogram of the ith tile, N_i is the histogram of the ith tile, and L is the number of grey levels. Fourth, the total number of clipped pixels is computed (Pizer et al., 1987):

$$N_c = (N_X \times N_Y) - \sum_{1=0}^{L-1} H_i \qquad (17)$$

where N_c is the number of the clipped pixels. Fifth, the clipped pixels have to be redistributed either uniformly or non-uniformly (Pizer et al., 1987):

$$N_R = N_c / L \qquad (18)$$

where N_R is the number of the redistributed pixels. Sixth, the clipped histogram is normalized (Pizer et al., 1987):

$$H_i = \begin{cases} N_{cl} & if \; N_i + \; N_R \geq N_{cl} \\ N_i + N_R & \quad else \end{cases} \; i = 1, 2, \ldots, L - 1 \qquad (19)$$

All the pixels need to be redistributed. Finally, a cumulative histogram of the contextual region is be expressed by (Pizer et al., 1987):

$$C_i = \frac{1}{(N_X \times N_Y)} \sum_{j=0}^{i} H_j \qquad (20)$$

Then, the histogram of the contextual region is matched with a uniform and exponential probability distributions that provides a prefixed brightness and visuals quality. The pixel $P(x, y)$ with a value of s and four center points belong to the neighbor tiles $(R_{1,}\ R_2, R_3, and\ R_4)$. A weighted sum is then computed over the contextual regions $(R_{1,}\ R_2, R_3, and\ R_4)$. The tiles are merged for the output image and elimination of the artifacts between independent tiles are done utilizing the bilinear interpolation, the new value is denoted as s'(Pizer et al., 1987):

$$s' = (1 - y) \times (\big((1 - x) \times R_1(s) + x \times R_2(s)\big) + y((1 - x) \times R_3(s) + x \times R_4(s)))$$

(21)

Then, the enhanced image is obtained.

CLAHE was used as an image-enhancement technique for dropped citrus fruit detection on the ground in varying illumination conditions in the study of Choi et al. (2016) and compared with the standard histogram equalization image enhancement technique. CLAHE controlled the over-contrasting, and the images were preserved in detail as a result of the CLAHE algorithm. CLAHE demonstrated to enhance the images that had more significant gaps in brightness levels between dark and bright regions. CLAHE enhanced the brightness of all images to have a constant image brightness level among the in-field collected datasets (Choi et al., 2016). De-noising techniques help recover the original image from a noisy measurement (Buades et al., 2005). Buades, Coll, and Morel (2005) proposed a new image denoising technique called the non-local means. This technique is based on the non-local averaging of all pixels in the image.

In this study, CLAHE image enhancement technique was applied to the turfgrass ROI images. The image enhancement and denoising steps are demonstrated in Figure 3.

Step 1. Each RGB ROI image was converted into an HSV (hue, saturation, value) image.

Step 2. CLAHE was applied on the V channel of the HSV image.

Step 3. The processed V channel of the image was merged back with its HS channels.

Step 4. The new HSV image was converted into an RGB image.

Step 5. The new RGB turfgrass images were converted into gray-scale images.

Step 6. The gray-scale image was de-noised with the non-local means image denoising technique.

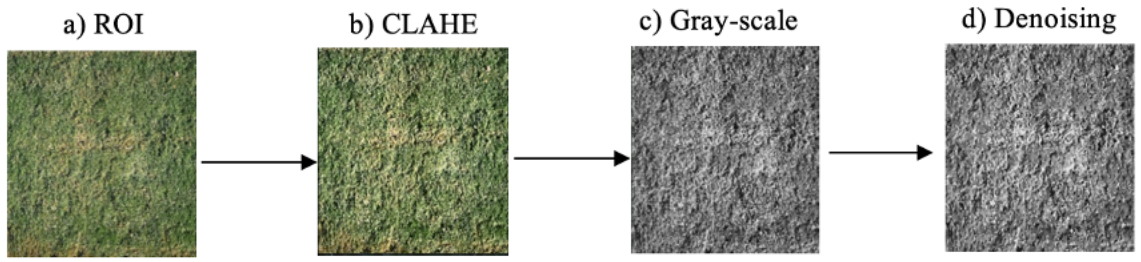


Figure 3 Image enhancement and denoising procedures: a) the ROI image with a size of 312 pixels ×312 pixels; b) the ROI image after CLAHE; c) The Gray-scale image of the enhanced ROI image; and d) the Gray-scale after image enhancement and denoising.

Dataset Balancing

The collected turfgrass images were divided into four classes based on the ground truth and used to develop the image classification algorithms. Class 1 included images rated ≥ 7, labeled as "High" (Figure 4a). Class 2 had images rated as 6 and 6.5, labeled as "Acceptable" (Figure 4b). Class 3 included images rated 5 and 5.5, labeled as "Poor" (Figure 4c). Class 4 included images with ratings between 1 and 4, labeled as "Very poor" (Figure 4d). A borderline between the ratings of 6 and 7, 5 and 6 could be crucial for the interest of turfgrass evaluators. The collected turfgrass image dataset was imbalanced among various rated classes due to dissimilar turfgrass quality expression under field conditions, as demonstrated in (Table 2).

The random under-sampling technique was applied for the majority class ("Poor" class) to eliminate the majority of this class samples. Hansen and Hurwitz (1946) proposed an estimator for random under-sampling for the population mean $\bar{Y}$ (Hansen & Hurwitz 1946):

$$\bar{y}^{*\prime}_{S(HH)} = \sum_{h=1}^{L} P_h \bar{z}^{*}_{h} \quad (22)$$

where $\bar{z}^{*}_{h} = \left(\frac{n_{1h}}{n_h}\right) \bar{z}_{n1h} + \left(\frac{n_{2h}}{n_h}\right) \bar{z}_{kh}\ and\ P_h = \frac{N_h}{N}$. The variance of $\bar{y}^{*\prime}_{S(HH)}$ is given (Hansen & Hurwitz 1946):

$$var\left(\bar{y}^{*\prime}_{S(HH)}\right) = \sum_{h=1}^{L} P_h^2 A_h^{*\prime} \quad (23)$$

where $A_h^{*\prime} = \lambda_{2h}\left(S^2_{hZ} + S^2_{hQ}\right) + \Theta_h\left(S^2_{hZ(2)} + S^2_{hQ(2)}\right), \Theta_h = \frac{P_{2h}(g_h - 1)}{n_h}, P_{2h} = \frac{N_{2h}}{N_h}, and\ \lambda_{2h} = \left(n_h^{-1} - N_h^{-1}\right).$

An oversampling technique, the synthetic minority oversampling technique (SMOTE), was adopted for the minority classes (“High,” “Acceptable,” and “Very poor” classes) to solve the imbalanced data issue (Chawla et al., 2002). SMOTE allows the decision boundaries for the minority class to be spread further into the majority class space (Chawla et al., 2002). SMOTE is a data augmentation technique that generates synthetic data. SMOTE was the first developed oversampling technique that became a favored method for imbalanced datasets. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). Douzas, Bacao, and Last (2018) proved that the overfitting issue was avoided effectively by using SMOTE and also prevented noise generation. Ramenol et al. (2012) addressed the imbalanced datasets with SMOTE and proved it with the rough set theory. Rough set theory in computer science is based on establishing equivalence classes within the given training data (Pawlak 2002).

SMOTE outperforms other data augmentation methods like random oversampling, resulting in overfitting when duplicated data is used. SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data, so the new dataset becomes balanced. SMOTE algorithms linearly interpolate a selected sample and its nearest neighbor. If one image is x, the k nearest neighbors for that x within the class are chosen by matching the selected image x and the remaining images in the same class pixel by pixel and comparing their Euclidean distances. All the images are normalized to the exact resolution and stored according to their image format. The difference between the attribute matrix of each selected nearest neighbor and the sample x is taken. Then the difference is added to the attribute matrix of x after multiplying it with the coefficient of interpolation of the line segment joining two samples, which will be the weighted coefficient w. A weighted coefficient is a random number between 0 and 1. The following equation demonstrates the synthesized sample (Gao et al., 2019):

$$x' = x + w \times (x_{nn} - x) \qquad (24)$$

where x' refers to the new sample synthesized by the original image x, and x_{nn} denotes the nearest neighbors of image x. The exact process is repeated for every image in each class of the original image dataset (Gao et al., 2019).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (25)$$

where $d(x, y)$ is the Euclidean distance for spectral values of any samples, x_i and y_i are the spectral values of two different samples.

SMOTE has several variants (Kovacs, 2019), but in this study, the minority classes were oversampled with the original SMOTE. The majority class ("Poor") was under-sampled from 309 to 200 and 309 to 250 to reduce the computational time.

Table 2 The turfgrass classes in the original and balanced datasets

Class	Name	Total number of images by class	The number of images after random under-sampling	The number of images after SMOTE
Class 1	High	17	17	200 or 250
Class 2	Acceptable	192	192	200 or 250
Class 3	Poor	309	200 or 250	200 or 250
Class 4	Very poor	129	129	200 or 250

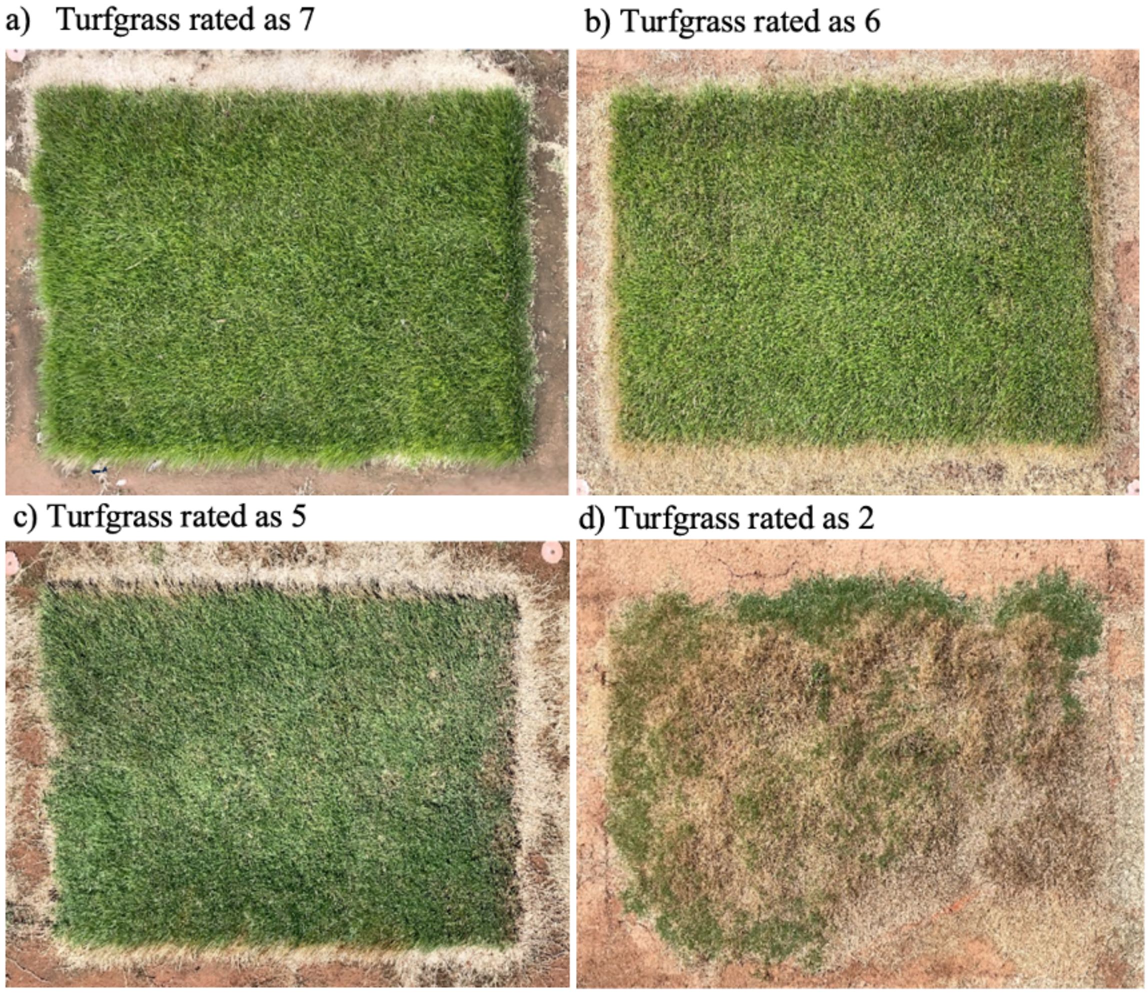


Figure 4 Images of turfgrass plots rated by an experienced evaluator a) turfgrass rated as 7, b) turfgrass rated as 6, c) turfgrass rated as 5, and d) turfgrass rated as 2.

3.2.3 Feature Extraction

Feature extraction is an essential step in digital image analysis next to classification. Three features can be extracted from RGB images: color, texture, and shape (Septiarini et al., 2021). Texture analysis has interested researchers for a long time. Texture analysis algorithms have a comprehensive range of methods, from random field models to multiresolution filtering techniques. Researchers consider texture features for pattern retrieval. Features are critical elements during pre-processing because they facilitate machine learning imitating human visual perception. Many feature extraction methods exist and are performed by different mathematical models. Choosing the best feature extraction algorithm suitable for a study application is necessary (Humeau-Heurtier, 2019).

Gray Level Co-occurrence Matrix (GLCM) is a feature extractor representing a statistical approach for texture analysis in image processing (Haralick et al., 1973). GLCM has been widely used in remote sensing (Rodriguez-Galiano et al., 2012), material science (Prasad et al., 2022), biomedical studies (Ossai & Wickramasinghe, 2022), forensic science (Zhu et al., 2022), and precision agriculture (Wang et al., 2021). GLCM demonstrates promising results for a rapid and straightforward texture feature analysis (Humeau-Heurtier, 2019).

GLCM is utilized to retrieve texture properties from the images. GLCM can develop fourteen feature metrics. Specific feature metrics are measured in-depth and repeatedly introduced. GLCM demonstrated to reduce the occurrence of False Positive (FP) in the evaluation metrics of the classifier performance (Prakash & Saradha, 2021). The commonly selected features for GLCM are energy, inertia, entropy, inverse difference, correlation, and contrast. Energy (Equation 8) can be determined as (Naeem et al., 2021; Prakash & Saradha, 2021):

$$\xi = \sum_u \sum_v (\rho_{uv})^2 \qquad (8)$$

where u and v are the spatial coordinates and ρ_{uv} is gray level values. Energy is the total of the grey stage's matrix squares of every element.

Inertia (Equation 9) can be obtained:

$$\tau = \sum_u \sum_v (u - v)^2 \rho_{uv} \qquad (9)$$

Entropy (ψ), in Equation 10, is the calculation of the volume of knowledge on the pixel intensities that represents the extent of standardized variations in details on the image texture. ρ_{uv} is gray level value, and u and v are the spatial coordinates. It can be described as follows:

$$\psi = -\sum_u \sum_v \rho_{uv} \log_2 \rho_{uv} \qquad (10)$$

Inverse difference (IDE) represents (Equation 11) the identified image texture and can quantify the volume of shift in various part of the frame. It can be determined as:

$$IDE = \sum_u \sum_v \frac{\rho_{uv}}{|u-v|} \qquad (11)$$

Correlation (Equation 12) is the measurement of column and row similarity values:

$$\varphi = \frac{1}{\sigma_u \sigma_v} \sum_u \sum_v (u - \mu_u)(v - \mu_v)\, \rho_{uv} \qquad (12)$$

Contrast (Ge et al., 2021) (Equation 13) represents the difference in the displayed or image signal intensity between two areas of interest:

$$CON = \sum_u \sum_v (u - v)^2\, \rho_{uv} \qquad (13)$$

Gabor filter is a linear filter utilized for texture analysis in image processing (Daugman, 1985; Gabor, 1945). As a feature extractor, Gabor filter is demonstrated to be optimal in minimizing the combined two-dimensional uncertainty in space and frequency.

The statistics of its microfeatures (such as orientation, scale tunable edge, and bar detectors) in a given region are utilized to characterize the underlying texture information (Manjunath & Ma, 1996). Gabor filter can perform a robust multiresolution decomposition due to its localization in the spatial and frequency domain. Gabor filter can simulate specific cells' characteristics in some mammals' visual cortex, facilitating machine learning. Therefore, in image analysis, Gabor filter is considered to be similar in perception to the human visual system (Humeau-Heurtier, 2019).

The two-dimensional Gabor functions $g(x, y)$ (Equation 3) and its Fourier transform $G(u, v)$ (Equation 4) is as follows (Manjunath & Ma, 1996):

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right] \quad (3)$$

$$G(u, v) = exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \quad (4)$$

where $\sigma_u = \frac{1}{2}\pi\sigma_x$ and $\sigma_v = \frac{1}{2}\pi\sigma_y$.

Each image was convolved with a multiple spatial resolution, a multiple orientation set of Gabor filters. The two-dimensional Gabor transform is a complex value (Lyons et al., 1998).

$$G_{\vec{k},+}(\vec{r}) = \frac{k^2}{\sigma^2} e^{-k^2\|\vec{r}-\vec{r}_0\|^2/2\sigma^2} \cos\left(\vec{k} \times (\vec{r} - \vec{r}_0)\right) - e^{-\sigma^2/2} \quad (5)$$

where $G_{\vec{k},+}$ indicates for the filter of even phase and $\vec{k}$ is the filter wave-vector.

$$G_{\vec{k},-}(\vec{r}) = \frac{k^2}{\sigma^2} e^{-k^2\|\vec{r}-\vec{r}_0\|^2/2\sigma^2} \sin\left(\vec{k} \times (\vec{r} - \vec{r}_0)\right) \quad (6)$$

where $G_{\vec{k},-}$ indicates for the filter of odd phase. The components of the Gabor filter ($R_{\vec{k}}$) are defined (Equation 7) as the amplitude of the combined even and odd filter responses (Lyons et al., 1998):

$$R_{\vec{k}} = \sqrt{R^2_{\vec{k},x} + R^2_{\vec{k},-}} \quad (7)$$

The response amplitude is less sensitive to position changes than are the linear filter responses (Lyons et al., 1998).

In this study, Gray Level Co-Occurrence Matrix (GLCM) and Gabor filter feature extractors were applied on the turfgrass images to examine and compare their performance on turfgrass quality classification. The selected features for GLCM were energy, correlation, dissimilarity, homogeneity, and contrast (Haralick et al., 1973). A Python program was developed to compute GLCM with a Scikit-learn (v1.0.2) machine learning library for Python (Pedregosa et al., 2011). The algorithm configuration included the distance of (1, 3, 5) between the pixel pair and their angular regulation of (0°, 45°, 90°, and 135°). The total number of configurations used for the GLCM filter was 60.

Four essential parameters were incremented in the 2-dimensional Gabor construction: the orientation of a Gabor function (θ), standard deviation (σ), the range of wavelengths (λ), and the spatial aspect ratio (γ) (Gabor, 1945). The ratio sigma/lambda (the range of wavelengths (λ) in Table 3) determines the spatial frequency bandwidth of simple cells and thus the number of parallel excitatory and inhibitory stripe zones observed in their receptive fields (Gabor, 1945). The total number of configurations that made up the Gabor filter was 64 using OpenCV 4.5.5 version open-source machine learning library for Python. The utilized Gabor filter parameters are demonstrated in Table 3.

Table 3 The utilized Gabor filter parameters.

Parameters	**Values**
Kernel size	9
Gabor function (θ)	$\frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}, \frac{4\pi}{4}$
Standard deviation (σ)	1,3

Range of wavelengths (λ)	$\frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}, \frac{4\pi}{4}$
Spatial aspect ratio (γ)	0.05, 0.5

3.2.4 Feature Selection

Feature selection is an essential step in data processing for machine learning-based classification. Feature selection assists in selecting the most beneficial feature and removes the features with no importance in the classification process (Lindeberg, 1998). It also removes the features with no importance in the classification process. Feature selection identifies the minimum number of significant features in building the classifier model. It is time-consuming when many features are applied to the classifier to give the result. The accuracy rate can increase, and the time can decrease if a feature selection method is used before running the model (Naeem et al., 2021). Several feature selection techniques exist, which select features that maximize model performance but can be computationally time-consuming (Kursa & Rudnicki, 2010). One of the feature selection techniques is the Boruta algorithm.

The Boruta algorithm creates random features compared to the real features. Boruta is a robust and statistically grounded feature selection method. Boruta algorithm finds all relevant features in the information system. The algorithm utilizes a wrapper approach built around a random forest classifier. Boruta algorithm assists in determining relevance by comparing the relevance of the real features to that of the random probes (Kursa & Rudnicki, 2010). Boruta algorithm, developed by Kursa and Rudnicki (2010), provides criteria for feature selection of important attributes. The idea behind Boruta is that a randomized copy of the system is made. The copy is merged with the original system, and the classifier is built for this extended system. The importance of the variable in the original system is assessed by comparing it with that of the randomized variables. Only the variables are considered important that importance is higher than the importance of the randomized variables. Overall, Boruta algorithm was developed as a random forest-based feature selection method that provides unbiased and stable selection of important attributes from an information system (Kursa, Jankowski, & Rudnicki, 2010). This study used the Boruta algorithm for a feature selection using the BorutaPy library for Python.

3.2.5 Image Classification

Four commonly used machine learning classifiers in texture analysis were chosen in this study: Support Vector Machine (SVM), Random Forest (Sperfeld et al.), XGboost, and Logistic Regression (LR). SVM, Random Forest, and Logistic Regression were selected because they worked effectively for turfgrass quality ratings in the preliminary study. XGBoost classifier was selected based on its computational speed and model performance on large datasets. Figure 5 demonstrates the implementation of the image classifiers.

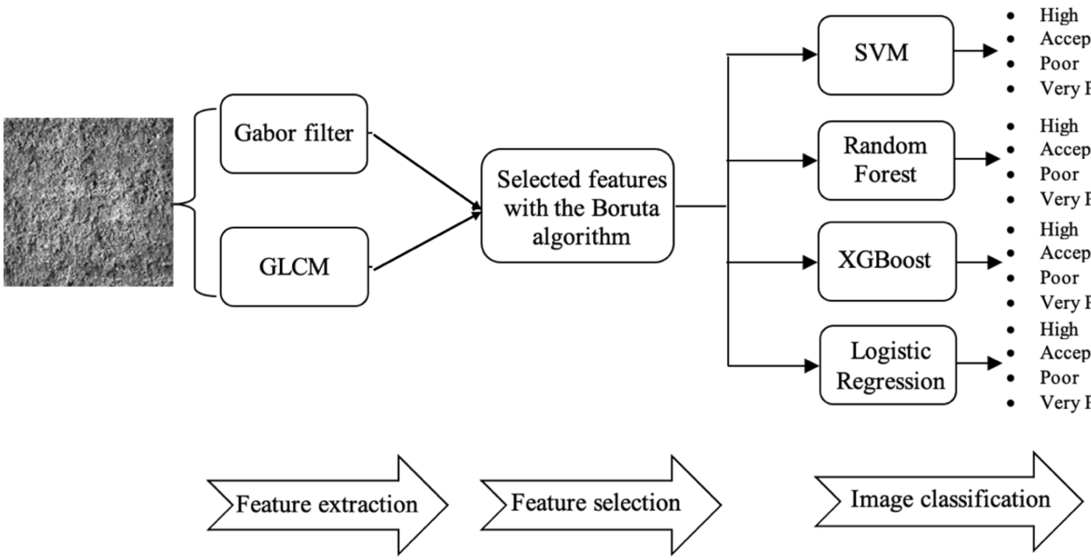


Figure 5 The procedure of the image classification.

3.2.6 Evaluation Metrics For Classifiers Performance

The following metrics were used to assess the performance of the trained models on testing datasets:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{26}$$

$$Precision = \frac{TP}{TP+FP} \tag{27}$$

$$Recall = \frac{TP}{TP+FN} \tag{28}$$

$$F_1\ metric = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \tag{29}$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. The precision rate is equivalent to the proportion of correctly classified instances (Stehman, 1997). Accuracy is the proportion of correct predictions (true positives and true negatives) among the total number of cases analyzed. Precision is the fraction of relevant instances among the retrieved instances, where predictions are relevant or correct. It is the concept of consistency or the ability to group well. Recall, also known as sensitivity, is the fraction of relevant retrieved instances. F_1 metric is a weighted harmonic mean of Recall and Precision (Powers, 2019).

3.2.7 Experimental Design

The experiments (Figure 6) were carried out to test the developed image processing algorithms to find the best classifiers for turfgrass quality rating. The experiments were conducted in the open-source Anaconda Spyder (Anaconda Inc., Austin Texas, USA) environment using Python programming language. The developed algorithms ran on a computer with a processor of an 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80 GHz, and a memory of 32.0 GB RAM at the speed of 3200 MHz.

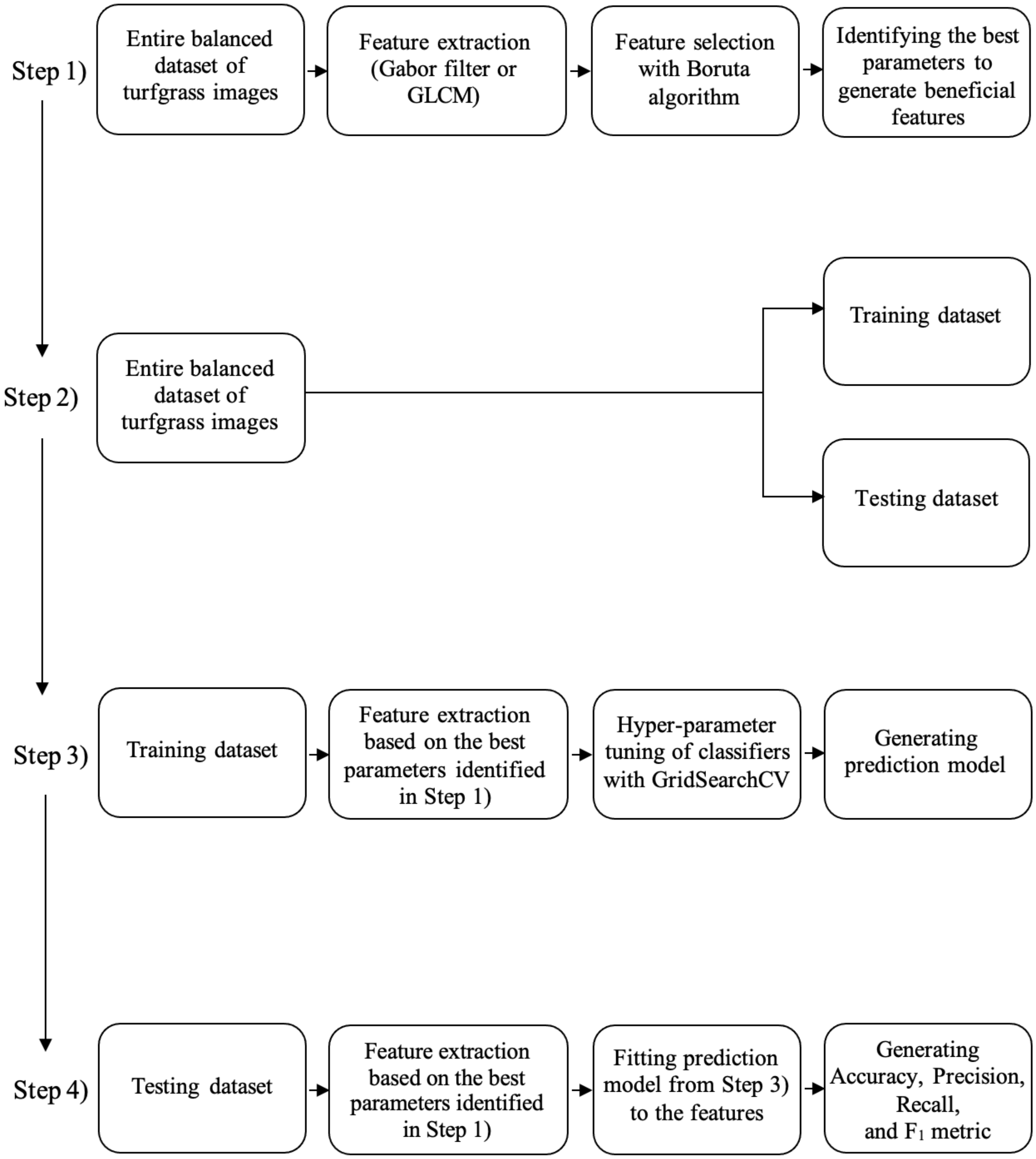

Step 1)
Entire balanced dataset of turfgrass images
Feature extraction (Gabor filter or GLCM)
Feature selection with Boruta algorithm
Identifying the best parameters to generate beneficial features
Step 2)
Entire balanced dataset of turfgrass images
Training dataset
Testing dataset
Step 3)
Training dataset
Feature extraction based on the best parameters identified in Step 1)
Hyper-parameter tuning of classifiers with GridSearchCV
Generating prediction model
Step 4)
Testing dataset
Feature extraction based on the best parameters identified in Step 1)
Fitting prediction model from Step 3) to the features
Generating Accuracy, Precision, Recall, and F1 metric

Figure 6 Experiment procedures for training and testing the proposed methods for turfgrass quality classification.

Fitting the prediction model is essential since it measures how well the machine learning model generalizes to similar data to that on which it was trained. A model is considered well-fitted if it produces more accurate outcomes (Wood, 2011).

The dataset was randomly divided into 80% training and 20% testing datasets. The training dataset was used to train the classifiers (SVM, Random forest, XGBoost, and Logistic Regression), and the testing dataset was utilized to determine the performance metrics of the best machine learning classifiers. Selecting the best feature extractor (Gabor filter or GLCM) with the best classifier was the goal of the experiments. Two groups of tests were conducted, respectively, to classify the samples into two classes: "Acceptable" and "Poor," and four classes: "High," "Acceptable," "Poor," and "Very poor." For each feature extractor method, two sample sizes (number of images), two types of classifications, four image sizes for Gabor filter and five image sizes for GLCM, and four classifiers were used. Total 288 different tests were conducted (Table 4).

Table 4 The experimental setup to find the best classifiers for turfgrass rating.

Feature extractor	Sample size	Number of classes to be classified	Image size (pixels)	Classifiers applied	Number of tests
Gabor filter	200	Four Two	64×64 80×80 100×100 200×200	SVM Random Forest XGBoost Logistic Regression	128
	250	Four Two			
GLCM	200	Four Two	64×64 80×80 100×100 200×200 300×300	SVM Random Forest XGBoost Logistic Regression	160
	250	Four Two			

Resizing the images is an essential step in computer vision. The importance of image resizing is that the machine learning models train faster on small images. Saponara and Elhanashi (2022) tested different input image sizes to assess the effects on the training time and its impact on the model's performance. Saponara and Elhansashi (2022) upscaled and downscaled their original image size to perform their experiment on the image resizing. The resolution also changes with the resizing, but machine learning models tend to learn efficiently from a variety of images with different image resolutions (Saponara & Elhanashi, 2022). Image resizing also improves the model's performance (Talebi & Milanfar, 2021). In this study, OpenCV library on Python was used to resize the images. This image resizing function utilized the nearest-neighbor interpolation method to scale down the images.

Hyper-parameter tuning was applied to all selected classifiers to choose a set of optimal hyperparameters for better prediction results (Bishop, 1995). GridSearchCV library in Python 3.9.8 was used for hyper-parameter optimization and preventing overfitting (Chen et al., 2021). Table 5 demonstrates the hyper-parameter tunning in the chosen classifiers. GridSearchCV library performed *k*-fold cross-validation for the training dataset. The default option of 5-fold cross-validation was used in this study.

Table 5 The hyper-parameters for the selected classifiers.

Classifier	**Parameters**	**Options**
SVM	kernel regularization (C)	linear, poly, rbf, sigmoid 1, 2, 3, …, 10
Random forest	n-estimator random-state	10, 20, 30, …, 100 10, 20, 30, …, 200
XGBoost	learning-rate max-depth subsample n-estimators	0.1, 0.2, 0.3, …, 1 1, 2, 3, …, 10 0.1, 0.2, 0.3, …, 1 1, 10, 20, …, 100
Logistic Regression	penalty solver multi-class	L1, elasticnet, none, L2 newton-cg, sag, saga, liblinear, lbfgs auto, ovr, multinominal

3.3 Results

3.3.1 Experimental Results of the Two-class Tests

For the two classes tests (Table 6), the GLCM feature extractor achieved higher classification accuracy rates for the testing dataset with a sample size of 200. SVM achieved 74%, Random forest, XGBoost, and Logistic Regression achieved 79% classification accuracy rates. GCLM feature extractor proved to be the best feature extractor for the sample size of 250. SVM achieved 76%, Random forest 81%, XGBoost 78% and Logistic Regression achieved 79% classification accuracy rates. Overall, in the two classes ("Acceptable" and "Poor"), the GLCM feature extractor with Random forest classifier and 250 sample size achieved the best accuracy rate with 80 pixels × 80 pixels image size.

Table 6 Results of the two classes' experiments for each classifier in the testing dataset.

Sample size	Feature extractor	Classifier applied	The best accuracy for the testing dataset (%)	Image size (pixels)
200	GLCM	SVM	74	80×80
		Random forest	79	64×64 and 300×300
		XGBoost	79	100×100 and 300×300
		Logistic Regression	79	80×80
200	Gabor filter	SVM	64	200×200
		Random forest	70	100×100
		XGBoost	61	64×64
		Logistic Regression	66	80×80
250	**GLCM**	SVM	76	64×64
		Random forest	**81**	**80×80**
		XGBoost	78	80×80
		Logistic Regression	79	64×64
250	Gabor filter	SVM	74	64×64
		Random forest	75	100×100 and 200×200
		XGBoost	72	64×64
		Logistic Regression	74	64×64 and 100×100

The precision rate was higher for the "Poor" class than the "Acceptable," as shown in Table 7, possibly because the original number of the rendered images before balancing for the "Poor" class was 309, while for "Acceptable" it was 192. The "Poor" class had more varieties of images than "Acceptable", giving more data for training.

Table 7 The best classifier performed on two classes with a sample size of 250.

Classifier applied	Feature extractor	Accuracy (%)	Class	Precision (%)	Recall (%)	F$_1$ metric (%)
Random forest	GLCM	81	Poor	85	72	78
			Acceptable	79	89	83

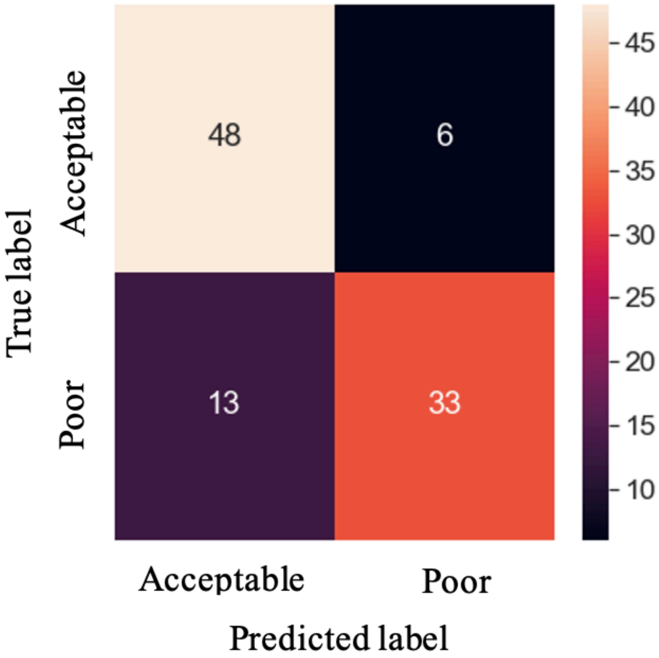


Figure 7 Confusion matrix for Random forest classifier with GLCM features extractor applied to two classes with a sample size of 250.

A confusion matrix (Figures 7) was generated to visualize the performance of the best classifier on turfgrass quality ratings. The confusion matrix helped to visualize each class's prediction errors within the selected classifiers. Random forest classifier with GLCM feature extractor incorrectly predicts 13 images as "Acceptable" and six as "Poor." Distinguishing the "Poor" quality turfgrass from "Acceptable" is challenging, particularly differentiating turfgrass rated as 5 and 6. Creating a two-classes dataset was necessary to test the capability of the proposed method in this study and to analyze the performance of the classifiers on these two critical classes ("Poor" and "Acceptable").

3.3.2 Experimental Results of the Four-class Tests

For the four classes tests (Table 8), Gabor filter feature extractor achieved higher classification accuracy rates for the testing dataset with a sample size of 200. All classifiers with Gabor filter achieved 79% classification accuracy rates but with different image sizes. Gabor filter proved to be the best feature extractor for the sample size of 250. SVM and XGBoost achieved 82%, Random Forest 81%, and Logistic Regression achieved 78% classification accuracy rates. Overall, in the four classes ("High," "Acceptable," "Poor," and "Very poor"), Gabor filter feature extractor performed the best with SVM and XGBoost classifier, and 250 sample sizes achieved the best accuracy rates with 100×100 and 80×80 pixels image sizes, respectively.

Table 8 Results of the four classes' experiments for each classifiers in the testing dataset.

Sample size	Feature extractor	Classifier applied	The best accuracy for the testing dataset (%)	Image size (pixels)
200	GLCM	SVM	69	300×300
		Random forest	79	100×100
		XGBoost	77	300×300
		Logistic Regression	70	200×200
200	Gabor filter	SVM	79	64× 64
		Random forest	79	100×100
		XGBoost	79	80×80
		Logistic Regression	76	200×200
250	GLCM	SVM	65	300×300
		Random forest	73	300×300
		XGBoost	76	200×200
		Logistic Regression	61	64×64
250	**Gabor filter**	**SVM**	**82**	**100×100**
		Random forest	81	200×100
		XGBoost	**82**	**80×80**
		Logistic Regression	78	64×64

Gabor filter as a feature extractor (Table 9) worked better for four classes and performed the best with SVM and XGBoost classifiers. It follows the weakness of the GLCM feature extractor, stating that GLCM lowers the accuracy rates in the region near the class borders. At the same time, Gabor filter is more effective in border regions (Mirzapour & Ghassemian, 2013). GLCM is a suitable feature extractor if only two classes are tested, but Gabor filter outperforms GLCM when more classes are available. The required image sizes were 100 pixels × 100 pixels for SVM and a smaller size for XGBoost (80 pixels × 80 pixels image size) to achieve an 82% accuracy rate. The precision rate was the highest (100%) for the "High" class (rated as 7), probably due to the low number of originally rendered images (17) from the 3D scans that gave less variety among the balanced images (with SMOTE). Since our original dataset is unbalanced, SMOTE generated synthetic data of the original 17 images to increase the number of images to 250. A high precision rate for the "High" class was expected. The second highest precision rate was achieved for the "Very poor" class (rated between 1 and 4) due to the very low quality of turfgrass samples, the texture of the turfgrass was less dense, and the turfgrass images contain bare soil. It was easy to rate them as poor quality.

Table 9 The best classifier performed on four classes with a sample size of 250.

Feature extractor	**Classifier applied**	**Accuracy (%)**	**Class**	**Precision (%)**	**Recall (%)**	**F_1 metric (%)**
Gabor filter	SVM	82	Very poor	90	86	88
			Poor	71	66	69
			Acceptable	71	80	75
			High	100	100	100
Gabor filter	XGBoost	82	Very poor	88	96	92
			Poor	77	59	67
			Acceptable	66	78	71
			High	100	100	100

SVM classifier with Gabor filter feature extractor predicted incorrectly 14 as "Acceptable" and nine images as "Poor" quality. XGBoost classifier with Gabor filter extractor misclassified 18 images as "Acceptable" quality and eight as "Poor." Higher precisions were obtained for "High" and "Very poor" classes. It also might be due to the limited number of images presented for these two classes, and their quality was more pronounced than for "Acceptable" and “Poor” classes. Comparing the precision rates of "Acceptable" and “Poor” classes when two or four classes were present, the "Poor" class always received a higher score than the "Acceptable." Precision rates ranged between 66% and 79% for "Acceptable” and between 71% and 85% for "Poor" class. The result of the confusion matrices (Figures 7, 8 and 9) confirmed that distinguishing the "Acceptable" class from the “Poor” is a challenging task both in real-life and in supervised machine learning.

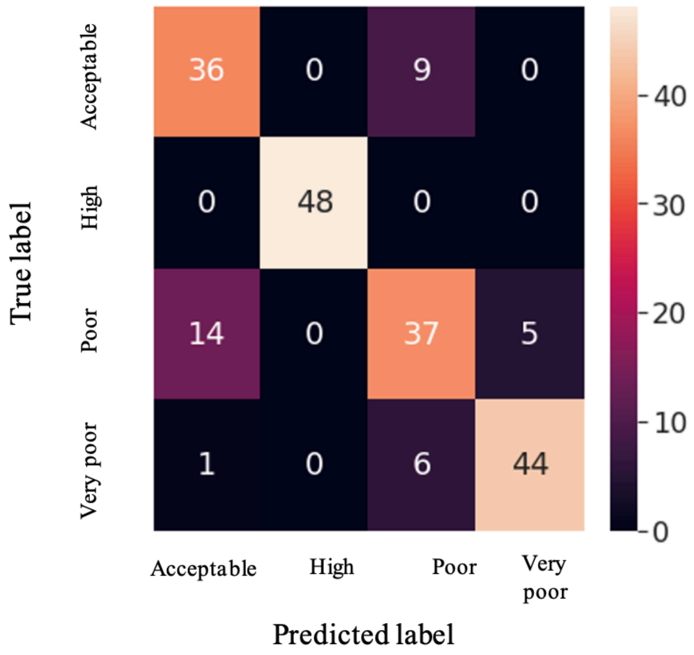


Figure 8 Confusion matrix for SVM classifier with Gabor filter features extractor when applied to four classes with a sample size of 250.

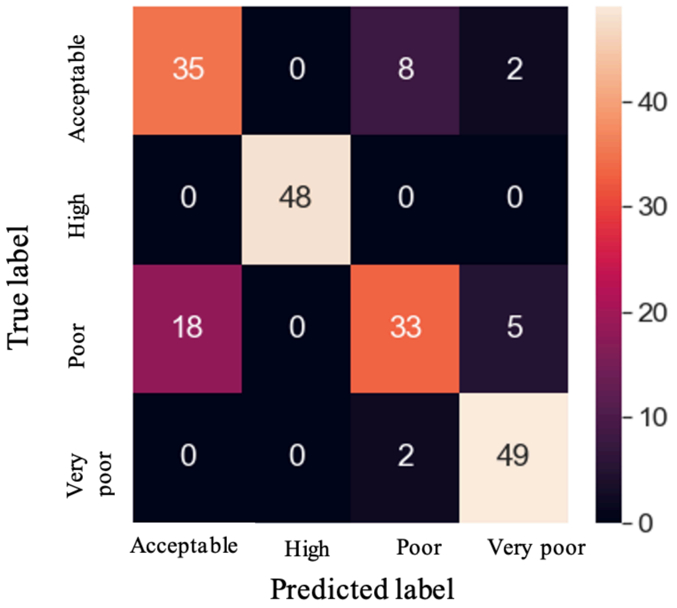


Figure 9 Confusion matrix for XGBoost classifier with Gabor filter features extractor applied to four classes with a sample size of 250.

3.4 Discussion

It is common to use different types of cameras mounted on various platforms in turfgrass image classification and weed detection studies, such as UAV platforms (Ding et al., 2016) and ground-based carts (Parra et al., 2020; Yu et al., 2019). These data acquisition techniques may be costly for small to medium scale turfgrass studies and require attention to additional controlled variables such as the distance between the camera and the object (Parra et al., 2020) light conditions (Gebhardt et al., 2006), local coordinates, camera calibration (Karcher & Richardson, 2003), and heavy equipment (Kazemi et al., 2020).

A smartphone is convenient and easy to carry to the field for data collection while reducing the cost spent on the equipment and time spent on the field. Using a smartphone's camera for image acquisition has several advantages. First, the same camera is used for image acquisition, and then those images are used for processing and training. Second, a smartphone app can be developed later for turfgrass classification, where images will be taken through the smartphone's camera, and the turfgrass plots can be classified on the same smartphone through the app.

Using an Apple iPhone 12 Pro with a built-in camera and LiDAR sensor proved to work well on a small to medium scale field size (252 Bermudagrass plots in nine columns and 28 rows) application capturing morphological features of the turfgrass in this study, directly measuring the distance between the sensor and the target object (Luetzenburg et al., 2021), but also in other studies as well for forestry (Mokroš et al., 2021) and geoscience application (Luetzenburg et al., 2021). In this study, the 3D Scanner App with Apple iPhone 12 Pro's camera with a LiDAR sensor provided high resolution of rendered turfgrass images without considering the additional variables mentioned earlier and proved to be a valuable tool on a small-scale turfgrass field. The battery life lasted for the entire time of data collection for an experimental field with 49 m × 15 m size that had 252 Bermudagrass plots in nine columns and 28 rows. Using such a smartphone with a high-resolution camera and built-in LiDAR sensor can be a valuable and convenient tool for turfgrass breeders, evaluators, and farmers to bring their smartphone and use it in the field. In the future, a smartphone application will be developed to evaluate turfgrass in the field. In that case, anyone can collect and analyze the data on their smartphone without the requirement of using and bringing other tools, sensors, and laptops to the field. For the same reason, using the latest and existing smart technologies (equipped with high-resolution of camera, and different sensors like LiDAR) for data acquisition, such as smartphones, also provides a valuable tool for research studies.

An issue with field data collection for precision agricultural studies is that the collected data generally produce imbalanced datasets, leading to overfitting and biased classifications. Generally, when data collection is required from the field (or from a natural environment) and not from a lab environment, those data will lead to imbalanced data due to the lack of consistency in natural data (like different turfgrass qualities due to the weather conditions, or fertilizer application). Resampling techniques in machine learning can provide a consistent number of samples for each class in multi-classification tasks and are commonly applied to medical data (Liew et al., 2021). Oversampling the minority class and/or under-sampling the majority class overcomes the original imbalanced dataset issue (Chawla et al., 2002).

In a study about estrus onset detection in dairy cows, SMOTE was also proved to overcome an imbalanced dataset issue naturally occurring in environmental data (Wang et al., 2022). In this study, random under-sampling was applied for the majority class ("Poor") to solve the imbalanced data issue. SMOTE was adopted for the minority classes ("High," "Acceptable," and "Very poor"). For four classes when the sample size was resampled to 250 (250 "Very poor," 250 "Poor," 250 "Acceptable," and 250 images of "High"), SVM with Gabor filter feature extractor achieved an 82% accuracy rate in the testing dataset for turfgrass quality classification.

There is a possibility of overfitting using SMOTE before dividing the original dataset into training and testing datasets, which could mean the training and testing datasets are dependent on each other. It is recommended in a future study to assess the likelihood of overfitting; the dataset could be divided into training and testing datasets, and then SMOTE should be applied to the training dataset. Then, these two approaches (applying SMOTE before and after the dataset was divided) could be compared. However, SMOTE is a synthetic minority oversampling technique, a data augmentation technique that generates synthetic data. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data; therefore, the new dataset becomes balanced (Gao et al., 2019), as it was explained on page 43, Dataset balancing; within the Material and Methods section.

Deep learning can also overcome the imbalanced data issue if more image collection is impossible for minority classes. Chen et al. (2022) utilized deep learning for multi-class identification of common weed species in a cotton field. Weighted cross entropy (WCE) loss function was introduced for model training to mitigate the issue of an imbalanced dataset encountered in the cotton weed dataset. The WCE function has been demonstrated to be effective in deep learning models for improving accuracies for the minority weed classes (Chen et al., 2022).

Ding et al. (2016) achieved similar accuracy rates with SVM (87%) but for the training dataset and a total sample size of 60 turfgrass images, without mentioning the total number of images within each studied class (between 4 to 9 quality ratings; (Ding et al., 2016), assuming the analysis of an imbalanced dataset. Imbalanced datasets can give unrealistically optimistic accuracy rates for the performance of machine learning classifiers, such as in the study Flores et al. (2021), where the seedlings of volunteer corn and soybean were successfully classified and distinguished from each other with 95% accuracy.

3.5 Conclusions

This study used a new image acquisition method for turfgrass classification based on the novel technology provided by the 3D Scanner App on an iPhone 12 Pro that has a built-in camera integrated with a LiDAR sensor. This data acquisition method has not been utilized yet for precision agriculture and turfgrass studies. An open-source software, CloudCompare (2021), was used to pre-process and render the collected 3D scans. After rendering the 3D scans, RGB images were generated. The Region of Interest (ROI) was identified on the RGB images. Then an image enhancement method was conducted with CLAHE and denoised with non-local averaging. The collected turfgrass images were divided into four (“High,” “Acceptable,” “Poor,” and “Very poor”) classes for image classification purposes based on the ground truth. An evaluator provided the ground truth for different turfgrass ratings. The collected turfgrass dataset was imbalanced. Random under-sampling was applied for the majority class (“Poor” class), synthetic minority oversampling technique (SMOTE) was adopted for the minority classes (“High,” “Acceptable,” and “Very poor” classes) to solve the imbalanced data issue.

Gabor filter and Gray Level Co-Occurrence Matrix (GLCM) feature extractors were applied to the turfgrass images to examine and compare their performance on turfgrass quality classification. The Boruta algorithm was used for feature selection using the BorutaPy library for Python to identify highly significant features to feed machine learning classifiers. Four commonly used machine learning classifiers were chosen in this study: Support Vector Machine (SVM), Random Forest (Sperfeld et al.), XGboost, and Logistic Regression (LR), to find the best classifiers for different turfgrass quality ratings.

Experiments were carried out with two (“Poor”, “Acceptable”) and four (“Very poor,” “Poor,” “Acceptable,” “High”) determined classes. GLCM feature extractor with Random forest classifier achieved the highest accuracy rate (81%) for the testing dataset for two classes (“Poor” and “Acceptable” turfgrass qualities). For four classes (“Very poor,” “Poor,” “Acceptable,” and “High” turfgrass qualities), Gabor filter proved to be the best feature extractor and performed the best with SVM and XGBoost classifiers achieving 82% accuracy rates.

If possible, it is recommended to collect more images in the future, especially for the minority classes where turfgrasses rated 6 and above. Having more than 17 images in the “High” class would give more realistic precision rates and a more negligible difference between resampling 17 images up to 200 or 250 sample sizes. The proposed classification method in this study, on a research field with 252 Bermudagrass plots, was supervised machine learning, which is still biased and highly dependent on the human evaluators’ judgment (ground truth). An unsupervised machine learning approach is advised for future turfgrass image classification. Another approach can be transfer learning, a machine learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For instance, knowledge gained while learning to recognize plants could be applied to detecting turfgrass (Turkoglu & Hanbay, 2019).

There is a possibility of overfitting using SMOTE before dividing the original dataset into training and testing datasets, which could mean the training and testing datasets are dependent on each other. It is recommended in a future study to assess the likelihood of overfitting; the dataset could be divided into training and testing datasets, and then SMOTE should be applied to the training dataset. Then, these two approaches (applying SMOTE before and after the dataset was divided) could be compared. However, SMOTE is a synthetic minority oversampling technique, a data augmentation technique that generates synthetic data. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data; therefore, the new dataset becomes balanced (Gao et al., 2019), as it was explained on page 43, Dataset balancing; within the Material and Methods section.

Deep learning techniques are also recommended for future turfgrass image classification studies since deep learning can outperform the accuracy rate of machine learning techniques. Deep learning can also overcome the imbalanced data issue if more image collection is not possible for minority classes (such as the “High” class was where turfgrasses rated 6 and above).

CHAPTER IV

SUPERVISED MACHINE LEARNING CLASSIFICATION OF TURFGRASS QUALITIES USING DEEP LEARNING-BASED FEATURES

ABSTRACT: Turfgrass evaluators conduct traditional evaluation of turf quality and related traits visually, which is labor-intensive, time-consuming, and subjective. New imaging technologies and machine learning can improve the efficiency and consistency of the evaluation while reducing labor requirements. Deep learning techniques may further increase classification performance metrics on turfgrass quality classification. This study identified deep learning-based features to feed machine learning classifiers used in Task 1 (Chapter III) and compared accuracy rates with those achieved in Task 1 for turfgrass quality ratings. The overall goal of this research was to utilize the advantages of machine and deep learning methods for turfgrass quality classification on data collected from a research field at the Oklahoma State University Turf Research Center in Oklahoma.

Experiments were carried out with two ("Poor," "Acceptable") and four ("Very poor," "Poor," "Acceptable," and "High") classes using machine learning classifiers: Support Vector Machine (SVM), Random forest, XGboost, and Logistic Regression. The classes were defined visually by a turfgrass expert and used as the “ground truth”. Ten deep feature extractors were tested. ResNet-101 deep feature extractor with SVM achieved the highest classification accuracy (91%) for the testing dataset when two classes were analyzed. ResNet-152 deep feature extractor worked better for four classes and performed the best with SVM classifier, achieving an 86% accuracy rate. Deep feature extractors improved accuracy rates of the tested machine learning classifiers, demonstrating feasibility of deep learning techniques for turfgrass image classification. In the future, deep learning models should perform turfgrass quality classification tasks directly from images since these models can achieve state-of-the-art accuracy rates outperforming machine learning models.

Keywords*:* *Deep Feature Extraction, ResNet-101, ResNet-152, Support Vector Machine, Turfgrass quality classification*

4. 1. Introduction

Turfgrass species including bermudagrass (*Cynodon* species) and zoysiagrass (*Zoysia* species) are commonly used in urban environment. It has been a demand in the United States for high-quality turfgrass. Turfgrass quality represents several features, including genetic color, leaf texture, density, uniformity, living ground cover, plant health, and drought tolerance (Morris, 2021; Morris & Shearmen, 2021; Stier et al., 2013). Turfgrass evaluators assess turfgrass qualities visually based on the guidelines provided by the National Turfgrass Evaluation Program (NTEP). The current evaluation is labor-intensive, time-consuming, subjective, and inaccurate, which data can cause costly errors. The current methods for evaluating plants visual damage and disease quantification may lack accuracy and precision. An alternative method is computer automated digital image analysis, which is consistent, unbiased, and precise (Yu et al., 2019). New imaging technologies and machine learning have substantial potential to improve the efficiency and consistency of the evaluation while reducing labor requirements. Deep learning techniques may further increase classification performance metrics on turfgrass quality classification since deep learning can outperform the accuracy rate of machine learning techniques (LeCun et al., 2015).

Deep learning is a subset of machine learning (O'Mahony et al., 2020). "Deep learning methods use multiple processing layers to discover patterns and structure in a very large dataset. Each layer learns a concept from the data that subsequent layers build on; the higher the level, the more abstract the concepts that are learned. Deep learning does not depend on prior data processing and automatically extracts features."(Rusk, 2016) page 35). Deep learning can achieve greater accuracy in image classification than machine learning (LeCun et al., 2015). It offers excellent accuracy and versatility compared to machine learning techniques when many computing resources are provided (O'Mahony et al., 2020).

Deep learning-based features can be extracted from images using pre-trained models. Extracting deep features is beneficial since they have an outstanding performance in different domains compared to machine learning-based features. Also, deep learning-based feature extraction requires little pre-processing steps compared to traditional machine learning methods (Almeida et al., 2022). Deep convolutional neural network

(DCNN) can transfer the weights learned from large generic datasets to assign them to another more specific but related data. Dourado-Filho and Calumby (2021) demonstrated that exploiting deep learning-based feature extraction with traditional machine learning classifiers, such as SVM, can outperform traditional machine learning-based feature extraction for plant species classification (Dourado-Filho & Calumby, 2021). Deep convolutional neural network systems for plant recognition are based on plant leaf features and morphological patterns (Yu et al., 2019).

This study aimed to utilize deep learning-based features to feed machine learning classifiers used in Task 1 (Chapter III) and compare the accuracy rates with those achieved in Task 1 for turfgrass quality ratings. The overall goal of this research was to utilize the advantages of machine and deep learning methods for turfgrass quality classification on data collected from a research field in Oklahoma. The hypothesis was that deep feature extraction would outperform the feature extractors (Gabor filter and GLCM) utilized in machine learning (from Task 1).

4.2 Material and Methods

The overall developed method consisted of four major steps, including a) a rapid, in-field image acquisition, b) image pre-processing, c) deep-learning based feature extraction, and d) turfgrass quality classifications with supervised machine learning. The methodology for Task 2 was the same or similar as in Task 1 (Chapter III) until the data processing step.

4.2.1 Data Acquisition

Data were acquired at the same time and using the same method as it was described in Chapter III, Sections 3.2.1 and 3.2.2. Turfgrass 3D scans were acquired using an Apple iPhone Pro 12 in a replicated bermudagrass and zoysiagrass research nursery field (36°07'03.9"N 97°06'11.4"W) at the Oklahoma State University Turfgrass Research Center in Stillwater, Oklahoma. The experimental field had 252 warm-season turfgrass plots arranged in nine columns and 28 rows. An experienced evaluator visually rated each plot regularly between one and seven (seven was the highest rating in this experimental field during the data collection time). The ratings provided by the evaluator were then used as a ground truth dataset. The turfgrass 3D scans were collected ten times between August 17 and November 2, 2021, using the 3D Scanner App version 1.9.5 (Laan Labs, New York) installed on an Apple iPhone 12 Pro. The turfgrass field was scanned column by column by holding the iPhone in hand, moving slowly, and panning the camera around slowly while taking the 3D scans. The exact process was repeated until all the nine columns were scanned (Figure 1 on page 37). The 3D Scanner App then generated three files: texture.jpg, mtl (wavefront material template library), and .obj, for every 3D scan. A complete raw dataset for the experimental field included nine columns × three files (27 data files). The original image resolution that has been used to create texture to align with the 3D scan had a size of 1920 × 1440 pixels. The raw data were exported later to an open-source software, CloudCompare (2021). A summary of the data acquisition can be found in Table 1.

4.2.2 Image Pre-Processing

Generating Turfgrass Images

The open-source software, CloudCompare (2021), was used to render the collected data files. Each 3D scan data included 28 turfgrass plots. Since every column was scanned individually, the turfgrass plots were separated after rendering. After the 3D scans were rendered, RGB (Red Green Blue) images were generated. An algorithm was developed in

Python 3.9.8 to crop each column image automatically into 28 separated plot images. Each separated turfgrass plot image from the entire column included turfgrass and bare soil. The images were saved in .bmp format. Figure 2 demonstrates the image pre-processing steps.

Region of Interest (ROI)

In turfgrass rating, the turfgrass evaluator rates the grass plots based on the quality of the central area of each plot. In this study, each image's region of interest (ROI) was determined by identifying the center of each individual turfgrass plot (separated from the entire column in the before-mentioned paragraph) and cropping the image with a size of 312 × 312 pixels for further analysis (Figure 2c and 2d). The image size of 312 × 312 pixels represented a physical size of 70 cm × 70 cm.

Image Enhancement and De-Noising

Image enhancement and de-noising techniques were applied in a same way as they were written in Chapter III, Section 3.2.2. Contrast limited adaptive histogram equalization (CLAHE) is an image enhancement method that was initially developed by (Pizer et al., 1987) and extensively used for medical images (Zuiderveld, 1994). CLAHE is based on adaptive histogram equalization (Ding et al.) and designed to broadly apply to low-quality images (Pizer et al., 1987) to avoid noise and reduce the edge-shadowing effect (Reza, 2004). CLAHE algorithm supplies a better contrast, preserves image details (Qiao et al., 2017), and provides maximum entropy compared to other histogram equalization algorithms (Kuran & Kuran, 2021). CLAHE adopted a regional scheme that divides an image into multiple sections and applies individual histograms for each section to prevent over-contrast. CLAHE generates relatively uniform levels of intensities of an image regardless of the illumination of the original image. CLAHE avoids amplifying the noise by limiting sudden increases of slopes in the cumulative probability density function of the histogram (Choi et al., 2016).

CLAHE was used as an image-enhancement technique for dropped citrus fruit detection on the ground in varying illumination conditions in the study of Choi et al. (2016) and compared with the standard histogram equalization image enhancement technique. CLAHE controlled the over-contrasting, and the images were preserved in detail as a result of the CLAHE algorithm. CLAHE demonstrated to enhance the images that had more significant gaps in brightness levels between dark and bright regions. CLAHE enhanced the brightness of all images to have a constant image brightness level among the in-field collected datasets (Choi et al., 2016). De-noising techniques help recover the original image from a noisy measurement (Buades et al., 2005). Buades, Coll, and Morel (2005) proposed a new image denoising technique called the non-local means. This technique is based on the non-local averaging of all pixels in the image.

In this study, CLAHE image enhancement technique was applied to the turfgrass ROI images. The image enhancement and denoising steps are demonstrated in Figure 3 on page 41.

Step 1. Each RGB ROI image was converted into an HSV (hue, saturation, value) image.

Step 2. CLAHE was applied on the V channel of the HSV image.

Step 3. The processed V channel of the image was merged back with its HS channels.

Step 4. The new HSV image was converted into an RGB image.

Step 5. The new RGB turfgrass images were converted into gray-scale images.

Step 6. The gray-scale image was de-noised with the non-local means image denoising technique.

Dataset Balancing

The collected turfgrass images were divided into four classes based on the ground truth and used to develop the image classification algorithms. Class 1 included images rated ≥ 7, labeled as "High" (Figure 4a on page 44). Class 2 had images rated as 6 and 6.5, labeled as "Acceptable" (Figure 4b on page 44). Class 3 included images rated 5 and 5.5, labeled as "Poor" (Figure 4c on page 44). Class 4 included images with ratings between 1 and 4, labeled as "Very poor" (Figure 4d on page 44). A borderline between the ratings of 6 and 7, 5 and 6 could be crucial for the interest of turfgrass evaluators. The collected turfgrass image dataset was imbalanced among various rated classes due to dissimilar turfgrass quality expression under field conditions, as demonstrated in (Table 2 on page 43).

The random under-sampling technique was applied for the majority class ("Poor" class) to eliminate the majority of this class samples. An oversampling technique, the synthetic minority oversampling technique (SMOTE), was adopted for the minority classes ("High," "Acceptable," and "Very poor" classes) to solve the imbalanced data issue (Chawla et al., 2002). SMOTE was the first developed oversampling technique that became a favored method for imbalanced datasets. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). SMOTE outperforms other data augmentation methods like random oversampling, resulting in overfitting when duplicated data is used. SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data, therefore the new dataset becomes balanced. SMOTE has several variants (Kovacs, 2019), but in this study, the minority classes were oversampled with the original SMOTE. The majority class ("Poor") was under-sampled from 309 to 200 and 309 to 250 to reduce the computational time.

4.2.3 Feature Extraction with Pre-trained Convolutional Neural Networks

Selecting the right set of features for machine learning algorithms has always been a challenging task (Simon & Uma, 2020). Deep learning models can process complex input data without giving specific pre-configurations to the traditional feature extractors and without the need for feature selection. Deep learning can be used as a powerful feature extraction method and combined with traditional machine learning techniques. Features have a central role in machine learning systems, but features are detailed data. Deep learning-based feature extraction can effectively compress this complicated data in a lower dimensional representation without losing information (Maggipinto et al., 2019). The deep neural network can capture a high level of features from the initial convolution layers. Then other layers are promoted to generate a high level of features in the last layers of the DNN architecture (Simon & Uma, 2020).

In this study, pre-trained Convolutional Neural Networks (CNN) based on transfer learning, such as VGG Networks, Inception, Xception, ResNet, NASNet, and EfficientNet, were used. These pre-trained models are among the top pre-trained models with the highest accuracy rates (Misimi et al., 2017; Chen et al., 2022; Kounalakis et al., 2019; Astani et al., 2022; Cruz et al., 2019; d'Andrimont et al., 2022; Bollis et al., 2022); therefore, they were selected for this study. Using pre-trained CNN models is faster and simpler than training a CNN model with randomly initialized weights. Even though the before-mentioned pre-trained models achieved the highest accuracy rates, their accuracy rate is not significantly different from each other. To the best of the author's knowledge, there has not been enough study on deep learning-based feature extraction for turfgrass classification. This study used deep learning-based feature extraction to find the best extractor and feed the supervised machine learning classifiers. The following deep learning-based feature extractors were used: VGG16, VGG19, Xception, InceptionResNet-v2, ResNet-101, ResNet-152, NASNetLarge, EfficientNet-B0, EfficientNet-B1, and EfficientNet-B2.

VGG16 is a convolutional neural network model proposed by Simonyan and Zisserman (2014). VGG16 has an increased depth using an architecture with very small (3 × 3) convolution filters. The model has a depth of 16 weight layers (Simonyan & Zisserman, 2015) and the default input layer takes images of 224 × 224 pixels (Atila et al., 2021). VGG19 has 16 convolutional layers and 3 fully connected layers. VGG19 uses a stack of small convolutional filters of 3 × 3, followed by multiple non-linearity layers, just like VGG16. VGG19 increases the depth of the network and contributes to learning more complex features. The network depth is essential for obtaining high classification accuracy (Mahdianpari et al., 2018). The default input layer takes images of 224 × 224 pixels, like VGG16 (Atila et al., 2021).

Xception was developed by Google, Inc. (Mountain View, CA). Xception is a CNN that has 71 deep layers. Xception significantly outperforms Inception V3 on a larger image classification dataset comprising 350 million images and 17,000 classes. Xception has the same number of parameters as Inception V3 but uses the model parameters more efficiently than Inception V3 (Chollet, 2016). InceptionV3, which is the third release of Inception V1 architecture, which was developed by (Szegedy et al., 2013). The default input layer takes images of 299 × 299 pixels.

InceptionResNet-v2 is a CNN that is trained on more than a million images from the ImageNet database and is 164 layers deep. (Szegedy et al., 2015). ResNet is a residual network that has deeper layers than other neural networks. It has a deeper bottleneck architecture. ResNet is learning the residual representation functions instead of learning the signal representation directly. ResNet-101 is a convolutional neural network that is 101 layers deep, and ResNet-152 is 152 layers deep. ResNet-152 has lower complexity than VGG16 and 19 (He et al., 2015).

NASNetLarge is another CNN trained on more than a million images from the ImageNet database. NASNet can perform state-of-the-art accuracy (between 82.7% and 96.2%) with a 2.4% error rate when transferred to ImageNet classification without requiring many changes. Various versions of NASNet networks exist with different computational demands. Image features learned by NASNet can be transferred to other computational vision problems (Zoph et al., 2018). The default input layer takes images of 331 × 331 pixels.

EfficientNet is one of the state-of-the-art models that can reach 84.4% accuracy rate with 66 M parameter in the ImageNet classification problem. It achieves more efficient results by uniformly scaling depth, width, and resolution while scaling down the model. The compound scaling method searches for a grid to find the relationship between the different scaling dimensions of the baseline network under a fixed resource constraint. Then, a suitable scaling factor for depth, width and resolution dimensions will be defined. The coefficients will be applied to scale the baseline network to the desired target network

(Atila et al., 2021).

4.2.4 Image Classification

Turfgrass images were classified the same way as it was written in Section 3.2.5. Four commonly used machine learning classifiers in texture analysis were chosen in this study: Support Vector Machine (SVM), Random Forest (Sperfeld et al., 2018), XGboost, and Logistic Regression (LR). SVM, Random Forest, and LR were selected because they worked effectively for turfgrass quality ratings in the preliminary study. XGBoost classifier was selected based on its computational speed and model performance on large datasets. Figure 10 demonstrates the implementation of the image classifiers.

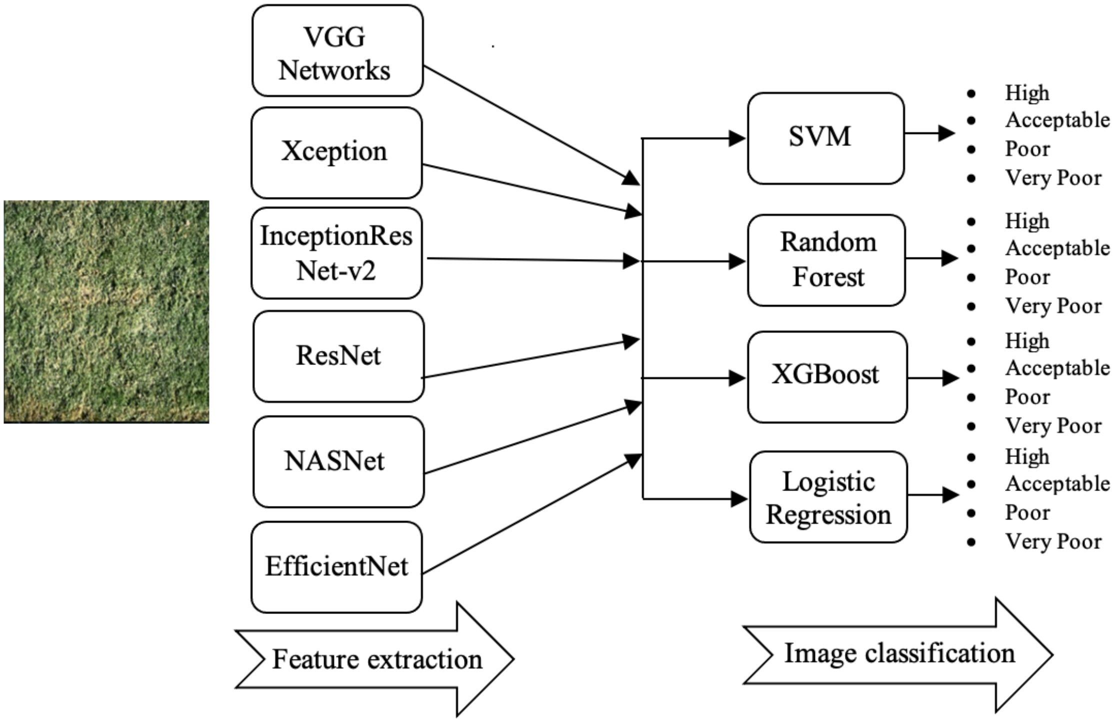


Figure 10 The procedure of the image classification.

4.2.5 Evaluation Metrics for Classifiers Performance

The same evaluation metrics were used as in Chapter III. The following metrics were used to assess the performance of the trained models on testing datasets:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{26}$$

$$Precision = \frac{TP}{TP+FP} \tag{27}$$

$$Recall = \frac{TP}{TP+FN} \tag{28}$$

$$F_1\ metric = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \tag{29}$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. The precision rate is equivalent to the proportion of correctly classified instances (Stehman, 1997). Accuracy is the proportion of correct predictions (true positives and true negatives) among the total number of cases analyzed. Precision is the fraction of relevant instances among the retrieved instances, where predictions are relevant or correct. It is the concept of consistency or the ability to group well. Recall, also known as sensitivity, is the fraction of relevant retrieved instances. F_1 metric is a weighted harmonic mean of Recall and Precision (Powers, 2019).

4.2.6 Experimental Design

Experiments (Figure 11) were carried out to test the developed image processing algorithms to find the best classifiers for turfgrass quality rating. The experiments were conducted in the open-source Anaconda Spyder (Anaconda Inc., Austin Texas, USA) environment using Python programming language. The developed algorithms ran on a computer with a processor of an 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80 GHz, and a memory of 32.0 GB RAM at the speed of 3200 MHz.

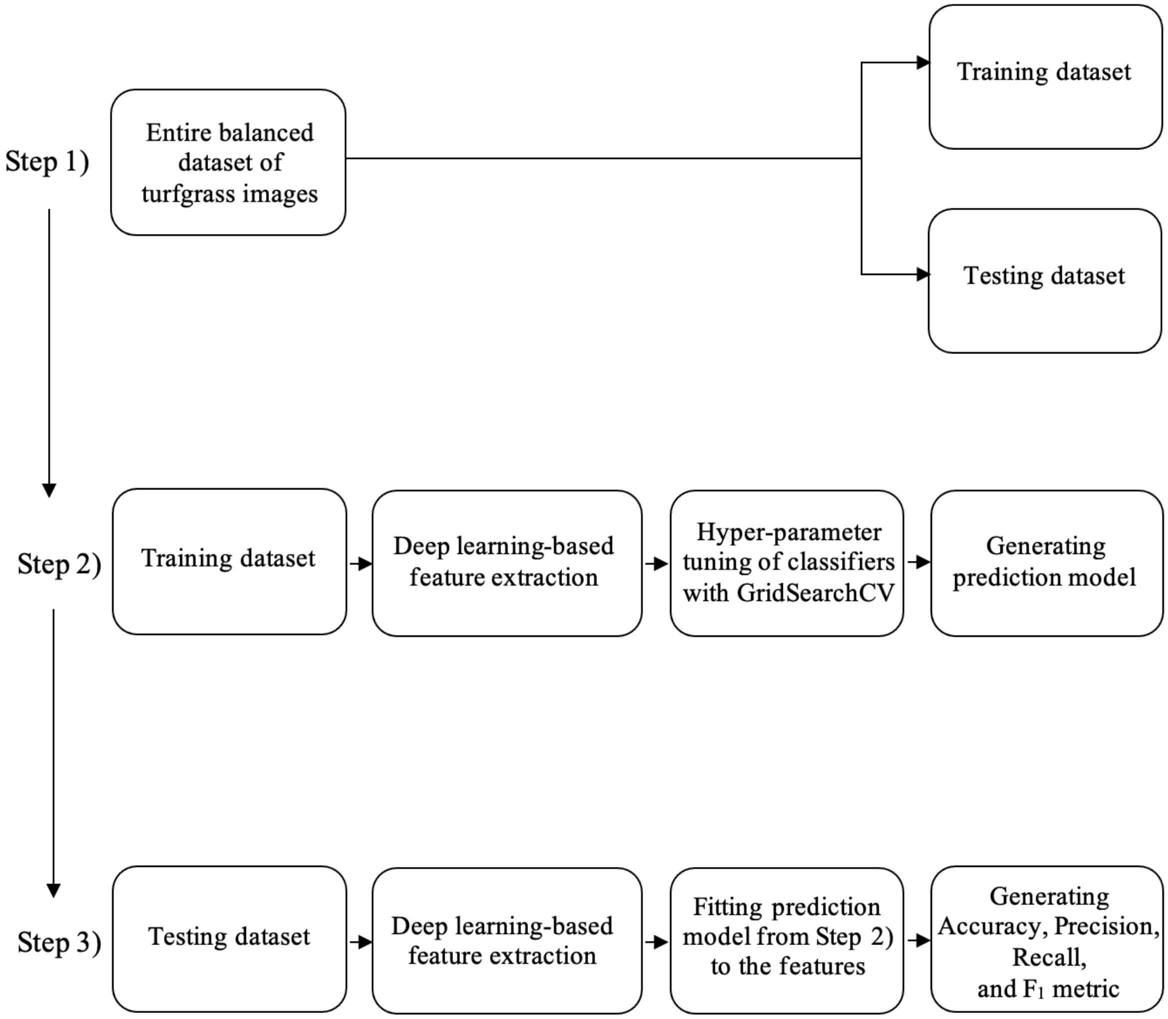

Step 1)
Entire balanced dataset of turfgrass images
Training dataset
Testing dataset
Step 2)
Training dataset
Deep learning-based feature extraction
Hyper-parameter tuning of classifiers with GridSearchCV
Generating prediction model
Step 3)
Testing dataset
Deep learning-based feature extraction
Fitting prediction model from Step 2) to the features
Generating Accuracy, Precision, Recall, and F1 metric

Figure 11 Experimental procedures for training and testing the proposed methods for turfgrass quality classification using deep features.

The dataset was randomly divided into 80% training and 20% testing datasets after dataset balancing. The training dataset was used to train the classifiers, and the testing dataset was utilized to determine the performance metrics of the best machine learning classifiers. Selecting the best deep learning-based feature extractor with the best classifier was the goal of the experiments. Another objective of this study was to compare deep feature extraction with feature extractors (Gabor filter and GLCM) utilized in machine learning from Chapter III. Two groups of tests were conducted, respectively, to classify the samples into two classes: “Acceptable” and “Poor,” and four classes: “High,” “Acceptable,” “Poor,” and “Very poor.” Ten deep-learning based feature extractors were tested. For each feature extractor method, one sample size (number of images), two group of tests, the default input image size, and four classifiers were used. Total 80 different tests were conducted (Table 9).

Table 10 The experimental setup to find the best classifiers and deep learning-based feature extractor for turfgrass quality classification.

Sample size	Number of classes to be classified	Feature extractor	Classifiers applied	Default input image size (pixels)
250	Two classes Four classes	VGG16	Logistic Regression Random forest SVM XGBoost	224 × 224
250	Two classes Four classes	VGG19	Logistic Regression Random forest SVM XGBoost	224 × 224
250	Two classes Four classes	Xception	Logistic Regression Random forest SVM XGBoost	299 × 299
250	Two classes Four classes	InceptionResNet-v2	Logistic Regression Random forest SVM XGBoost	299 × 299
250	Two classes Four classes	ResNet-101	Logistic Regression Random forest SVM XGBoost	224 × 224
250	Two classes Four classes	ResNet-152	Logistic Regression Random forest SVM XGBoost	224 × 224
250	Two classes Four classes	NASNetLarge	Logistic Regression Random forest SVM XGBoost	331 × 331
250	Two classes Four classes	EfficientNet-B0	Logistic Regression Random forest SVM XGBoost	224 × 224
250	Two classes Four classes	EfficientNet-B1	Logistic Regression Random forest SVM XGBoost	240 × 240
250	Two classes Four classes	EfficientNet-B2	Logistic Regression Random forest SVM XGBoost	260 × 260

Hyper-parameter tuning was applied to all selected classifiers to choose a set of optimal hyperparameters for better prediction results (Bishop, 1995). GridSearchCV library in Python 3.9.8 was used for hyper-parameter optimization and preventing overfitting (Chen et al., 2021). Table 4 on page 40 demonstrates the hyper-parameter tunning in the chosen classifiers. GridSearchCV library performed *k*-fold cross-validation for the training dataset. The default option of 5-fold cross-validation was used in this study.

4.3 Results and Discussion

4.3.1 Experimental Results of the Two-Class Tests

For the two classes tests ("Acceptable" and "Poor" demonstrated in Table 9), SVM achieved 91% classification accuracy rate with ResNet-101 deep learning-based feature extractor with the default image size 224 × 224 pixels. ResNet-101 deep learning-based feature extractor proved to be the best feature extractor for the sample size of 250. Logistic Regression also performed well with ResNet-101, achieving an 89% accuracy rate. XGBoost achieved 86% accuracy rate, while Random forest achieved 85% accuracy rate.

VGG16 deep learning-based feature extractor performed the second highest overall accuracy rates for turfgrass image classification. SVM achieved 89% accuracy rate, Random forest achieved 88% accuracy rates, XGBoost achieved 85%, and Logistic Regression 83% accuracy rate. EfficientNet-B1 deep learning-based feature extractor performed the third highest overall accuracy rates for turfgrass image classification. SVM and Logistic Regression achieved 87% accuracy rate, XGBoost 82% accuracy rate, and Random forest 81% accuracy rate. InceptionResNet-v2 demonstrated to be the weakest deep learning-based feature extractor with machine learning classifiers achieving the lowest accuracy rates for turfgrass image classification. SVM achieved 61% accuracy rate, Logistic Regression 56%, Random forest 55%, and XGBoost 52% accuracy rate.

Overall, SVM machine learning classifier demonstrated to be the best performing classifier with deep learning-based feature extractors, and Random forest was the weakest classifier for ResNet-101 and EfficientNet-B1 deep learning-based feature extractors. Logistic regression was the weakest classifier for VGG16 deep learning-based feature extractor.

Table 11 Results of the two classes' experiments for each classifiers and deep learning-based feature extractors in the testing dataset for a 250-sample size.

Feature extractor	**Classifier applied**	**The best accuracy for the testing dataset (%)**	**Default input image size (pixels)**
VGG16	SVM	89	224 × 224
	Random forest	88	
	XGBoost	85	
	Logistic Regression	83	
VGG19	SVM	81	224 × 224
	Random forest	79	
	XGBoost	80	
	Logistic Regression	84	
Xception	SVM	71	299 × 299
	Random forest	74	
	XGBoost	80	
	Logistic Regression	73	
InceptionResNet-v2	SVM	61	299 × 299
	Random forest	55	
	XGBoost	52	
	Logistic Regression	56	
ResNet-101	**SVM**	**91**	**224 × 224**
	Random forest	85	
	XGBoost	86	
	Logistic Regression	89	
ResNet-152	SVM	86	224 × 224
	Random forest	82	
	XGBoost	81	
	Logistic Regression	86	
NASNetLarge	SVM	79	331 × 331
	Random forest	81	
	XGBoost	82	
	Logistic Regression	79	
EfficientNet-B0	SVM	84	224 × 224
	Random forest	81	
	XGBoost	79	
	Logistic Regression	83	
EfficientNet-B1	SVM	87	240 × 240
	Random forest	81	
	XGBoost	82	
	Logistic Regression	87	
EfficientNet-B2	SVM	81	260 × 260
	Random forest	79	
	XGBoost	82	
	Logistic Regression	78	

The precision rate was higher again for the “Poor” class (95%) than the “Acceptable” (87%), as shown in Table11. It is because the original number of the rendered images before balancing for the “Poor” class was 309, while for “Acceptable” it was 192. The “Poor” class had more varieties of images than “Acceptable”, providing more images for training.

Table 12 The best classifier performed on two classes with a sample size of 250.

Classifier applied	Feature extractor	Accuracy (%)	Class	Precision (%)	Recall (%)	F_1 metric (%)
SVM	ResNet-101	91	Poor	95	87	91
			Acceptable	87	95	91

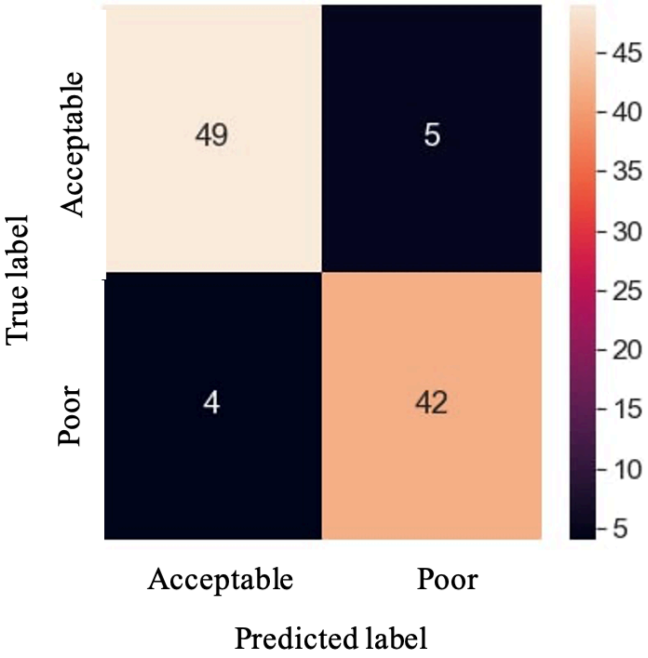


Figure 12 Confusion matrix for Support Vector Machine classifier with ResNet-101 deep learning-based feature extractor applied to two classes with a sample size of 250.

A confusion matrix (Figure 12) was generated to visualize the performance of the best classifier on turfgrass quality ratings. The confusion matrix helped to visualize each class's prediction errors within the selected classifiers. SVM machine learning classifier with ResNet-101 deep learning-based feature extractor incorrectly predicted four images as "Acceptable" and five as "Poor." Distinguishing the "Poor" quality turfgrass from "Acceptable" was challenging, particularly differentiating turfgrass rated as 5 or 6. Creating a two-classes dataset was necessary to test the capability of the proposed method in this study and to analyze the performance of the classifiers on these two critical classes ("Poor" and "Acceptable").

4.3.2 Experimental Results of the Four-Class Tests

For the four classes tests (“High,” “Acceptable,” “Poor,” and “Very poor” demonstrated in Table 13), ResNet-152 deep learning-based feature extractor achieved the highest classification accuracy rate (86%) with SVM for the testing dataset with a sample size of 250, with 224 × 224 pixels image sizes. XGBoost also performed well with ResNet-152, achieving an 85% accuracy rate. Random forest and Logistic Regression achieved 80% accuracy rates.

ResNet-101 deep learning-based feature extractor performed the second highest overall accuracy rates for turfgrass image classification. SVM achieved 84% accuracy rate, XGBoost and Logistic Regression achieved 82% accuracy rates, and Random forest 81% accuracy rate. VGG16 deep learning-based feature extractor performed the third highest overall accuracy rates for turfgrass image classification. SVM achieved 85% accuracy rate, Logistic Regression achieved 84% accuracy rate, XGBoost 81% accuracy rate, and Random forest 78% accuracy rate. InceptionResNet-v2 demonstrated to be the weakest deep learning-based feature extractor with machine learning classifiers achieving the lowest accuracy rates for turfgrass image classification. XGBoost achieved 64% accuracy rate, Logistic Regression 61%, Random forest 59%, and SVM 58% accuracy rate.

Overall, SVM machine learning classifier demonstrated to be the best performing classifier with deep learning-based feature extractors, and Random forest was the weakest classifier.

Table 13 Results of the four classes' experiments for each classifiers and deep learning-based feature extractors in the testing dataset for a 250-sample size.

Feature extractor	**Classifier applied**	**The best accuracy for the testing dataset (%)**	**Default input image size (pixels)**
VGG16	SVM	85	224 × 224
	Random forest	78	
	XGBoost	81	
	Logistic Regression	84	
VGG19	SVM	83	224 × 224
	Random forest	78	
	XGBoost	79	
	Logistic Regression	82	
Xception	SVM	81	299 × 299
	Random forest	71	
	XGBoost	76	
	Logistic Regression	77	
InceptionResNet-v2	SVM	58	299 × 299
	Random forest	59	
	XGBoost	64	
	Logistic Regression	61	
ResNet-101	SVM	84	224 × 224
	Random forest	81	
	XGBoost	82	
	Logistic Regression	82	
ResNet-152	**SVM**	**86**	**224 × 224**
	Random forest	80	
	XGBoost	85	
	Logistic Regression	80	
NASNetLarge	SVM	79	331 × 331
	Random forest	73	
	XGBoost	77	
	Logistic Regression	81	
EfficientNet-B0	SVM	84	224 × 224
	Random forest	77	
	XGBoost	80	
	Logistic Regression	82	
EfficientNet-B1	SVM	80	240 × 240
	Random forest	82	
	XGBoost	82	
	Logistic Regression	83	
EfficientNet-B2	SVM	84	260 × 260
	Random forest	79	
	XGBoost	79	
	Logistic Regression	83	

ResNet-152 deep feature extractor (Table 14) worked better for four classes and performed the best with SVM classifier. ResNet-101 worked better for two classes, possibly because ResNet-101 has 101 layers, therefore performing better on a two classes, and ResNet-152 has 152 layers performing better on four classes. Both ResNets have 224 × 224 pixels default image sizes. The precision rate was the highest (98%) for the "High" class (rated as 7), probably due to the low number of originally rendered images (17) from the 3D scans that gave less variations among the balanced images. Since our original dataset was imbalanced, SMOTE generated synthetic samples of the original 17 images to increase the number of images to 250. A high precision rate for the "High" class was expected. The second highest precision rate (90%) was achieved for the "Very poor" class (rated between 1 and 4) due to the very low quality of turfgrass samples, the texture of the turfgrass was less dense, and the turfgrass images contain bare soil. It was easy to rate them as poor quality.

There is a possibility of overfitting using SMOTE before dividing the original dataset into training and testing datasets, which could mean the training and testing datasets are dependent on each other. It is recommended in a future study to assess the likelihood of overfitting; the dataset could be divided into training and testing datasets, and then SMOTE should be applied to the training dataset. Then, these two approaches (applying SMOTE before and after the dataset was divided) could be compared. However, SMOTE is a synthetic minority oversampling technique, a data augmentation technique that generates synthetic data. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data; therefore, the new dataset becomes balanced (Gao et al., 2019), as it was described in Section 4.2.2.

Table 14 The best classifier performed on four classes with a sample size of 250.

Feature extractor	Classifier applied	Accuracy (%)	Class	Precision (%)	Recall (%)	F_1 metric (%)
ResNet-152	SVM	86	Very poor	90	90	90
			Poor	78	71	75
			Acceptable	78	84	81
			High	98	100	99

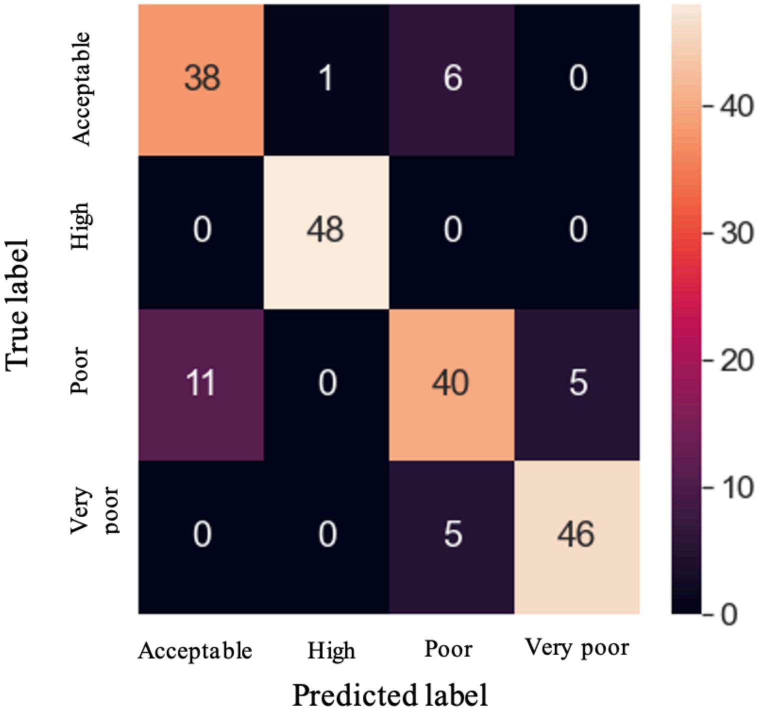


Figure 13 Confusion matrix for Support Vector Machine classifier with ResNet-152 deep learning-based feature extractor applied to four classes with a sample size of 250.

A confusion matrix (Figure 13) was generated to visualize the performance of the best classifier (SVM) on turfgrass quality classification with ResNet-152 deep learning-based feature extractor. SVM classifier with ResNet-152 deep feature extractor predicted incorrectly one image as “High” quality turfgrass, 11 images as "Acceptable”, 11 images as "Poor", and five images as “Very poor” quality turfgrass. It also might be due to the limited number of images presented for these two classes, and their quality was more pronounced than for "Poor" and "Acceptable" classes. Comparing the precision rates of

"Poor" and "Acceptable" classes when four classes were present (as shown in Table 13), the "Poor" class the same precision score (78%) as the "Acceptable" turfgrass quality. The confusion matrix (Figures 13) confirmed that distinguishing the "Poor" class from the "Acceptable" is a challenging task.

4.3.3 Comparison of machine learning- and deep learning-based feature extractors

This section focused on comparing the traditional machine learning feature extractors (Gabor filter and GLCM) utilized in Chapter III with the deep learning-based feature extractors used in this Chapter (IV). SVM demonstrated to be the best classifier with Gabor filter (for four classes) and the deep feature extractors (for both classes). Therefore, only the accuracy rates of SVM were compared in this section. Since the default input image sizes differed for each deep learning-based feature extractor (images ranged between 224 × 224 pixels to 331 × 331 pixels), the accuracy results of SVM with Gabor filter or GLCM were compared with image size 200 × 200 pixels.

Eight out of ten (Figure 14) deep learning-based feature extractors achieved higher accuracy rates for two classes than the traditional machine learning classifier (Gabor filter and GLCM), improving turfgrass quality classification on in-field images. Xception and InceptionResNet-v2 presented the same or worse performance than Gabor filter (71%) and GLCM (69%), with 71% and 61% accuracy, respectively. Table 15 demonstrates the comparison of the best machine (GLCM) and deep learning (ResNet-101) feature extractors for the two classes (“Poor” and “Acceptable”).

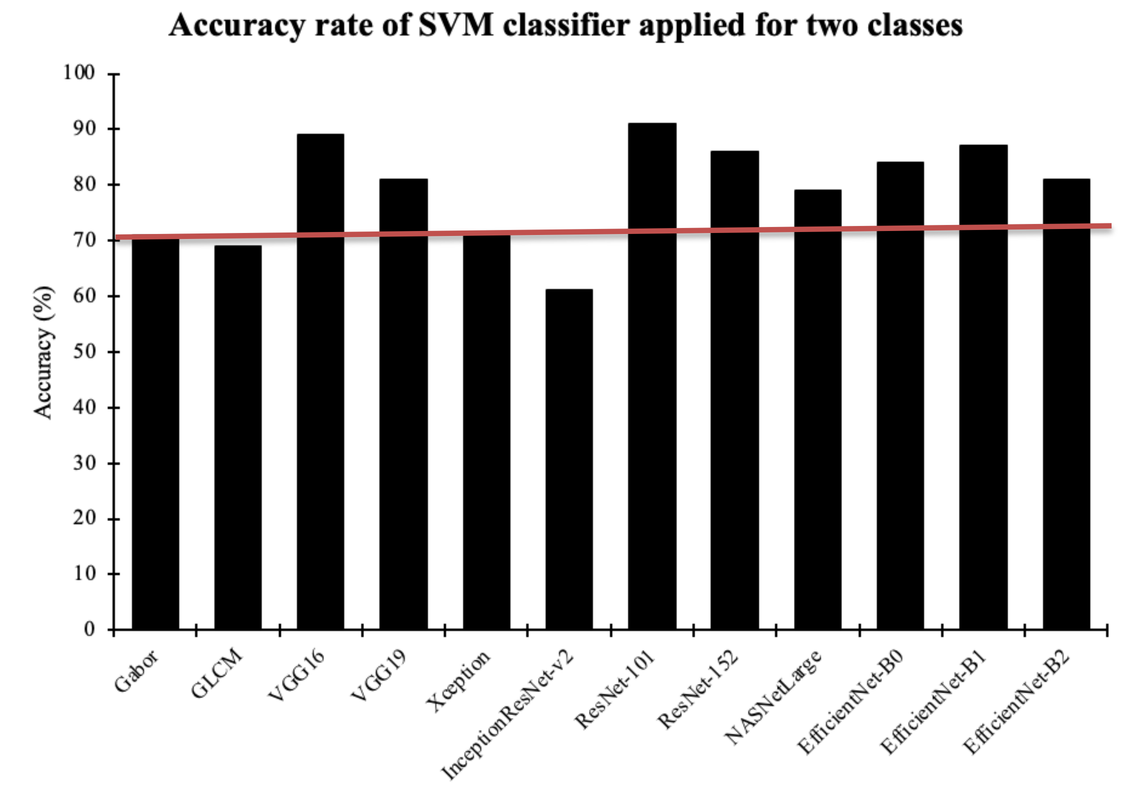


Figure 14 Accuracy rates (%) of SVM when machine and deep learning-based feature extractors were applied to two ("Acceptable" and "Poor") turfgrass quality classes. The red line was set above the machine learning-based feature extractors for comparison.

Table 15 Comparison of the best Machine and Deep Learning feature extractors applied for two classes ("Poor" and "Acceptable")

Technique	Classifier	Feature extractor	Accuracy (%)
Machine learning	Random forest	GLCM	81
Deep learning	**SVM**	**ResNet-101**	**91**

Six out of ten (Figure 15) deep learning-based feature extractors achieved higher accuracy rates for four classes than the traditional machine learning classifier (Gabor filter and GLCM. Deep learning-based feature extraction provides a better image recognition than machine learning-based feature extraction since the neural networks become deeper, which means it has more layers, and due to the complexity of the learning method (Subasi, 2020). The results demonstrated the complexity of classifying multiple classes. However, SVM can solve multiclass problems (Koklu & Ozkan, 2020), with the latest state-of-the-art classification models, it is recommended to split the multiclass problem into two binary classification problem (Sevastianov & Shchetinin, 2020). SVM presented to be an effective learning algorithm with deep features for different plant species recognition (Dourado-Filho & Calumby, 2021). Koklu and Ozkan (2020) achieved 93% accuracy rate with SVM model for classifying seven different dry bean varieties (Koklu & Ozkan, 2020).

Xception, InceptionResNet-v2, NASNetLarge and EfficientNet-B1 presented the same or worse performance than Gabor filter (81%) and GLCM (52%), with 81%, 58%, 79% and 79% accuracy, respectively. The accuracy rate did not improve that much for four classes using deep feature extractors than for two classes. Table 16 demonstrates the comparison of the best machine (Gabor filter) and deep learning (ResNet-152) feature extractors for four classes (“Very poor”, “Poor”, “Acceptable”, and “High”).

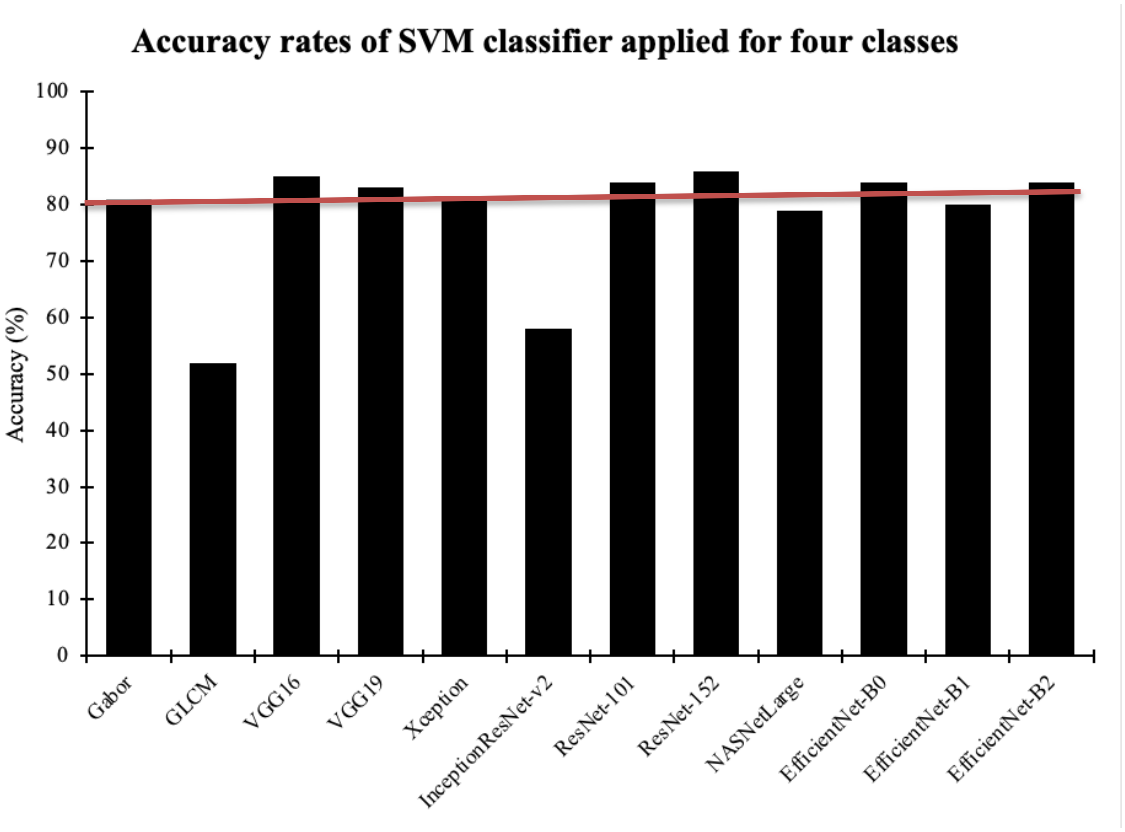


Figure 15 Accuracy rates (%) of SVM when machine and deep learning-based feature extractors were applied to four ("High", "Acceptable", "Poor" and "Very poor") turfgrass quality classes. The red line was set above the machine learning-based feature extractors.

Table 16 Comparison of the best Machine and Deep Learning feature extractors applied for four classes ("Very poor", "Poor", "Acceptable", and "High")

Technique	Classifier	Feature extractor	Accuracy (%)
Machine learning	SVM XGBoost	Gabor filter	82
Deep learning	**SVM**	**ResNet-152**	**86**

4.4 Conclusions

This study identified deep learning-based features to feed machine learning classifiers used in Task 1 (Chapter III) and compared accuracy rates with the accuracy rates achieved in Task 1 for turfgrass quality ratings. The overall goal of this research was to utilize the advantages of machine and deep learning methods for turfgrass quality classification on data collected from a research field in Oklahoma. It was the first-time deep feature extractors were utilized to feed the machine learning classifiers for turfgrass quality classification. Deep learning-based feature extractors maximized the performance of the machine learning classifiers compared to the traditional (Gabor filer and GLCM) feature extractors.

ResNet-101 deep feature extractor with SVM classifier achieved the highest precision rate (91%) for two classes ("Acceptable" and "Poor" turfgrass quality). ResNet-152 deep feature extractor with SVM classifier achieved the best precision rate (86%) for four classes ("High", "Acceptable", "Poor", and "Very poor" turfgrass quality). The performance of the developed algorithms for turfgrass quality classification was validated and reported. The presented method and pipeline will further assist researchers to develop a smartphone application and software for turfgrass quality evaluation. The software and smartphone application will assist turfgrass breeders and evaluators to conveniently rate the turfgrass in-field in real time.

CHAPTER V

REAL-TIME DETECTION OF PEANUT FLOWERS WITH THE NEW HIGH-PERFORMANCE DEEP LEARNING DETECTOR YOLOX

ABSTRACT: In several studies, deep learning algorithms demonstrated feasibility for flower detection and recognition in images. YOLOX is a new high-performance object detector that is the latest in the YOLO (You Only Look Once) deep learning network series that trains on full images and directly optimizes detection performance. No study has been done on peanut flower detection before. This study used deep learning algorithms using the latest YOLOX models to detect peanut flowers (*Arachis hypogaea*) from images collected from a peanut research station. YOLOX-L and YOLOX-X models were compared with each other with different data augmentation configurations to find the best YOLOX object detector for peanut flower detection. The original Mosaic and MixUp data augmentation strategies built-in the YOLOX models were applied to the training and validation datasets. For experimental purposes, the data augmentation was reduced or enhanced. Another experiment was when the data augmentation strategy, MixUp, was turned off or on. The mean average precision (mAP) evaluation metric was used to evaluate the performance of the YOLOX models. The original 300 epoch recommended by YOLOX developers was tested on the training/validation dataset and was reduced to 100 based on several preliminary tests. YOLOX-X achieved 90.37% mAP with weak/reduced data augmentation on the testing dataset. The average inference time for the unseen images in the testing dataset was 0.134 seconds. YOLOX-X demonstrated feasibility for detecting peanut flowers from in-field acquired images. The presented method will assist researchers in developing a counting method on peanut flowers in images and implementing the detection technique with required minor modifications for other crops or flowers.

Keywords*:* *Flower Detection, In-Field Images, Peanut flower, Reduced Data Augmentation, YOLOX Models*

5. 1. Introduction

Flower detection is challenging because it requires significant training process. Classification is another challenge to represent images of flowers mathematically (Lodh & Parekh, 2017). Another challenging issue with flower detection and counting from images is the heavily overlapped objects by leaves or another flower or fruit (Lin & Chen, 2018). Individual flower detection and accurate location determination by a vision-based system could help farmers obtain yield estimation and mapping. Knowledge of the accurate number of flowers also assists farmers in making a better decision in advance on cultivation practices and the size of the harvest (Lin & Chen, 2018).

It is possible to identify objects like flowers in images with a deep learning system. The learning subsystem is often a classifier and can detect or classify patterns in the input image (LeCun et al., 2015). Objects recognition is challenging in computer vision (Lin et al., 2017), especially for small objects. Large scale differences between object instances produce an extreme challenge to the scale invariance properties of convolutional neural networks (Singh & Davis, 2018). A few studies proposed multiple solutions to lessen the issues arising from scale variation and small object instances. These solutions include features from the layers near the input, combined with deep layers to detect small object instances. Therefore, training can be performed over various scales and several other architectural innovations (Dai et al., 2016; Lin et al., 2017; Singh & Davis, 2018; Szegedy et al., 2013).

Traditional machine vision algorithms are used to detect flowers from images taken within the lab or indoor conditions but are mainly based on the color and shape of the flowers (Li et al., 2022). Therefore, traditional machine vision algorithms are not robust enough and proved inaccurate in complex environments like images taken from the field (Tian, Chen, & Wang, 2019). Studies that used traditional machine vision techniques were focused on flowers with distinguishable colors from the background. They used complicated algorithms with many fixed thresholds that resulted in methods that are only applicable to some specific conditions (Liu et al., 2020, Fu et al., 2021). Deep learning algorithms have many advantages over traditional machine learning algorithms for object detection and recognition. With deep learning techniques, extracting and selecting discriminating features can be addressed efficiently due to the robust feature learning capabilities (Hasan et al., 2021).

YOLO (You Only Look Once) is a deep learning network structure that divides the image into sparse grids and makes multi-class and multi-scale predictions per grid cell. YOLO came out in 2016 as a new object detection approach (Redmon et al., 2016). Sparse grids are numerical techniques first introduced by Zenger (1991) that represent, integrate, and interpolate high dimensional functions. Computer algorithms use this high dimensional function to overcome the problem of dimensionality. Sparse grids prevent using classical numerical discretion schemes in more than three dimensions under the appropriate regulatory assumption (Jochen, 2012). With YOLO, object detection became a single regression problem, where a single convolutional network simultaneously can predict multiple bounding boxes and their class probabilities. This means YOLO trains on full images and directly optimizes detection performance, which results in several benefits over traditional object detection techniques (Redmon et al., 2016).

The benefits of the YOLO series are based on its architecture, which has subsets of fixed-size anchor boxes at different resolutions of feature maps at different levels inside the network for multiscale training. This makes YOLO a fast (faster than Faster R-CNN) but an accurate framework for object detection (Koirala et al., 2019). Since 2016, YOLO went through many development and improvements to the YOLO series. The speed and accuracy of YOLOv1 were improved in YOLOv2, which received a higher resolution classifier and anchor boxes (Koirala et al., 2019). YOLOv3 improved the detection of YOLOv2 by adding detection of multiple scales to help detect smaller objects, such as flowers (Mazzia et al., 2020). YOLOv3 outperformed in terms of speed Mask R-CNN (Dorrer & Tolmacheva, 2020). The accuracy of YOLOv4 was significantly higher than SSD (single shot detector) and Faster R-CNN, and the detection speed of YOLOv4 was higher than YOLOv3 (Kim, Sung, & Park, 2020). YOLOv4 achieved the best detection results on kiwifruit flowers and buds. YOLOv4 demonstrated to perform well on small objects like kiwi buds. (Li et al., 2022). Wu et al. (2020) proposed an apple flower detection method based on YOLOv4. Using YOLOv4 resulted in fast and accurate apple flower detection from in-field images. The different species of apples and the different angles of sunlight or cloudy conditions did not impact the flower detection results (Wu et al., 2020). The speed and accuracy again were improved in the YOLOv5 model.

The latest YOLO series is YOLOX, which came out in 2021. YOLOX is a new high-performance object detector. YOLOX has a YOLOv3 baseline with a decoupled head (Figure 16) to overcome the conflict between classification and regression tasks in object detection. Get et al. (2021) demonstrated through a couple of experiments that the original coupled detection head of the former YOLO detectors harmed the performance; therefore, the coupled detection head was replaced with a decoupled head that also improved the convergence speed (Ge et al., 2021). Convergence speed demonstrates how quickly the sequence of a numerical analysis approaches its limit. Faster algorithms usually use second-order information about the problem functions when calculating the search direction (Arora, 2012). A decoupled detection head in the YOLOX models is an advanced detection technique. The decoupled head was widely used for classification and localization in most of other detectors such as RetinaNet, R-CNN, and SSD (Ge et al., 2021).

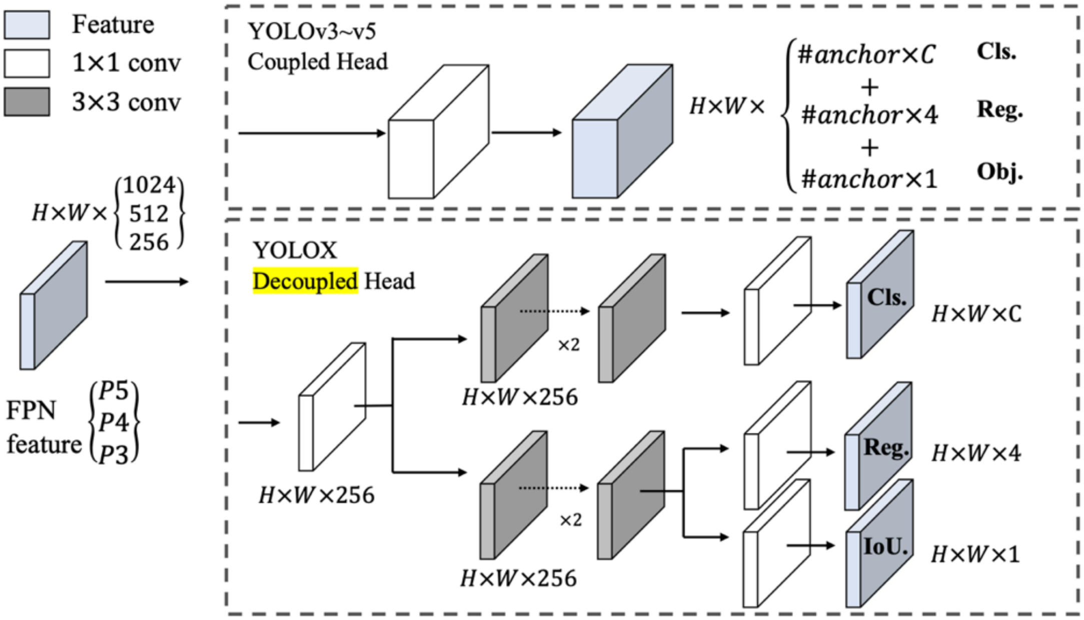


Figure 16 Illustration of the proposed decoupled detection head in the YOLOX models by Ge et al. (2021).

The architecture of YOLOX consists of three main parts: the backbone, neck, and head. The backbone refers to the feature extractor network. The neck part of YOLOX detectors operates as feature fusion. The decoupled head of YOLOX works as the classifier and regressor. The decoupled head significantly improved the converging speed and achieved better results than the original YOLOv3 head. YOLOX performance was enhanced by adding Mosaic and MixUp built-in, robust data augmentation strategies inside the object detector. Mosaic is an efficient data augmentation strategy and has been widely utilized in previous YOLO detectors (Ge et al., 2021). Mosaic is a data augmentation method that was first introduced in YOLOv4. Mosaic mixes four training images together to have four different contexts to allow detection of objects outside their normal context

(Bochkovskiy, Wang, & Liao, 2020). The MixUp was designed for image classification, but later it was modified in Bag of Freebies (BoF) for object detection (Zhang et al., 2018; Zhang et al., 2019). Bag of Freebies is an image MixUp method that was developed to improve the object detector performance (Zhang et al., 2019).

Peanut flower counts may indicate future yield. But collecting flower count data is labor-intensive and scales poorly in large fields. Detecting flowers in images collected from the field is crucial before developing a peanut flower counting method. Digital imaging combined with deep learning approaches may provide a fast and non-destructive alternative to manual flower counts in the future. This study aimed to use deep learning algorithms using YOLOX models to detect peanut flowers (*Arachis hypogaea*) from RGB images collected from a field. To the best of the author's knowledge, no study has been done on peanut flower detection; neither YOLOX deep learning model has been implemented in the agricultural field yet. YOLOX was chosen based on its capability to detect small objects such as peanut flowers.

5. 2. Material and Methods

5.2.1 Image acquisition

This study acquired RGB (red, green, and blue color model) images of peanut plants during the flowering stage from a peanut research field at Oklahoma State University (OSU) Caddo research station in Fort Cobb, Oklahoma. The research field consisted of 24 peanut canopies planted with Spanish cultivars (FloRun' 107', OLé, and Valencia C). Peanut plant images were collected three times between July 19th and August 9th, 2021, during the flowering stage (reproductive growth stage of the peanut), using a Microsoft Kinect v2 camera (Microsoft Inc., Redmond, WA, USA), capturing the full size of a peanut plot with an extra bare soil on the view. The camera settings included a field of view between 70 × 60 degrees, with a 1920 × 1080 pixels resolution. The other camera specifications can be found in Table 9. The pictures were taken during different weather conditions (sunny and cloudy) between 9 am and 11 am (Farjon et al., 2020). The changing sun angle naturally causes variation in the images that helps to develop a more robust detection system (Gao et al., 2020).

Table 17 Microsoft Kinect v2 Camera specification.

Characteristics	**Value**
Color camera resolution	1920 × 1080 pixels
Framerate	30 frames/second
Shutter type	Global shutter
Operative measuring range	From 0.5 m to 4.5 m

The data acquisition program ran on a Dell laptop (Table 15). The camera was installed on a remote-controlled vehicle (Figure 17) at 110 cm (1.1 m) above the ground to capture the full size of a peanut plot at all growing stages. The spatial resolution of the image captured by a Microsoft Kinectv2 camera was 1.4 mm × 1.4 mm when the camera is installed at a height ranging between 50 cm to 300 cm (Lachat et al., 2015; Kurillo et al., 2022). The purpose of utilizing the remote-controlled vehicle was to make data collection consistent and fast. The vehicle speed of 0.6 m/s was preliminarily determined through tests for the maximum velocity that could run over the plots while collecting clear images based on the camera framerate. Each image (Figure 18) was recorded in JPG file format on a Dell personal computer. The digital images were acquired by facing the camera vertically above the peanut plant. The size of the images was 1920 × 1080 pixels. A data acquisition program was written in Python 3.7 using libraries Pykinectv2, NumPy, and OpenCV 4.5.5 version, to collect the data streaming from the camera.

Table 18 Dell personal computer specification.

Information	
Operating system	Windows 10 Enterprise 64-bit
System model	Dell Inc. Inspiron 5567
Processor	Intel® Core™ i7-7500U CPU @ 2.70 GHz 2.90 GHz
Installed memory (Chauhan et al.)	8.00 GB
System type	64-bit Operating System, x64-based processor

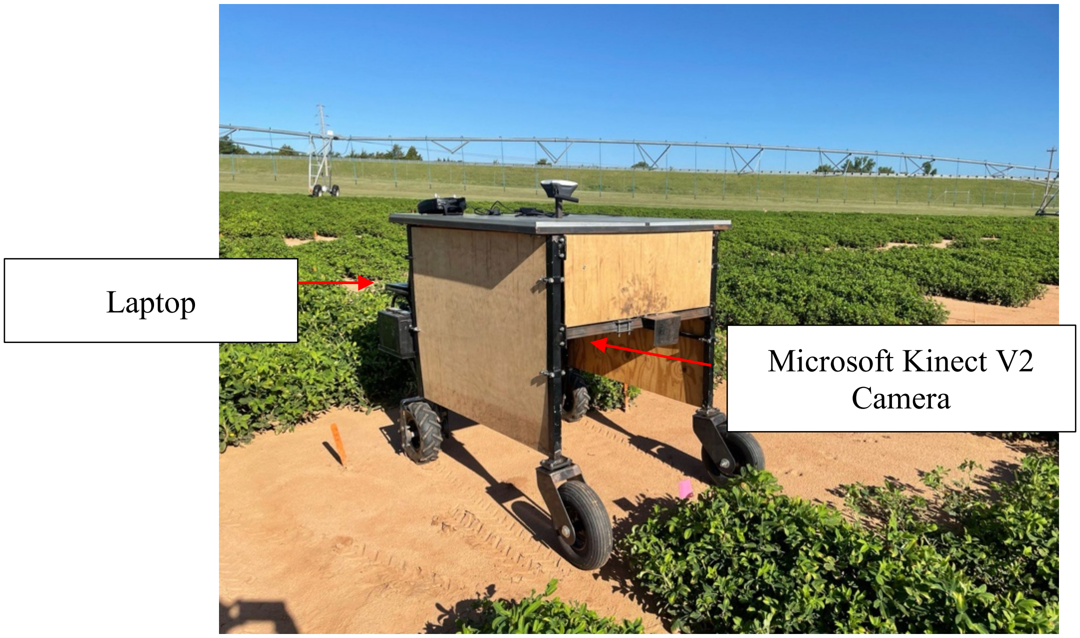


Figure 17 The remote-controlled vehicle built to capture peanut plot images.

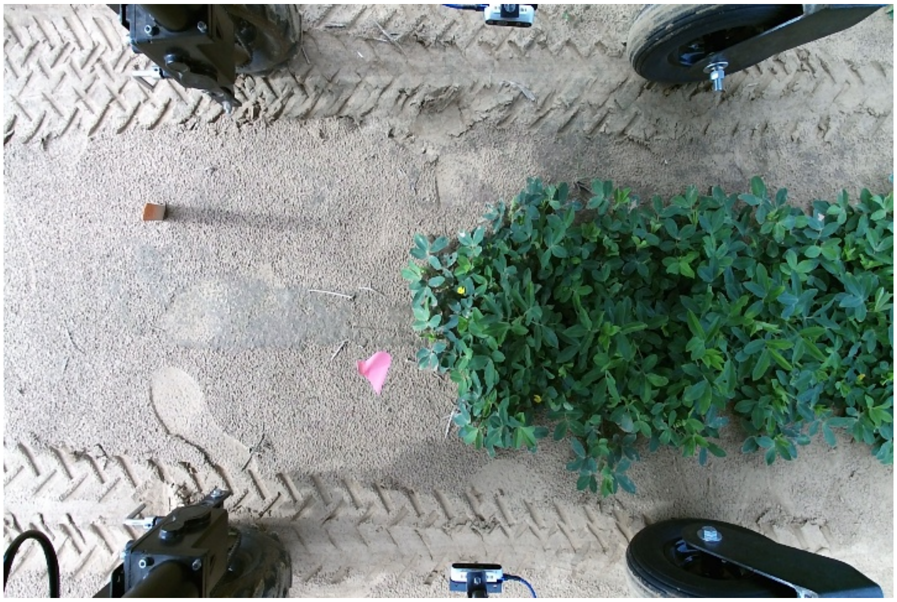


Figure 18 Example for an RGB image of the peanut plant.

5.2.2 YOLOX-L and YOLOX-X Object Detection Models

To detect peanut flowers in the selected RGB images, the new high-performance detectors YOLOX-L and YOLOX-X were chosen in this study. Ge et al. (2021) tested different backbones with different sizes for YOLOX models. The speed and accuracy of different YOLOX object detectors with 300 epochs were compared. YOLOX-L and YOLOX-X achieved the highest average precision rate (AP) with Modified CSP v5 backbones. In YOLOX detection models, Ge et al. (2021) developed an anchor-free pipeline since the original anchor-based pipeline for YOLOv3 had many problems. The anchor-free pipeline lowers the number of designed parameters and applies several heuristic tuning and maneuvers resulting in better performance. The tuning causes the training and decoding phase to be more straightforward (Ge et al., 2021)

Ge et al. (2021) appointed the center location of each object as the positive sample and pre-defined a scale range. These changes decreased the parameters of giga floating-point operations per second (GFLOPS) of the YOLOX detectors. Floating-point operations per second (FLOPS) is a computer performance unit (Society, 1985). As a result, YOLOX detectors achieved better performance and faster detection time than the earlier versions of the YOLO series (Ge et al., 2021).

The difference between YOLOX-L and YOLOX-X models is the number of parameters and GFLOPS. The number of parameters for YOLOX-L is 54.2 million, and for YOLOX-X is 99.1 million. The number of parameters improve the performance of the detector. The GFLOPS is one for YOLOX-L, and 155.6 for 281.9 YOLOX-X (Ge et al., 2021). Also, YOLOX-L and YOLOX-X models were selected based on the original YOLOX documentation. The documentation states that the YOLOX-X and L models can receive better training results for cloud or high-performance GPU (Graphics Processing Unit) deployment (Ge et al., 2021). Therefore, YOLOX-L and YOLOX-X were selected in this study to achieve better training results because the experiments (discussed in detail in the 5.2.6 section) were performed on a GPU provided by Google Colab Pro.

5.2.3 Preparation for the Object Detection Models

Images that contained peanut flowers were manually selected from the original, acquired RGB images. Images that did not contain any flowers were deleted. Due to the framerate of the camera and the speed of the remote-controlled vehicle, the same flower appeared in several images at a slightly different angle. Only one of these images was selected to prevent the object detection model from memorizing the flower rather than learning it, which leads to overfitting. With the manual selection, 176 RGB images were chosen for flower detection. The selected 176 RGB images were uploaded to Roboflow (Roboflow, Inc. Des Moines, Iowa), where the images were manually annotated as rectangles with a label called "peanut." These annotated images served as the "ground truth" for the evaluating the performance (detail in section 5.2.5) of the deep learning models. Labeling served both as training the images and providing the "ground truth". The size of the rectangles was only big enough to cover the entire peanut flower. The number of the generated annotations was 331, which means the total number of flowers in the 176 images. Then, the images were resized to 640 × 640 pixels on an OpenCV library on Python. This image resizing function utilized the nearest-neighbor interpolation method. The reason this size of the image (640 × 640 pixels) was chosen was that the original YOLOX creators suggested it for YOLOX models (Ge et al., 2021).

The bounding box is an imaginary rectangle used to identify a target, serve as a reference point (ground truth) for object detection, and create a collision box for the detected object. Therefore, this is a crucial step for training the object detection model. In solving complex problems (that come from the complexity of environmental data), the differences in bounding box sizes are inevitable because even two identical objects have different sizes on an image because of their distance and angle of the camera. In this study for peanut flower detection, there are several challenges to face with. First, there are no identical flowers in size, even in nature. Second, flowers do not bloom at the same time. Therefore, some might be fully open while others just started to open. Third, since the moving camera is used for data acquisition, even the same peanut flower in the same position in the canopy has different sizes due to the changing angle of the moving camera. Fourth, the position of the flowers compared to the camera are different because some peanut flowers are located on the bottom of the canopy or the top. Lastly, some peanut flowers might be occluded, which changes the size of the peanut flowers compared to those peanut flowers that are fully visible (not occluded). Therefore, because of the before-mentioned reasons, having the exact size of bounding boxes would mean that some annotations will not have the whole image of the flower. For smaller flowers, extra information would be given to the detection model, which leads to false and incomplete training that leads to low and false object detection results.

The labeled and resized images were exported in XML (Pascal VOC) format (recommended format for YOLOX detectors) to a notebook developed by Roboflow (Roboflow, Inc. Des Moines, Iowa) on Google Colab Pro. Google Colab is a product of Google Research that hosted the Jupyter notebook service. This notebook service does not require setup but provides access to computing resources, including GPU (https://research.google.com). As shown in Figure 19, a variety of images were collected affected by the natural light conditions that led to having a robust detection model (Gao et al., 2020). These images demonstrate the original height of the camera without zooming into images. Images were pre-processed, processed, and analyzed without zooming into the pictures.

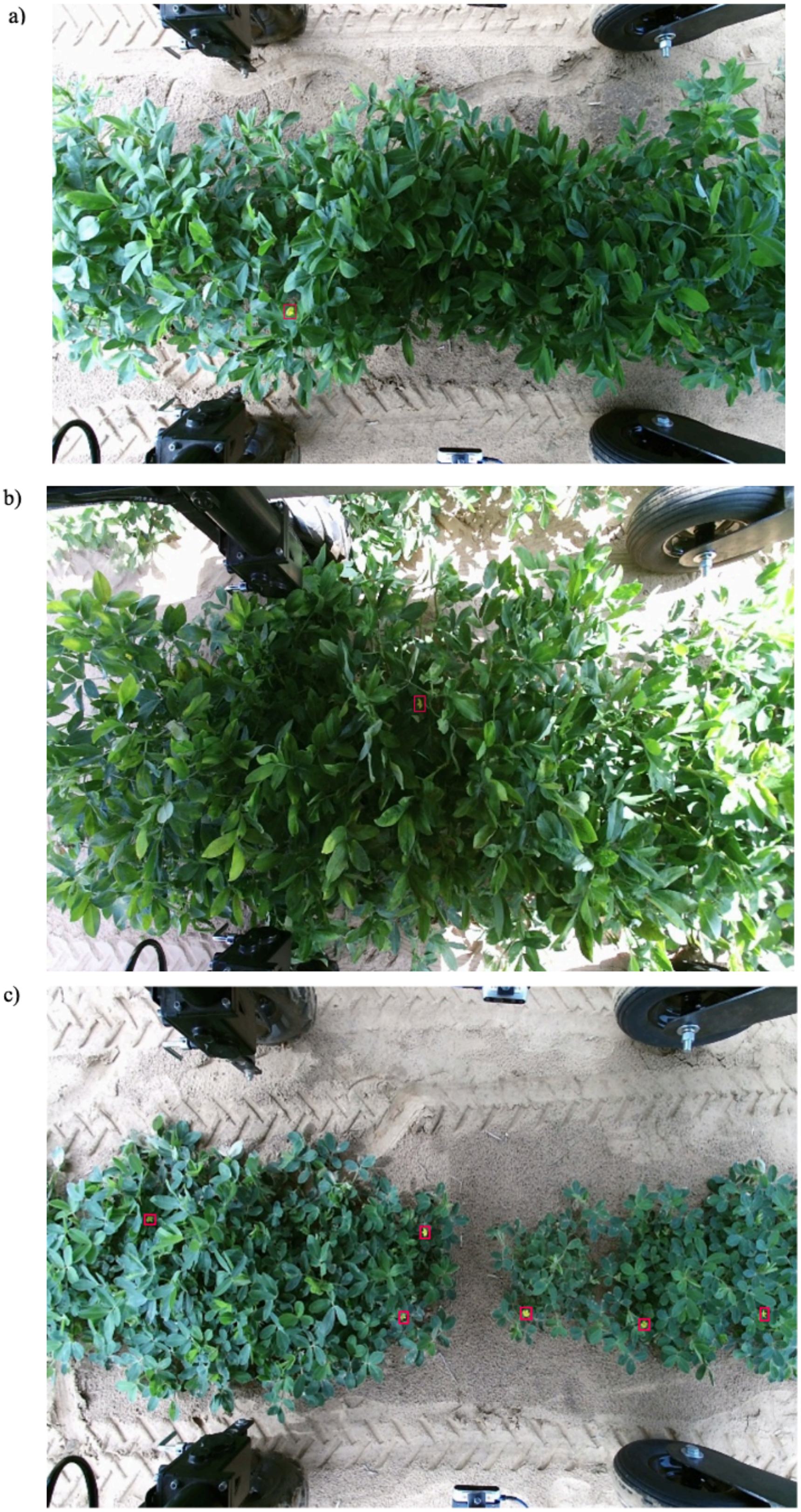


Figure 19 Examples for annotated images with a variety of natural light conditions.

These images demonstrate the original height of camera without zooming in the picture.

5.2.4 Dataset Splitting and Augmentation Configuration

In this study, 176 RGB images were divided into 80% (142) training dataset, 10% (17) validation, and 10% (17) testing using the supported VOC (Visual Object Challenge) data format for YOLOX models. Data augmentation is an important step in deep learning that helps to teach the network the preferred invariance and robustness properties (Ronneberger, Fischer, & Brox, 2015; Dosovitskiy et al., 2014). The added Mosaic and MixUp data augmentation strategies for YOLOX detectors were utilized. Mosaic data augmentation was introduced by Wei et al. (2020). Mosaic data augmentation improved the recognition ability of the deep learning detection models in images with complex backgrounds. The complex backgrounds helped to detect objects in a different context. Mosaic allows deep learning detection models to learn to identify objects at a smaller scale than other data augmentation strategies (Wei et al., 2020). Mosaic data augmentation was first introduced and utilized in the YOLOv4 model (Bochkovskiy et al., 2020), an earlier series before YOLOX models. MixUp data augmentation generates weighted combinations of random image pairs from the training data. Mosaic expands the training distribution by incorporating prior knowledge into the raw input vectors. In this way, linear interpolations of feature vectors will lead to linear interpolations of the associated targets. MixUp only needs a few lines of code and minimal computation (Zhang et al., 2018).

The built-in data augmentation configurations for YOLOX-L and YOLOX-X are shown in Table 11. The data augmentation strategies were applied to the training and validation datasets. For experimental purposes, the configurations of the data augmentation were changed to generate a reduced or an enhanced data augmentation. The configuration for the weak and strong data augmentation strategy can also be found in Table 11. The original YOLOX documentation suggested that depending on the model size, the degree of the data augmentation should be changed. A weak data augmentation works better for small models, while an enhanced data augmentation is recommended for large models or small size of datasets (Ge et al., 2021). This statement from the original YOLOX documentation has been tested.

Table 19 Data augmentation configurations.

Operation	**Values for original augmentation**	**Values for weak augmentation**	**Values for strong augmentation**
Degrees	10.0	5.0	50.0
Translate	0.1	0.05	0.5
Scale	(0.1,2)	(0.1,2)	(0.1,2)
Mosaic scale	(0.8, 1.6)	(0.8, 1.6)	(0.8, 1.6)
Shear	2.0	1.0	5.0
Perspective	0	0	0.5
Enable MixUp	True/False	False	False

5. 2. 5 Evaluation of the Performance of the Detectors

The primary performance metric to measure the accuracy of the object detectors is average precision (AP, in Equation 18). The model predictions' outcome must be determined to define average precision. These outcomes are true positive (TP), false positive (FP), and false negative (FN). In object detection, true negative (TN) does not apply due to the infinite number of bounding boxes that should not be detected on the image. True positive (TP) is the correct detection of a ground-truth bounding box (Figure 20). False positive (FP) is an incorrect detection of a non-existent object or a misplaced detection of an existing object. False negative (FN) is an undetected ground truth bounding box (Padilla, Netto, & da Silva, 2020).

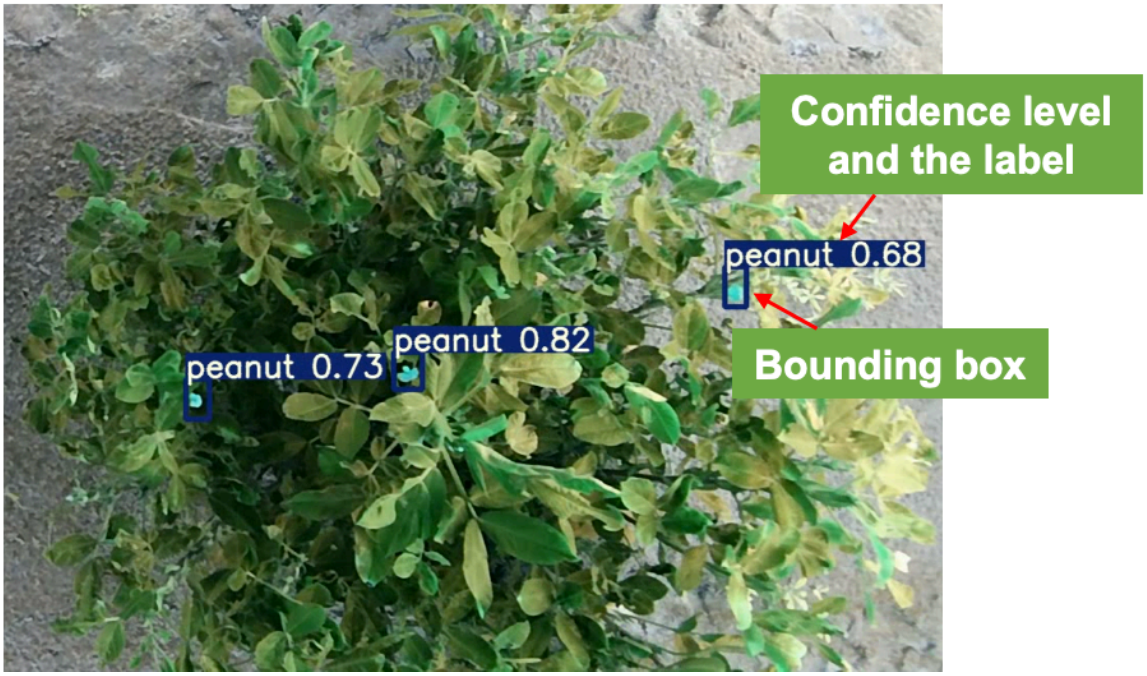
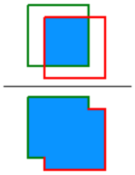


Figure 20 An example for peanut (flower) label, bounding box, and confidence score of the object detector.

The intersection over the union (IoU, in Equation 15) should be calculated to determine correct or incorrect detection. Intersection over the union is based on the Jaccard index, a coefficient of similarity for two data sets. Intersection over the union (Figure 21) measures the overlapping area between the predicted bounding box and the ground truth bounding box. In the formula, "A" refers to the ground truth bounding box, and "B" refers to the predicted bounding box. It is a number between 0 and 1, where 0 means there is not any overlap between the ground truth and predicted boxes, while 1 means a perfect overlap. (Padilla, Netto, & da Silva, 2020).

$$IoU = \frac{|A \cap B|}{|A \cup B|} \qquad (30)$$



Where the green box demonstrates the ground truth bounding box, and the red box demonstrates the prediction box.

Figure 21 Intersection over the union (IoU).

After the intersection over the union is calculated, it has to be compared with a given threshold (*t*). If IoU ≥ *t*, the detection is correct. If IoU < *t,* the detection is incorrect. The threshold value (*t*) is commonly set to 50% or greater than that (e.g., 50% to 95%) for object detection (Padilla, Netto, & da Silva, 2020). However, some studies (Bargoti and Underwood, 2017; Sa et al., 2016) suggested that IoU ≥ 20% and IoU ≥ 40% worked better to detect small flowers (like apple flowers) and relatively small fruit sizes in the images. In this study, the intersection over union (IoU) was set to thresholds of IoU ≥ 50% and IoU ≥ 50% to 95% to evaluate the performance of the YOLOX-L and YOLOX-X models and to define the detection accuracy of peanut flowers.

The evaluation of object detection methods is based on precision and recall (Padilla, Netto, & da Silva, 2020). Precision is the fraction of relevant instances among the retrieved instances, where predictions are relevant or correct. It is the concept of consistency or the ability to group well. Recall, also known as sensitivity, is the fraction of relevant retrieved instances (Stehman, 1997). Precision (P) and recall (R) can be defined as (Equations 27 and 28):

$$P = \frac{TP}{TP+FP} = \frac{TP}{all\ detections} \qquad (27)$$

$$R = \frac{TP}{TP+FN} = \frac{TP}{all\ ground\ truths} \qquad (28)$$

The shape of the precision (P) × recall (R) curve can be summarized by averaging the maximum values at a set of all equally spaced recall levels. Therefore, average precision (AP) can be obtained by interpolating the precision at each level, taking the maximum precision ($P_{interp}(R)$) whose recall value is greater than or equal R_{n+1} (each class = n) (Padilla, Netto, & da Silva, 2020).

$$AP_{all} = \sum_n (R_{n+1} - R_n) P_{interp}(R_{n+1}) \qquad (31)$$

where

$$P_{interp}(R_{n+1}) = \ max_{\widetilde{R}:\widetilde{R} \geq R_{n+1}} P(\widetilde{R}) \qquad (32)$$

In this study, mean average precision (mAP) evaluation metric was used to evaluate the performance of the YOLOX-L and YOLOX-X models. Mean average precision (mAP) is the mean of average precision (AP) over all classes. In this study, the mean average precision (mAP) rates were reported for the given thresholds of IoU ≥ 50% (mAP $_{50}$) and IoU ≥ 50% to 95% (mAP $_{50-95}$).

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \qquad (33)$$

Confidence score (Equation 34) is another evaluation metric that demonstrates the probability that the bounding box will contain the target object, which is the peanut flower in this study. Object detectors predict the location of the objects of a given class in an image with a certain confidence score (Figure 20). The confidence score reflects how likely the bounding box contains an object of interest and how confident the classifier is about it (Erhan et al., 2014).

$$Confidence = p_r(object) \times IoU \qquad (34)$$

Where p_r (object) refers to the object in the predicted bounding box.

5. 2. 6 Experimental Design

The experiments were performed on the GPU provided by Google Colab Pro. The system was initialized with a Tesla T4 professional graphics card (NVIDIA, Santa Clara, CA) and 24 GB of RAM. The YOLOX-L and YOLOX-X detectors were trained and tested on the network parameters suggested by (Ge et al., 2021) since those parameters achieved the highest performance with the selected detectors. The network initialization parameters were the following: the input image sizes were 640 pixels × 640 pixels, batch size was set up to 1, and the original recommended epoch size was reduced to 100 to overcome the overfitting issue in this study. The other training configurations can be found in Table 17. The parameters demonstrated in Table 17 correspond to the original parameters in YOLOX-L and YOLOX-X models. MixUp data augmentation strategy was turned on and off for experimental purposes. Also, the overall data augmentations were enhanced or reduced for experimental purposes based on preliminary results. The experimental design for peanut flower detection is demonstrated in Figure 22.

Table 20 Initialization parameters and training configurations for YOLOX-L and YOLOX-X detectors.

Parameter	Value
Size of the input images	640 pixels ×640 pixels
Batch	1
Momentum	0.9
Maximum epoch	100
Decay	0.0005
Basic learning/image	0.01/64
Minimum learning ratio	0.05
Classes	1

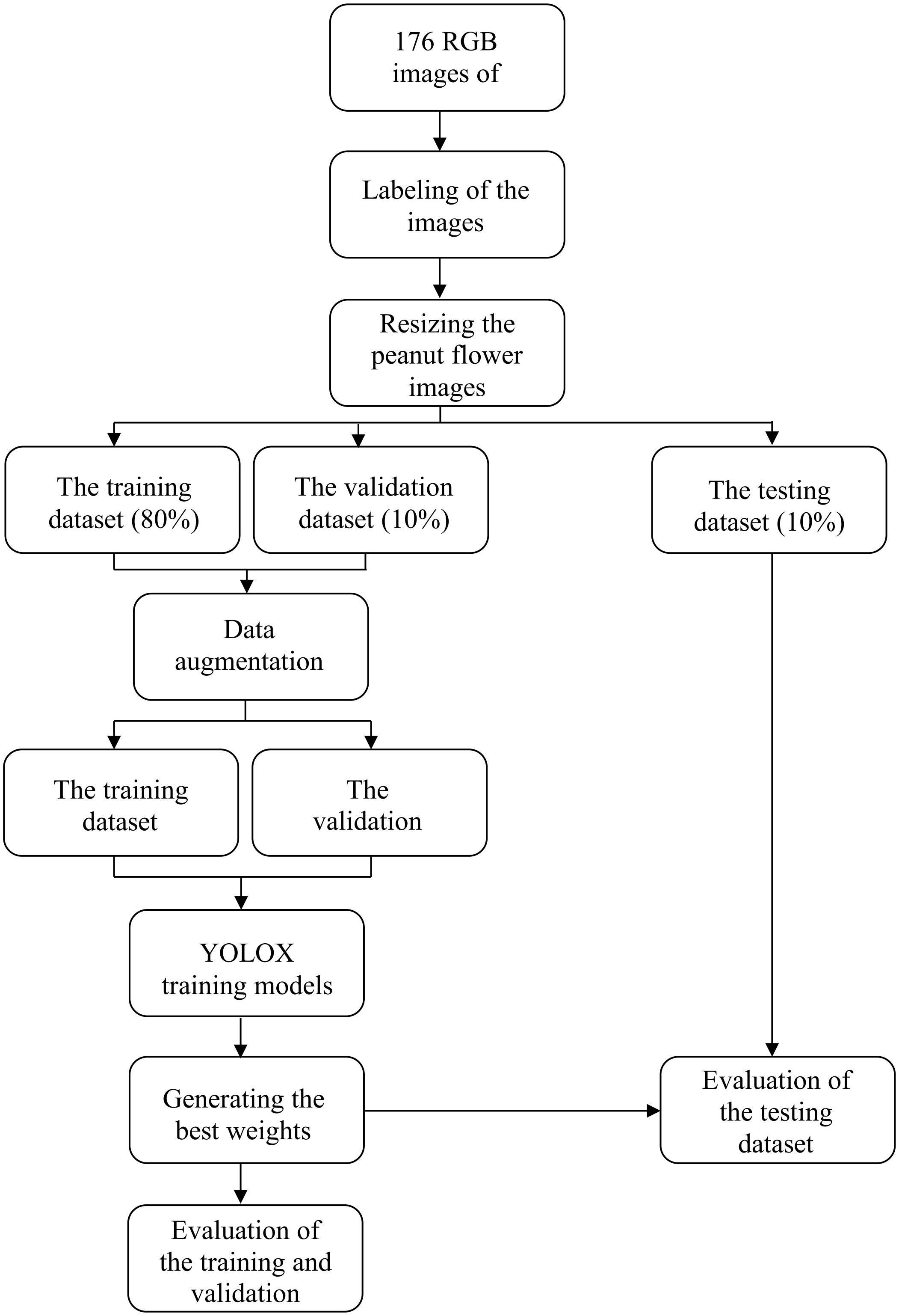

176 RGB images of
Labeling of the images
Resizing the peanut flower images
The training dataset (80%)
The validation dataset (10%)
The testing dataset (10%)
Data augmentation
The training dataset
The validation
YOLOX training models
Generating the best weights
Evaluation of the testing dataset
Evaluation of the training and validation

Figure 22 Experimental procedures for YOLOX detection models.

5. 3. Experimental Results and Discussion

To find the best YOLOX detection model for peanut flower detection, including occluded flowers, YOLOX-L and YOLOX-X were utilized and compared. In this study, the mean average precision (mAP) rates were reported for the given thresholds of IoU ≥ 50% (mAP $_{50}$) and IoU ≥ 50% to 95% (mAP $_{50\text{-}95}$).

5. 3. 1 Evaluation Metrics on the Training/Validation Dataset

Based on several preliminary tests, the original recommended 300 epoch (Ge et al., 2021) was reduced to 100. Figure 23a demonstrates the number of epochs tested on the training/validation dataset for mAP$_{50}$ values, and Figure 23b demonstrates the same tests for mAP$_{50:95}$ values. As observable in Figure 23, after reaching 100 epoch the detection model was experiencing possible overfitting.

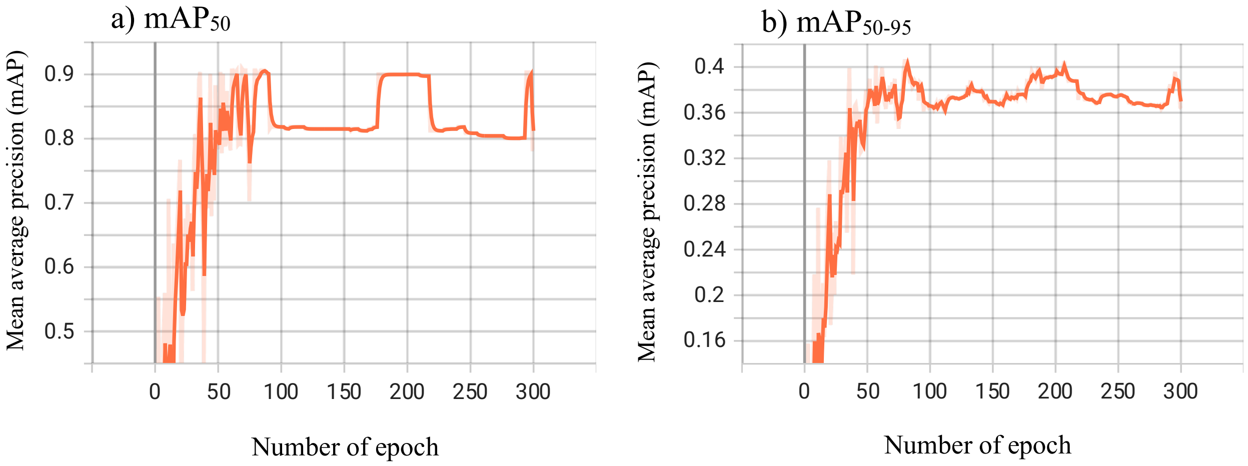


Figure 23 The tested number of epochs on the training/validation dataset for mean average precision (mAP): a) demonstrates the number of epoch tested for mAP$_{50}$, and b) for mAP$_{50\text{-}95}$ values on the training/validation dataset for YOLOX-X with the original data augmentation.

This study reported the mAP values for the training/validation dataset to compare and show no significant difference between the mAP values in the training/validation and testing dataset. It indicates that the detection model did not experience the overfitting problem. The overfitting issue in machine learning occurs when the performance on the training dataset increases but the performance decreases on the testing or unseen dataset (Hawkins, 2004). In this study, the detection model did not have any overfitting issues because

1. the original dataset was manually selected to prevent memorization,
2. the dataset was then randomly divided into training, validation, and testing datasets,
3. data augmentation technique was applied, and
4. the configuration was applied to prevent overfitting, including the modification of the epoch size recommended by the original YOLOX documentation (Ge et al., 2021). Preventing memorization is when any similar or same images were prevented from being placed in the training and testing dataset.

YOLOX-X achieved 89.81% mAP at the IoU threshold 0.5 (mAP_{50}) when MixUp was turned off. YOLOX-L achieved 80.59% mAP at the same IoU threshold without MixUp, as demonstrated in Table 20. As suggested in the original YOLOX documentation, no changes were made to the training settings first to obtain good training results (Ge et al., 2021). Then, the degree of the data augmentation was enhanced or reduced to improve the models and achieve better training results. Table 21 demonstrates both models' training/validation results with applied weak or strong data augmentation. YOLOX-X achieved 90.37% mAP at the IoU threshold 0.5 (mAP_{50}) when weak augmentation was applied, while YOLOX-L achieved 81.81%.

Table 21 Comparison of YOLOX-L and YOLOX-X models for peanut flower detection in terms of mean average precision for the training/validation dataset with and without MixUp.

Model	Image size	Parameters	GFLOP	With MixUp		Without MixUp	
				mAP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}
YOLOX-L	640 × 640 pixels	54.2 M	155.6	86.51%	42.60%	80.59%	37.41%
YOLOX-X	640 × 640 pixels	99.1 M	281.9	81.52%	38.22%	89.81%	49.56%

Table 22 Comparison of YOLOX-L and YOLOX-X models for peanut flower detection in terms of mean average precision for the training and validation dataset with weak and strong data augmentation.

Model	Image size	Parameters	GFLOP	Weak data augmentation		Strong data augmentation	
				mAP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}
YOLOX-L	640 × 640 pixels	54.2 M	155.6	81.81%	45.66%	89.94%	51.25%
YOLOX-X	640 × 640 pixels	99.1 M	281.9	90.37%	44.57%	81.81%	41.30%

5. 3. 2 Evaluation Metrics on the Testing Dataset

After the training step, the acquired optimal weight for each model was chosen to run the evaluation metrics on the testing dataset. Weights are internal adjustable parameters inside the model that will be modified to reduce the error between the output scores and the desired pattern of scores during training. Weights define the input and output function of the deep learning system. The learning algorithm computes a gradient vector. The gradient vector for each weight demonstrates how much the error would increase or decrease if the weight increased slightly. Then, the weight vector is adjusted in the opposite direction to the gradient vector (LeCun et al., 2015). The confidence score was set up to 25%. There was no significant difference between the YOLOX-L and YOLOX-X detectors during the testing when the MixUp data augmentation strategy was turned on and off (Table 22).

YOLOX-L demonstrated a higher mAP rate with the MixUp data augmentation strategy on the testing dataset than YOLOX-X and achieved similar results to the training dataset. YOLOX-X achieved better mAP results when the MixUp data augmentation strategy was turned off. The reason the detector performed better without MixUp might be because the peanut flowers were small relative to the full size of the images. It is also notable to mention that the peanut flowers appeared small when leaves occluded them. Table 23 demonstrates both models' testing results with applied weak or strong data augmentation. YOLOX-X achieved 90.37% mAP at the IoU threshold of 0.5 (mAP$_{50}$) when weak augmentation was applied, like for the training/validation dataset. YOLOX-L achieved 81.82% for the mAP$_{50}$ value same result as for the training/validation dataset. Overall, YOLOX-X with weak data augmentation achieved the highest mAP result at the IoU threshold of 0.5 (mAP$_{50}$).

Table 23 Comparison of YOLOX-L and YOLOX-X models for peanut flower detection in terms of mean average precision for the testing dataset with and without MixUp.

Model	Image size	Parameters	GFLOP	With MixUp		Without MixUp	
				mAP$_{50}$	**mAP$_{50-95}$**	**mAP$_{50}$**	**mAP$_{50-95}$**
YOLOX-L	640 × 640 pixels	54.2 M	155.6	84.83%	39.98%	80.59%	38.59%
YOLOX-X	640 × 640 pixels	99.1 M	281.9	78.12%	37.10%	**88.96%**	**48.74%**

Table 24 Comparison of YOLOX-L and YOLOX-X models for peanut flower detection in terms of mean average precision for the testing dataset with weak and strong data augmentation.

Model	Image size	Parameters	GFLOP	Weak data augmentation		Strong data augmentation	
				mAP$_{50}$	**mAP$_{50-95}$**	**mAP$_{50}$**	**mAP$_{50-95}$**
YOLOX-L	640 × 640 pixels	54.2 M	155.6	81.81%	44.71%	89.25%	50.44%
YOLOX-X	640 × 640 pixels	99.1 M	281.9	**90.37%**	**43.10%**	81.82%	42.00%

Yan et al. (2021) achieved 81.7% mAP for apple detection with a YOLOv5 model (the model before YOLOX), 82.01% mAP with a YOLOv4 model, and 71.8% mAP with a YOLOv3 model (the baseline of YOLOX) (Yan et al., 2021). Li et al. (2022) detected kiwifruits and buds comparing YOLOv3 and YOLOv4 models. They reported 80.98% mAP for a YOLOv3 model and 91.49% mAP for a YOLOv4 model with an input image size of 608 pixels ×608 pixels (Li et al., 2022). Zhou et al. (2021) achieved 89% mAP for strawberry maturity classification using a YOLOv3 model using near-ground digital camera images (Zhou et al., 2021).

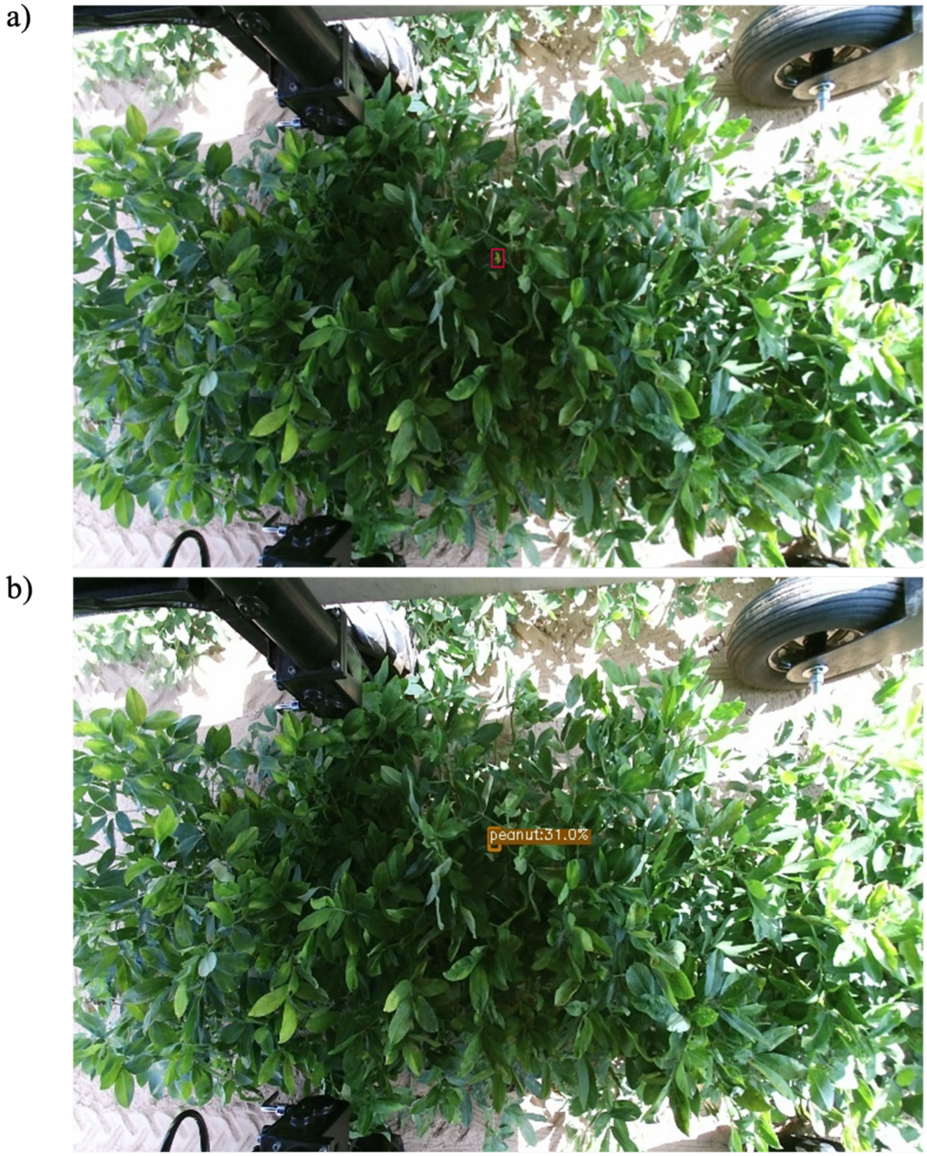


Figure 24 Peanut flower inference performed with YOLOX-X with weak augmentation. a) demonstrates the annotated image before detection, and b) is the detected peanut flower.

Figure 24a demonstrates an example for the peanut flower before inference, while Figure 24b demonstrates the correctly detected peanut flower. Inference refers to the process of predicting detections for the testing of unseen images that only require one network. The training process also involves inference since each time an image is fed into the deep neural network (DNN), the network will try to classify the image (Redmon et al., 2016). After identifying YOLOX-X as the best model for this study with weak data augmentation configurations, the average inference time was considered for possible future practical application (Figure 25). The average inference time for 17 unseen images in the testing dataset was 0.134 seconds using NVIDIA Tesla T4 GPU 24 GB of RAM provided by Google Colab Pro. Birell, Hughes, Cai, and Iida (2019) detected iceberg lettuce in 0.212 seconds. The author used a YOLOv3 model (which is the baseline of the new YOLOX models) and network was trained on a PC with a 4.5 GHz Intel i7-7700k CPU and an NVIDIA 1080Ti GeForce GTX GPU (Birrell et al., 2020).

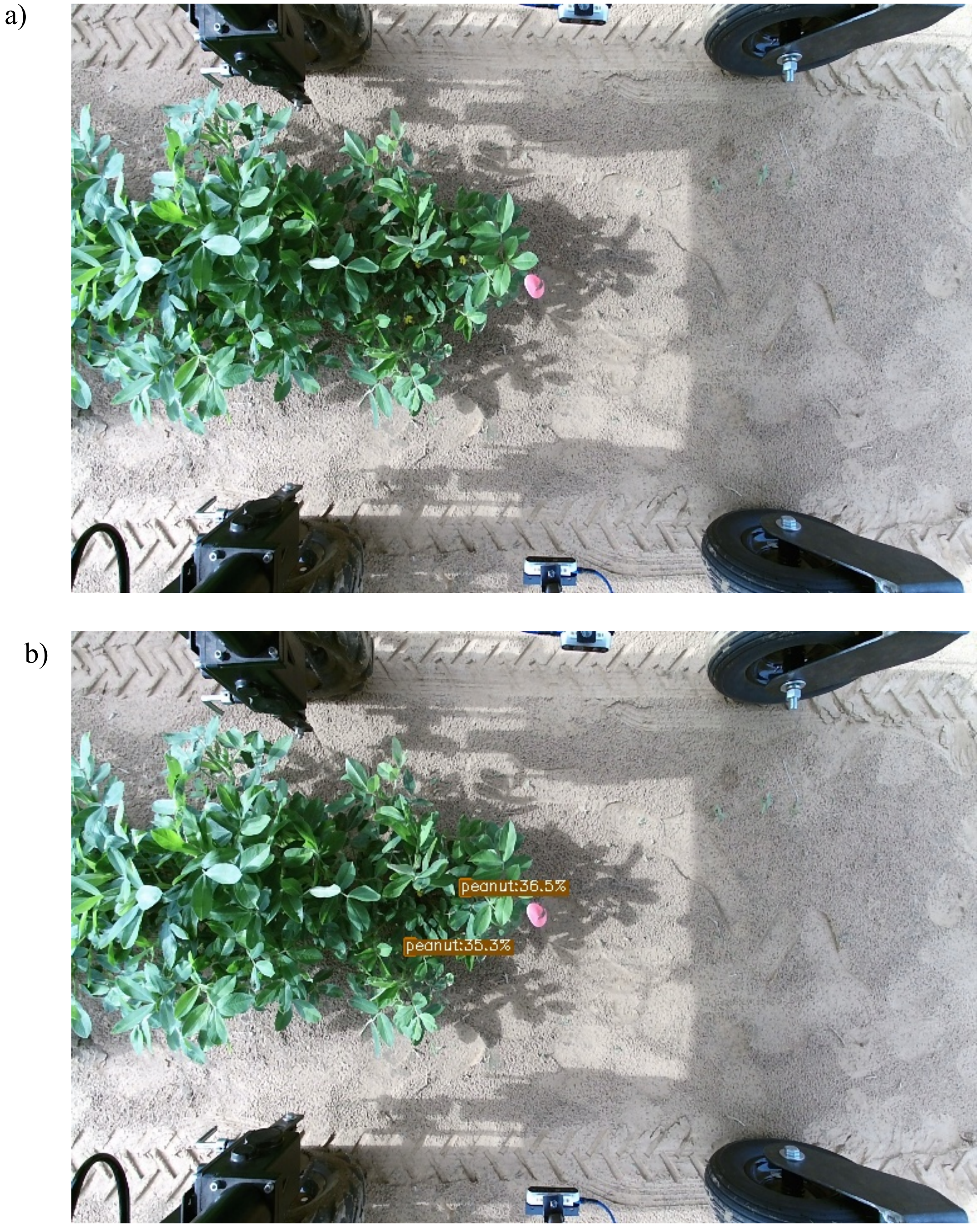


Figure 25 a) demonstrates an example for input image and b) the output image for the YOLOX-X detection model with weak data augmentation when inference time was 0.1606 s. The pink flag placed at the entry point of the peanut canopy on the field just served to identify the Spanish peanut cultivar.

5. 4. Conclusions

This study developed deep learning algorithms using YOLOX models to detect peanut flowers (*Arachis hypogaea*) from RGB images collected from a peanut research station in Oklahoma. It is the first-time peanut flowers developed during the blooming stage have been detected on RGB images. The RGB images of peanut flowers included occluded peanut flowers and visible ones. This study was also the first attempt to implement the new high-performance detector YOLOX. YOLOX-X with weak data augmentation configurations achieved the highest mean average precision result (mAP) at the IoU threshold of 0.5 (mAP_{50}). The feasibility of detecting peanut flowers from in-field acquired images with YOLOX-X was demonstrated. The suitable data augmentation configurations were tested, and the average inference time for the unseen images in the testing dataset was reported. The presented method and pipeline will further assist researchers and agricultural scientists in developing a counting method on flowers in images and implementing the detection technique with required minor modifications for other crops or flowers.

CHAPTER VI

SUMMARY AND GENERAL CONCLUSIONS

This dissertation developed in-field data acquisition systems and machine learning-based data processing and analysis approaches for turfgrass (*Cynodon* species) quality classification and peanut flower (*Arachis hypogaea*) detection. The first chapter introduced the background of the problem and need for the research study. Most of the turfgrass quality studies (Ding et al., 2016; Karcher & Richardson, 2003; Parra et al., 2020; Richardson et al., 2001) using image processing focused on the general percentage of turfgrass cover, color evaluation, and weed detection. To the best of the author's knowledge, no turfgrass image classification study has been done to distinguish turfgrass quality classes such as “Very poor”, “Poor”, “Acceptable”, and “High”. Differentiating acceptable class from poor is challenging for human evaluators but is a crucial step to do. The critical question is to investigate the possibility of distinguishing acceptable class from poor turfgrass quality using digital image processing techniques.

To the best of the author's knowledge, no study has been done on peanut flower detection on images, neither on in-field images nor collected in the lab. Neither YOLOX model has been implemented in the agricultural field before. Overall, data acquisition systems for peanut flower detection and turfgrass quality rating based entirely on Python's high-level programming language are unavailable and have not been reported yet to the best of the author's knowledge. The developed data acquisition and evaluation system in this dissertation can be applied in the future to other plants and crops, such as strawberries. The study also prepared image datasets for turfgrass quality rating and peanut flower detection, which have not been accessible in any online database before.

The second chapter introduced recent literature related to this research study. The third chapter classified different turfgrass qualities into two (“Poor” and “Acceptable”) and four classes (“Very poor,” “Poor,” “Acceptable,” “High”) from images using supervised machine learning techniques. The study aimed to identify highly significant features to feed classifiers and eventually find the best machine learning classifiers. The study also proposed a straightforward in-field data collection method, which was time and cost-effective, and convenient for an experimental field (49 m × 15 m size) that had 252 warm-season turfgrass plots in nine columns and 28 rows. The utilized small-field image acquisition tool at the same time achieved a similar outcome to the traditional image acquisition methods described in other studies. This was the first study to use a novel in-field data acquisition tool, the 3D Scanner App with Apple iPhone 12 Pro's camera with a LiDAR sensor that provided high resolution of rendered turfgrass images. The battery life lasted for the entire time of data acquisition. Using the latest and existing smart technologies for data collection, such as smartphones, can serve as valuable tools for research studies.

Random under-sampling was applied for the majority class (“Poor” class), synthetic minority oversampling technique (SMOTE) was adopted for the minority classes (“High,” “Acceptable,” and “Very poor” classes) to solve the imbalanced data issue. There is a possibility of overfitting using SMOTE before dividing the original dataset into training and testing datasets, which could mean the training and testing datasets are dependent on each other. It is recommended in a future study to assess the likelihood of overfitting. The dataset could be divided into training and testing datasets, and then SMOTE should be applied to the training dataset. Then, these two approaches (applying SMOTE before and after the dataset was divided) could be compared. However, SMOTE is a synthetic minority oversampling technique, a data augmentation technique that generates synthetic data. SMOTE generates artificial but related minority samples at the data level by randomly sampling the line segments between neighboring minority instances (Gao et al., 2019; Kovacs, 2019). SMOTE solved the overfitting issue and did not duplicate data but generated synthetic data; therefore, the new dataset becomes balanced (Gao et al., 2019).

Gabor filter and Gray Level Co-Occurrence Matrix (GLCM) feature extractors were applied to the turfgrass images to examine and compare their performance on turfgrass quality classification. Four commonly used machine learning classifiers were chosen in this study: Support Vector Machine (SVM), Random Forest (Sperfeld et al., 2018), XGboost, and Logistic Regression, to find the best classifiers for different turfgrass quality ratings.

Experiments were carried out with two (“Poor”, “Acceptable”) and four (“Very poor,” “Poor,” “Acceptable,” “High”) determined classes. GLCM feature extractor with Random forest classifier achieved the highest accuracy rate (81%) for the testing dataset for two classes (“Poor” and “Acceptable” turfgrass qualities). For four classes (“Very poor,” “Poor,” “Acceptable,” and “High” turfgrass qualities), Gabor filter proved to be the best feature extractor and performed the best with SVM and XGBoost classifiers achieving 82% accuracy rates. An unsupervised machine learning approach is advised for future turfgrass image classification. Deep learning techniques are also recommended for future turfgrass image classification studies since deep learning can outperform the accuracy rate of machine learning techniques.

The fourth chapter used the same turfgrass images rendered from the 3D scans obtained in the third chapter. Also, the same or very similar approaches were utilized for turfgrass quality classification. The study aimed to identify deep learning-based features to feed machine learning classifiers used in Chapter III and compare accuracy rates with the accuracy rates achieved in Chapter III for turfgrass's quality ratings. The overall goal of this research was to utilize the advantages of machine and deep learning methods for turfgrass quality classification on data collected from a research field in Oklahoma. Deep learning-based feature extractors maximized the performance of the machine learning classifiers compared to the traditional (Gabor filter and GLCM) feature extractors.

ResNet-101 deep feature extractor with SVM classifier achieved the highest accuracy rate (91%) for two classes (“Acceptable” and “Poor” turfgrass quality). ResNet-152 deep feature extractor with SVM classifier achieved the best accuracy rate (86%) for four classes (“High”, “Acceptable”, “Poor”, and “Very poor” turfgrass quality). The performance of the developed algorithms for turfgrass quality classification was validated and reported. The presented method and pipeline will further assist researchers to develop a smartphone application and software for turfgrass quality rating. The software and smartphone application will assist turfgrass breeders and evaluators to conveniently rate the turfgrass on-field and real time.

The fifth chapter aimed to develop deep learning algorithms to identify peanut (*Arachis hypogaea*) flowers for the Spanish varieties 107, OLE, and Val_C using the new high-performance YOLOX detection models. YOLOX-L and YOLOX-X models were compared with different data augmentation configurations to find the best YOLOX object detector for peanut flower detection. Peanut flowers (*Arachis hypogaea*) were detected from RGB images collected from a peanut research station in Oklahoma. It is the first-time peanut flowers developed during the blooming stage have been detected on RGB images. The RGB images of peanut flowers included occluded peanut flowers and visible ones.

YOLOX-X with weak data augmentation configurations achieved the highest mean average precision result (mAP) at the IoU threshold of 0.5 (mAP_{50}). The feasibility of detecting peanut flowers from in-field acquired images with YOLOX-X was demonstrated. The suitable data augmentation configurations were tested, and the average inference time for the unseen images in the testing dataset was reported. The presented method and pipeline will further assist researchers and agricultural scientists in developing a counting method on flowers in images and implementing the detection technique with required minor modifications for other crops or flowers.

<u>Overall, this dissertation contributed to:</u>

- The generation of new image datasets for a variety of turfgrass qualities and peanut flower data that have not been accessible in any online database before to the best of the author's knowledge. The online database can reduce the time spent on in-field data acquisition, which is time-consuming, limited to seasons, costly, and provides uncertain quality of images. Access to an online database can help researchers develop new methods and algorithms for data processing with already tested and good-quality data.
- Tested and validated the performance of three methods (Chapter III: supervised machine learning for classifying different turfgrass qualities; Chapter IV: supervised machine learning with deep learning-based feature extractors for classifying different turfgrass qualities, and Chapter V: using a deep learning approach to detect peanut flowers from in-field images).
- The tested and validated methods in Chapter IV for turfgrass quality classification will further assist researchers to develop a smartphone application and software for turfgrass quality rating. The software and smartphone application will assist turfgrass breeders and evaluators to conveniently rate the turfgrass on-field and real time.
- The presented method and pipeline for peanut flower detection in Chapter V will further assist researchers and agricultural scientists in developing a flower counting method in images.

- Implementing the peanut flower detection technique proposed in this dissertation will assist researchers in developing their detection method for other crops or flowers with required minor modifications.

For future work:

- Random under-sampling for the majority class (“Poor”) and SMOTE adopted for the minority classes (“High,” “Acceptable,” and “Very poor”) could be applied after dividing the dataset into training and testing datasets and compare two approaches (applying SMOTE before and after the dataset was divided) to assess the likelihood of overfitting.
- It is recommended to collect more images in the future, especially for the minority classes where turfgrasses rated 6 and above. Having more than 17 images in the “High” class would give more realistic precision rates for this class and a more negligible difference between resampling 17 images up to 200 or 250 sample sizes.
- An unsupervised machine learning approach is advised for future turfgrass image classification.
- Another approach can be transfer learning, a machine learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For instance, knowledge gained while learning to recognize plants could be applied to detecting turfgrass (Turkoglu & Hanbay, 2019).
- Deep learning techniques are also recommended for future turfgrass image classification studies since deep learning can outperform the accuracy rate of machine learning techniques.

- It is recommended to test and validate other object detectors than YOLOX models for peanut flower detection to see if the mean average precision rate can be improved.

REFERENCES

d'Andrimont, R., Yordanov, M., Martinez-Sanchez, L., & van der Velde, M. (2022). Monitoring crop phenology with street-level imagery using computer vision. Computers and Electronics in Agriculture 196, 106866, https://doi.org/10.1016/j.compag.2022.106866.

Afonso, M., Mencarelli, A., Polder, G., Wehrens, R., Lensink, D., & Faber, N. (2019). Detection of Tomato Flowers from Greenhouse Images Using Colorspace Transformations. In *Progress in Artificial Intelligence* (pp. 146-155). https://doi.org/10.1007/978-3-030-30241-2_13

Aitkenhead, M. J., Dalgetty, I. A., Mullins, C. E., McDonald, A. J. S., & Strachan, N. J. C. (2003). Weed and crop discrimination using image analysis and artificial intelligence methods. *Computers and Electronics in Agriculture*, *39*(3), 157-171. https://doi.org/https://doi.org/10.1016/S0168-1699(03)00076-0

Aleixos, N., Blasco, J., Navarrón, F., & Moltó, E. (2002). Multispectral inspection of citrus in real-time using machine vision and digital signal processors. *Computers and Electronics in Agriculture*, *33*(2), 121-137. https://doi.org/https://doi.org/10.1016/S0168-1699(02)00002-9

Almeida, A., de Villiers, J. P., De Freitas, A., & Velayudan, M. (2022). The complementarity of a diverse range of deep learning features extracted from video content for video recommendation. *Expert Systems with Applications*, *192*, 116335. https://doi.org/https://doi.org/10.1016/j.eswa.2021.116335

Angelova, A., & Zhu, S. (2013). *Efficient Object Detection and Segmentation for Fine-Grained Recognition* 2013 IEEE Conference on Computer Vision and Pattern Recognition,

Apple, I. (2022). *Augmented Reality - Apple Developer*. https://developer.apple.com/augmented-reality/

Armi, L., & Fekri-Ershad, S. (2019). Texture image analysis and texture classification methods - a review. *International Online Journal of Image Processing and Pattern Recognition 2*, 1-29. https://doi.org/10.48550/arXiv.1904.06554

Arora, J. S. (2012). Chapter 10 - Numerical Methods for Unconstrained Optimum Design, In Introduction to Optimum Design (Third Edition), 411-441

Astani, M., Hasheminejad, M., & Vaghefi, M. (2022). A diverse ensemble classifier for tomato disease recognition. Computers and Electronics in Agriculture 198, 107054, https://doi.org/10.1016/j.compag.2022.107054.

Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, *61*, 101182. https://doi.org/https://doi.org/10.1016/j.ecoinf.2020.101182

Balogh, J. C., & Walker, W. J. (1992). *Golf Course Management & Construction: Environmental Issues* (J. C. Balogh & W. J. Walker, Eds.). CRC Press.

Bargoti, S., & Underwood, J. P. (2017). Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards. *Journal of Field Robotics*, *34*.

Bird, J. J., Barnes, C. M., Manso, L. J., Ekárt, A., & Faria, D. R. (2022). Fruit quality and defect image classification with conditional GAN data augmentation. *Scientia Horticulturae*, *293*, 110684. https://doi.org/https://doi.org/10.1016/j.scienta.2021.110684

Birrell, S., Hughes, J., Cai, J. Y., & Iida, F. (2020). A field-tested robotic harvesting system for iceberg lettuce. *J Field Robot*, *37*(2), 225-245. https://doi.org/10.1002/rob.21888

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934v1 [cs.CV]* https://doi.org/10.48550/arXiv.2004.10934

Böhler, J. E., Schaepman, M. E., & Kneubühler, M. (2018). Crop Classification in a Heterogeneous Arable Landscape Using Uncalibrated UAV Data. *Remote Sensing*, *10*(8), 1282. https://www.mdpi.com/2072-4292/10/8/1282

Bollis, E., Maia, H., Pedrini, H., & Avila, S. (2022). Weakly supervised attention-based models using activation maps for citrus mite and insect pest classification, Computers and Electronics in Agriculture 195, 106839, https://doi.org/10.1016/j.compag.2022.106839.

Bonesmo , H., Kaspersen, K., & Kjersti Bakken, A. (2004). Evaluating an image analysis system for mapping white clover pastures. *Acta Agriculturae Scandinavica, Section B — Soil & Plant Science*, *54*(2), 76-82. https://doi.org/10.1080/09064710410024462

Buades, A., Coll, B., & Morel, J. (2005, 20-25 June 2005). A non-local algorithm for image denoising. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),

Burgos-Artizzu, X. P., Ribeiro, A., Tellaeche, A., Pajares, G., & Fernández-Quintanilla, C. (2010). Analysis of natural images processing for the extraction of agricultural elements. *Image and Vision Computing*, *28*(1), 138-149. https://doi.org/https://doi.org/10.1016/j.imavis.2009.05.009

Cassman, K. G. (1999). Ecological intensification of cereal production systems: Yield potential, soil quality, and precision agriculture. *Proceedings of the National Academy of Sciences*, *96*(11), 5952-5959. https://doi.org/doi:10.1073/pnas.96.11.5952

Chauhan, H., Bagyaraj, D. J., Selvakumar, G., & Sundaram, S. P. (2015). Novel plant growth promoting rhizobacteria—Prospects and potential. *Applied Soil Ecology*, *95*, 38-53. https://doi.org/https://doi.org/10.1016/j.apsoil.2015.05.011

Chawla, V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal Of Artificial Intelligence Research*, *16*, 321-357. https://doi.org/10.1613/jair.953

Chen, Y., Chen, B., Song, X., Kang, Q., Ye, X., & Zhang, B. (2021). A data-driven binary-classification framework for oil fingerprinting analysis. *Environmental Research*, *201*, 111454. https://doi.org/https://doi.org/10.1016/j.envres.2021.111454

Chen, Y., Lee, W. S., Gan, H., Peres, N., Fraisse, C., Zhang, Y., & He, Y. (2019). Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages. *Remote Sensing*, *11*(13), 1584. https://www.mdpi.com/2072-4292/11/13/1584

Chen, D., Lu, Y., Li, Z., & Young, S. (2022). Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. *Computers and Electronics in Agriculture 198*, 107091.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture*, *151*, 61-69. https://doi.org/https://doi.org/10.1016/j.compag.2018.05.012

Choi, D., Lee, W. S., Ehsani, R., Schueller, J., & Roka, F. M. (2016). Detection of dropped citrus fruit on the ground and evaluation of decay stages in varying illumination conditions. Computers and Electronics in Agriculture 127, 109-119. https://doi.org/10.1016/j.compag.2016.05.020

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *Computer Vision and Pattern Recognition (cs.CV)*. https://doi.org/10.48550/arXiv.1610.02357

Cope, J. S., Corney, D., Clark, J. Y., Remagnino, P., & Wilkin, P. (2012). Plant species identification using digital morphometrics: A review. *Expert Systems with Applications*, *39*(8), 7562-7573. https://doi.org/https://doi.org/10.1016/j.eswa.2012.01.073

Crimmins, M. A., & Crimmins, T. M. (2008). Monitoring Plant Phenology Using Digital Repeat Photography. *Environmental Management*, *41*(6), 949-958. https://doi.org/10.1007/s00267-008-9086-6

Cruz, A., Ampatzidis, Y.,Pierro, R., Materazzi, A., Panattoni, A., De Bellis, L., & Luvisi, A. (2019). Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence. Computers and Electronics in Agriculture 157, 63-76, https://doi.org/10.1016/j.compag.2018.12.028.

Dai, L., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Advances in Neural Information Processing Systems 29 (NIPS 2016)*.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, *2*(7), 1160-1169. https://doi.org/10.1364/JOSAA.2.001160

Dias, P. A., Tabb, A., & Medeiros, H. (2018). Multispecies Fruit Flower Detection Using a Refined Semantic Segmentation Network. *IEEE Robotics and Automation Letters*, *3*(4), 3003-3010. https://doi.org/10.1109/lra.2018.2849498

Ding, Raheja, Bhandari, & Green. (2016). *Application of machine learning for the evaluation of turfgrass plots using aerial images* Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping, https://doi.org/10.1117/12.2228695

Dorrer, M.; Tolmacheva, A. Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the smart refrigerator's computer vision. J. Phys. Conf. Ser. 2020, 1679, 042022.

Dosovitskiy, A., Fischer, P., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. Machine Learning; Computer Vision and Pattern Recognition; Neural and Evolutionary Computing arXiv:1406.6909

dos Santos Ferreira, A., Matte Freitas, D., Gonçalves da Silva, G., Pistori, H., & Theophilo Folhes, M. (2017). Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture*, *143*, 314-324. https://doi.org/https://doi.org/10.1016/j.compag.2017.10.027

Dourado-Filho, L. A., & Calumby, R. T. (2021). An experimental assessment of deep convolutional features for plant species recognition. *Ecological Informatics*, *65*, 101411. https://doi.org/https://doi.org/10.1016/j.ecoinf.2021.101411

Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences 465,* 1-20.

Eck, D. J. (2018). *Introduction to computer graphics. Version 1.2.* . Geneva, NY: Hobart and William Smith Colleges.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). *Scalable Object Detection Using Deep Neural Networks* 2014 IEEE Conference on Computer Vision and Pattern Recognition,

Farjon, G., Krikeb, O., Hillel, A. B., & Alchanatis, V. (2020). Detection and counting of flowers on apple trees for better chemical thinning decisions. *Precision Agriculture*, *21*(3), 503-521. https://doi.org/10.1007/s11119-019-09679-1

Flores, P., Zhang, Z., Igathinathane, C., Jithin, M., Naik, D., Stenger, J., Ransom, J., & Kiran, R. (2021). Distinguishing seedling volunteer corn from soybean through greenhouse color, color-infrared, and fused images using machine and deep learning. *Industrial Crops and Products*, *161*, 113223. https://doi.org/https://doi.org/10.1016/j.indcrop.2020.113223

Foley, J. D., & Van Dam, A. (1983). *Fundamentals of interactive computer graphics*. Princeton University Library: Addison-Wesley Pub. Co.

Frey, F. M., Robertson, A., & Bukoski, M. (2007). A method for quantifying rotational symmetry. *New Phytol*, *175*(4), 785-791. https://doi.org/10.1111/j.1469-8137.2007.02146.x

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., Al-Mallahi, A., Zhang, Q., Li, R., & Cui, Y. (2021). Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precision Agriculture*, *22*(3), 754-776. https://doi.org/10.1007/s11119-020-09754-y

Gabor, D. (1945). Theory of communication. . *British Thomson-Houston Co., Ltd.*, 429-457.

Gao, R., Peng, J., Nguyen, L., Liang, Y., Thng, S., & Lin, Z. (2019). Classification of Non-Tumorous Facial Pigmentation Disorders using Deep Learning and SMOTE. *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1-5, doi: 10.1109/ISCAS.2019.8702334.

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., & Zhang, Q. (2020). Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Computers and Electronics in Agriculture*, *176.* https://doi.org/10.1016/j.compag.2020.105634

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. *arXiv:2107.08430* https://doi.org/10.48550/arXiv.2107.08430

Gebhardt, S., Schellberg, J., Lock, R., & Kühbauch, W. (2006). Identification of broad-leaved dock (Rumex obtusifolius L.) on grassland by means of digital image processing. *Precision Agriculture*, *7*(3), 165-178. https://doi.org/10.1007/s11119-006-9006-9

Ghosal, S., Blystone, D., Singh, A. K., Ganapathysubramanian, B., Singh, A., & Sarkar, S. (2018). An explainable deep machine vision framework for plant stress phenotyping. *Proc Natl Acad Sci U S A*, *115*(18), 4613-4618. https://doi.org/10.1073/pnas.1716999115

Goap, A., Sharma, D., Shukla, A. K., & Rama Krishna, C. (2018). An IoT based smart irrigation management system using Machine learning and open source technologies. *Computers and Electronics in Agriculture*, *155*, 41-49. https://doi.org/10.1016/j.compag.2018.09.040

Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing.* (4th ed.). Pearson Education, Inc.

Gopinath, L., Moss, J. Q., & Wu, Y. (2021). Quantifying Freeze Tolerance of Hybrid Bermudagrasses Adapted for Golf Course Putting Greens. *HortScience horts*, *56*(4), 478-480. https://doi.org/10.21273/hortsci15606-20

Guijarro, M., Pajares, G., Riomoros, I., Herrera, P. J., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, *75*(1), 75-83. https://doi.org/https://doi.org/10.1016/j.compag.2010.09.013

Halko, N., Martinsson, P. G., & Tropp, J. A. (2009). Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *ACM Technical Reports, 2009-05*, 1-81.

Hansen, M. H., & Hurwitz, W. N. (1946). The Problem of Non-Response in Sample Surveys. *Journal of the American Statistical Association 41,* 517-529. http://dx.doi.org/10.1080/01621459.1946.10501894

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610-621. https://doi.org/10.1109/TSMC.1973.4309314

Hasan, A. S. M., Soher, F., Diepeveen, D., Laga, H., & Jones, M. G. K. (2021). A survey of deep learning techniques for weed detection from images. *Computer Vision and Pattern Recognition arXiv:2103.01415*. https://doi.org/10.48550/arXiv.2103.01415

Hawkins, D. M. (2004). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, *44*(1), 1-12. https://doi.org/10.1021/ci0342472

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition (cs.CV)*. https://doi.org/10.48550/arXiv.1512.03385

Hemming, J., & Rath, T. (2001). PA—Precision Agriculture: Computer-Vision-based Weed Identification under Field Conditions using Controlled Lighting. *Journal of Agricultural Engineering Research*, *78*(3), 233-243. https://doi.org/https://doi.org/10.1006/jaer.2000.0639

Herwitz, S. R., Johnson, L. F., Dunagan, S. E., Higgins, R. G., Sullivan, D. V., Zheng, J., Lobitz, B. M., Leung, J. G., Gallmeyer, B. A., Aoyagi, M., Slye, R. E., & Brass, J. A. (2004). Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. *Computers and Electronics in Agriculture*, *44*(1), 49-61. https://doi.org/https://doi.org/10.1016/j.compag.2004.02.006

Hiary, H., Saadeh, H., Saadeh, M. K., & Yaqub, M. (2018). Flower classification using deep convolutional neural networks. *IET Comput. Vis.*, *12*, 855-862.

Humeau-Heurtier, A. (2019). Texture Feature Extraction Methods: A Survey. *IEEE Access*, *7*, 8975-9000. https://doi.org/10.1109/ACCESS.2018.2890743

Janowski, A., Kaźmierczak, R., Kowalczyk, C., & Szulwic, J. (2021). Detecting Apples in the Wild: Potential for Harvest Quantity Estimation. *Sustainability*, *13*(14), 8054. https://www.mdpi.com/2071-1050/13/14/8054

Jochen, G. (2012). Sparse grids in a nutshell. Griebel, Michael (ed.). *Sparse Grids and Applications*. Springer. pp. 57–80. ISBN 978-3-642-31702-6.

Jin, X., Xu, A., Bie, R., & Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. . *Data Mining for Biomedical Applications 2006*, 106-115.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70-90. https://doi.org/https://doi.org/10.1016/j.compag.2018.02.016

Karcher, D. E., & Richardson, M. D. (2003). Quantifying Turfgrass Color Using Digital Image Analysis [https://doi.org/10.2135/cropsci2003.9430]. *Crop Science*, *43*(3), 943-951. https://doi.org/https://doi.org/10.2135/cropsci2003.9430

Kazemi, F., Golzarian, M. R., & Nematollahi, F. (2020). Quality assessment of turfgrasses using NTEP method compared to an image-based scoring system. *Journal of Ornamental Plants*, *10*, 167-178. http://jornamental.iaurasht.ac.ir/article_675375_d479fcdcf8cef90a749de9cc382599d6.pdf

Kim, J.-A., Sung, J.-Y., Park, S.-H. (2020). Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 1–4.

Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning - Method overview and review of use for fruit detection and yield estimation. Computers and Electronics in Agriculture 162, 219-234.

Koklu, M., & Ozkan, I. A. (2020). Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture*, *174*, 105507. https://doi.org/https://doi.org/10.1016/j.compag.2020.105507

Kounalakis, T., Triantafyllidis, G. A., & Nalpantidis, L. (2019). Deep learning-based visual recognition of rumex for robotic precision farming, Computers and Electronics in Agriculture 165, 104973, https://doi.org/10.1016/j.compag.2019.104973.

Kovacs, Gy. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. Applied Soft Computing 83, 105662. https://doi.org/10.1016/j.asoc.2019.105662

Kuran, U., & Kuran, E. C. (2021). Parameter selection for CLAHE using multi-objective cuckoo search algorithm for image contrast enhancement. *Intelligent Systems with Applications*, *12*, 200051. https://doi.org/https://doi.org/10.1016/j.iswa.2021.200051

Kurillo, G., Hemingway, E., Cheng, M.-L., & Cheng, L. (2022). Evaluating the accuracy of the Azure Kinect and Kinect v2. Sensors 22, 2469.

Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *Journal of Statistical Software*, *36*, 1-13. https://doi.org/10.18637/jss.v036.i11

Kursa, M. B., Jankowski, A., & Rudnicki, W. R. (2010). Boruta - A system for feature selection. Fundamenta Informaticae 101, 271-285.

Lachat, E., Macher, H., Landes, T., & Grussenmeyer, P. (2015). Assessment and calibration of a RGB-D camera (Kinectv2 sensor) towards a potential use for close-range 3D modeling. Remote sensing 7, 13070-13097.

Lanh, T. V., Chong, K. S., Emmanuel, S., & Kankanhalli, M. S. (2007). A Survey on Digital Camera Image Forensic Methods. *2007 IEEE International Conference on Multimedia and Expo*, 16-19.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*, *71*, 1-13. https://doi.org/https://doi.org/10.1016/j.patcog.2017.05.015

Li, F., Wang, C., Liu, X., Peng, Y., & Jin, S. (2018). A Composite Model of Wound Segmentation Based on Traditional Methods and Deep Neural Networks. *Computational Intelligence and Neuroscience*, *2018*, 4149103. https://doi.org/10.1155/2018/4149103

Li, G., Suo, R., Zhao, G., Gao, C., Fu, L., Shi, F., Dhupia, J., Li, R., & Cui, Y. (2022). Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. *Computers and Electronics in Agriculture*, *193*, 106641. https://doi.org/https://doi.org/10.1016/j.compag.2021.106641

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. *Sensors*, *18*(8), 2674. https://www.mdpi.com/1424-8220/18/8/2674

Liew, X. Y., Hameed, N., & Clos, J. (2021). An investigation of XGBoost-based algorithm for breast cancer classification. *Machine Learning with Applications*, *6*, 100154. https://doi.org/https://doi.org/10.1016/j.mlwa.2021.100154

Lin, P., & Chen, Y. (2018). Detection of Strawberry Flowers in Outdoor Field by Deep Neural Network. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 482-486.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *Computer Vision and Pattern Recognition arXiv:1612.03144*. https://doi.org/10.48550/arXiv.1612.03144

Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, *30*(2), 79-116. https://doi.org/10.1023/A:1008045108935

Liu, G., Nouaze, J. C., Mbouembe,P. L. T., & Kim, J. H. (2020). YOLO-tomato: a robust algorithm for tomato detection based on YOLOv3. Sensors (Switzerland), 20, 2145, 10.3390/s20072145

Lodh, A., & Parekh, R. (2017). Flower recognition system based on color and GIST features. *2017 Devices for Integrated Circuit (DevIC)*, 790-794.

Luetzenburg, G., Kroon, A., & Bjørk, A. A. (2021). Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences. *Scientific Reports*, *11*(1), 22221. https://doi.org/10.1038/s41598-021-01763-9

Lyons, M. J., Akamatsu, S., Kamachi, M. G., & Gyoba, J. (1998). Coding facial expressions with Gabor wavelets. *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 200-205.

Maggipinto, M., Beghi, A., McLoone, S., & Susto, G. A. (2019). DeepVM: A Deep Learning-based approach with automatic feature extraction for 2D input data Virtual Metrology. *Journal of Process Control*, *84*, 24-34. https://doi.org/https://doi.org/10.1016/j.jprocont.2019.08.006

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sensing*, *10*(7), 1119. https://www.mdpi.com/2072-4292/10/7/1119

Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 837-842. https://doi.org/10.1109/34.531803

Mathanker, S. K., Weckler, P. R., Bowser, T. J., Wang, N., & Maness, N. O. (2011). AdaBoost classifiers for pecan defect classification. *Computers and Electronics in Agriculture*, *77*(1), 60-68. https://doi.org/https://doi.org/10.1016/j.compag.2011.03.008

Mazzia, V., Khaliq, A., Salvetti F., & M. Chiaberge (2020). Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application. IEEE Access 8, 9102-9114.

Meyer, G. E., & Neto, J. C. (2008). Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture*, *63*(2), 282-293. https://doi.org/https://doi.org/10.1016/j.compag.2008.03.009

Misimi, E, Oye, E. R., Sture, O., Mathiassen, J. R. (2017). Robust classification approach for segmentation of blood defects in cod fillets based on deep convolutional neural networks and support vector machines and calculation of gripper vectors for robotic processing, Computers and Electronics in Agriculture 139, 138-152, https://doi.org/10.1016/j.compag.2017.05.021

Mirzapour, F., & Ghassemian, H. (2013, 14-16 May 2013). Using GLCM and Gabor filters for classification of PAN images. 2013 21st Iranian Conference on Electrical Engineering (ICEE),

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection [Methods]. *Frontiers in Plant Science*, *7*. https://doi.org/10.3389/fpls.2016.01419

Mokroš, M., Mikita, T., Singh, A., Tomaštík, J., Chudá, J., Wężyk, P., Kuželka, K., Surový, P., Klimánek, M., Zięba-Kulawik, K., Bobrowski, R., & Liang, X. (2021). Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives. *International Journal of Applied Earth Observation and Geoinformation*, *104*, 102512. https://doi.org/https://doi.org/10.1016/j.jag.2021.102512

Morris, K. N. (2021). *A Guide to NTEP Turfgrass Ratings*. https://ntep.org/reports/ratings.htm#quality

Morris, K. N., & Shearmen, R. C. (2021). *NTEP Turfgrass evaluation guidelines*. https://www.ntep.org/pdf/ratings.pdf

Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*, *114*(4), 358-371. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2012.08.009

Naeem, S., Ali, A., Chesneau, C., Tahir, M. H., Jamal, F., Sherwani, R. A. K., & Ul Hassan, M. (2021). The Classification of Medicinal Plant Leaves Based on Multispectral and Texture Feature Using Machine Learning Approach. *Agronomy*, *11*(2), 263. https://www.mdpi.com/2073-4395/11/2/263

Nisar, H., Zhou Yang, H., & Kim Ho, Y. (2016). Predicting Yield of Fruit and Flowers using Digital Image Analysis. *Indian Journal of Science and Technology*, *8*(32). https://doi.org/10.17485/ijst/2015/v8i32/93730

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2020, 2020//). Deep Learning vs. Traditional Computer Vision. Advances in Computer Vision, Cham.

Ossai, C. I., & Wickramasinghe, N. (2022). GLCM and statistical features extraction technique with Extra-Tree Classifier in Macular Oedema risk diagnosis. *Biomedical Signal Processing and Control*, *73*, 103471. https://doi.org/https://doi.org/10.1016/j.bspc.2021.103471

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transaction on Systems, Man, and Cybernetics* *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237-242, https://doi.org/10.1109/IWSSIP48289.2020.9145130

Parra, L., Marin, J., Yousfi, S., Rincón, G., Mauri, P. V., & Lloret, J. (2020). Edge detection for weed recognition in lawns. *Computers and Electronics in Agriculture*, *176*, 105684. https://doi.org/10.1016/j.compag.2020.105684

Parvathi, S., & Tamil Selvi, S. (2021). Detection of maturity stages of coconuts in complex background using Faster R-CNN model. *Biosystems Engineering*, *202*, 119-132. https://doi.org/10.1016/j.biosystemseng.2020.12.002

Pawlak, Z. (2002). Rough set theory and its applications. Journal of Telecomunications and Information Technology 3, 7-10.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Pierce, F. J., & Nowak, P. (1999). Aspects of Precision Agriculture. In D. L. Sparks (Ed.), *Advances in Agronomy* (Vol. 67, pp. 1-85). Academic Press. https://doi.org/10.1016/S0065-2113(08)60513-1

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., & Zuiderveld, K. (1987). Adaptive Histogram Equalization and Its Variations. *Computer Vision, Graphics, and Image Processing 38*, 355-368.

Powers, D. M. W. (2019). What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes. *arXiv:1503.06410*. https://doi.org/10.48550/arXiv.1503.06410

Prakash, K., & Saradha, S. (2021). Efficient prediction and classification for cirrhosis disease using LBP, GLCM and SVM from MRI images. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.03.418

Prasad, G., Vijay, G. S., & Kamath C, R. (2022). Comparative study on classification of machined surfaces using ML techniques applied to GLCM based image features. *Materials Today: Proceedings*. https://doi.org/https://doi.org/10.1016/j.matpr.2022.01.285

Qiao, X., Bao, J., Zhang, H., Zeng, L., & Li, D. (2017). Underwater image quality enhancement of sea cucumbers based on improved histogram equalization and wavelet transform. *Information Processing in Agriculture*, *4*(3), 206-213. https://doi.org/https://doi.org/10.1016/j.inpa.2017.06.001

Ramenol, E., Caballero, Y., Belloand, R., & Herrera, F. (2012). SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and Information Systems 33*(2), 245-265.

Rath, T., & Kawollek, M. (2009). Robotic harvesting of Gerbera Jamesonii based on detection and three-dimensional modeling of cut flower pedicels. *Computers and Electronics in Agriculture*, *66*(1), 85-92. https://doi.org/https://doi.org/10.1016/j.compag.2008.12.006

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, 27-30 June 2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Reza, A. M. (2004). Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, *38*(1), 35-44. https://doi.org/10.1023/B:VLSI.0000028532.53893.82

Richardson, M. D., Karcher, D. E., & Purcell, L. C. (2001). Quantifying Turfgrass Cover Using Digital Image Analysis [https://doi.org/10.2135/cropsci2001.1884]. *Crop Science*, *41*(6), 1884-1888. https://doi.org/https://doi.org/10.2135/cropsci2001.1884

Rodriguez-Galiano, V. F., Chica-Olmo, M., Abarca-Hernandez, F., Atkinson, P. M., & Jeganathan, C. (2012). Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture. *Remote Sensing of Environment*, *121*, 93-107. https://doi.org/https://doi.org/10.1016/j.rse.2011.12.003

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer International Publishing Switzerland 2015, LNCS 9351, 234–241, DOI: 10.1007/978-3-319-24574-4_28

Rorie, R., Purcell, L., Karcher, D., & King, A. (2011). The Assessment of Leaf Nitrogen in Corn from Digital Images. *Crop Science*, *51*, 2174. https://doi.org/10.2135/cropsci2010.12.0699

Rusk, N. (2016). Deep learning. *Nature Methods*, *13*(1), 35-35. https://doi.org/10.1038/nmeth.3707

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool C. (2016). Deepfruits: a fruit detection system using deep neural networks, *Sensors, 16,* 10.3390/s16081222

Saponara, S., & Elhanashi, A. (2022). Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance. Part of the Lecture Notes in Electrical Engineering book series (LNEE,volume 866).

Seeland, M., Rzanny, M., Alaqraa, N., Waldchen, J., & Mader, P. (2017). Correction: Plant species classification using flower images—A comparative study of local feature representations. *PLOS ONE*, *12*(3), e0175101. https://doi.org/10.1371/journal.pone.0175101

Septiarini, A., Sunyoto, A., Hamdani, H., Kasim, A. A., Utaminingrum, F., & Hatta, H. R. (2021). Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features. *Scientia Horticulturae*, *286*, 110245. https://doi.org/https://doi.org/10.1016/j.scienta.2021.110245

Sevastianov, L. A., & Shchetinin, E. Y. (2020). On methods for improving the accuracy of multi-class classification on imbalanced data. *CEUR Workshop Proceedings*.

Shi, Y., Wang, N., Taylor, R. K., & Raun, W. R. (2015). Improvement of a ground-LiDAR-based corn plant population and spacing measurement system. *Computers and Electronics in Agriculture*, *112*, 92-101. https://doi.org/https://doi.org/10.1016/j.compag.2014.11.026

Simon, P., & Uma, V. (2020). Deep Learning based Feature Extraction for Texture Classification. *Procedia Computer Science 171*, 1680–1687. https://doi.org/10.1016/j.procs.2020.04.180

Simonyan, K., & Zisserman, A. (2015). *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION* ICLR 2015, http://www.robots.ox.ac.uk/˜vgg/research/very_deep/

Singh, B., & Davis, L. S. (2018). An Analysis of Scale Invariance in Object Detection - SNIP. *Computer Vision and Pattern Recognition arXiv:1612.03144*. https://doi.org/10.48550/arXiv.1711.08189

Sinha, A., & Singh Shekhawat, R. (2020). A novel image classification technique for spot and blight diseases in plant leaves. *The Imaging Science Journal*, *68*(4), 225-239. https://doi.org/10.1080/13682199.2020.1865652

Society, I. C. (1985). IEEE Standard for Binary Floating-Point Arithmetic. In *An American National Standard*.

Sperfeld, M., Rauschenbach, C., Diekert, G., & Studenik, S. (2018). Microbial community of a gasworks aquifer and identification of nitrate-reducing Azoarcus and Georgfuchsia as key players in BTEX degradation. *Water Research*, *132*, 146-157. https://doi.org/https://doi.org/10.1016/j.watres.2017.12.040

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), 77-89. https://doi.org/https://doi.org/10.1016/S0034-4257(97)00083-7

Stier, J. C., Horgan, B. P., & Bonos, S. A. (2013). *Turfgrass Biology Use and Management* (Vol. 56). Agronomy Monograph. https://doi.org/10.2134/agronmonogr56

Subasi, A. (2020). Chapter 5 - Other classification examples in *Practical Machine Learning for Data Analysis Using Python*, Academic Press, 323-390, ISBN 9780128213797, https://doi.org/10.1016/B978-0-12-821379-7.00005-9

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep Neural Networks for Object Detection. *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2015). Rethinking the Inception Architecture for Computer Vision. *Computer Vision and Pattern Recognition (cs.CV)*. https://doi.org/10.48550/arXiv.1512.00567

Talebi, H., & Milanfar, P. (2021). Learning to Resize Images for Computer Vision Tasks. *Computer Vision and Pattern Recognition; Machine Learning, 1-10.* arXiv:2103.09950

Tavakoli, H., Alirezazadeh, P., Hedayatipour, A., Banijamali Nasib, A. H., & Landwehr, N. (2021). Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks. *Computers and Electronics in Agriculture*, *181*, 105935. https://doi.org/https://doi.org/10.1016/j.compag.2020.105935

Tavani, S., Billi, A., Corradetti, A., Mercuri, M., Bosman, A., Cuffaro, …, & Carminati, E. (2022). Smartphone assisted fieldwork: Towards the digital transition of geoscience fieldwork using LiDAR-equipped iPhones. Earth Science Reviews 227, 1039699. https://doi.org/10.1016/j.earscirev.2022.103969

Thorp, K. R., & Tian, L. F. (2004). A Review on Remote Sensing of Weeds in Agriculture. *Precision Agriculture*, *5*(5), 477-508. https://doi.org/10.1007/s11119-004-5321-1

Tian, M., Chen, H., & Wang, Q. (2019). Detection and recognition of flower image based on SSD network in video stream. J. Phys. Conf. Ser., 1237 (3), 032045, https://doi.org/10.1088/1742-6596/1237/3/032045

Turkoglu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering & Computer Sciences*, *27*(3), 1636-1651. https://doi.org/10.3906/elk-1809-181

Wang, J., Zhang, Y., Wang, J., Zhao, K., Li, X., & Liu, B. (2022). Using machine-learning technique for estrus onset detection in dairy cows from acceleration and location data acquired by a neck-tag. *Biosystems Engineering*, *214*, 193-206. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2021.12.025

Wang, X., Tang, J., & Whitty, M. (2021). DeepPhenology: Estimation of apple flower phenology distributions based on deep learning. *Computers and Electronics in Agriculture*, *185*, 106123. https://doi.org/https://doi.org/10.1016/j.compag.2021.106123

Wei, Z., Duan, C., Song, X., Tian, Y., & Wang, H. (2020). AMRNET: CHIP AUGMENTATION IN AERIAL IMAGE OBJECT DETECTION. *arXiv:2009.07168v2 [cs.CV]*. https://arxiv.org/pdf/2009.07168.pdf

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. J*ournal of the Royal Statistical Society Series B (Statistical Methodology) 73*(1), 3-36.

Wu, D., Lv, S., Jiang, M., & Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture 178. 105742.

Yan, B., Fan, P., Lei, X., Liu, Z., & Yang, F. (2021). A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sensing*, *13*(9), 1619. https://www.mdpi.com/2072-4292/13/9/1619

Yang, B., & Xu, Y. (2021). Applications of deep-learning approaches in horticultural research: a review. *Horticulture Research*, *8*(1), 123. https://doi.org/10.1038/s41438-021-00560-9

Yang, L., Ando, D., Hoshino, Y., Suzuki, S., & Cao, Y. (2017). Detection of the pumpkin flower to estimate its fruit position using a colour camera. *7th Asian-Australasian Conference on Precision Agriculture*, *1-11*.

Yu, J., Sharpe, S. M., Schumann, A. W., & Boyd, N. S. (2019). Deep learning for image-based weed detection in turfgrass. *European Journal of Agronomy*, *104*, 78-84. https://doi.org/https://doi.org/10.1016/j.eja.2019.01.004

Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *Computer Vision and Pattern Recognition* arXiv:1311.2901, https://doi.org/10.48550/arXiv.1311.2901

Zenger, C. (1991). Sparse grids, in Parallel Algorithms for Partial Differential Equa- tions (W. Hackbusch, ed.), Vol. 31 of Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig/Wiesbaden.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond Emperical Risk Minimization. *ICLR*.

Zhang, J., Marszałek, M., Lazebnik, S., & Schmid, C. (2007). Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, *73*(2), 213-238. https://doi.org/10.1007/s11263-006-9794-4

Zhang, Z., He, T., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2019). Bag of Freebies for Training Object Detection Neural Networks. *arXiv:1902.04103* https://doi.org/10.48550/arXiv.1902.04103

Zhou, X., Lee, W. S., Ampatzidis, Y., Chen, Y., Peres, N., & Fraisse, C. (2021). Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning. *Smart Agricultural Technology*, *1*, 100001. https://doi.org/https://doi.org/10.1016/j.atech.2021.100001

Zhu, J., Hong, R., Zhang, H., Gu, R., Wang, H., & Sun, F. (2022). Fired bullet signature correlation using the finite ridgelet transform (FRIT) and the gray level co-occurrence matrix (GLCM) methods. *Forensic Science International*, *330*, 111089. https://doi.org/https://doi.org/10.1016/j.forsciint.2021.111089

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *omputer Vision and Pattern Recognition (cs.CV); Machine Learning (cs.LG); Machine Learning (stat.ML)*. https://doi.org/10.48550/arXiv.1707.07012

Zuiderveld, K. J. (1994). Contrast Limited Adaptive Histogram Equalization. Graphics Gems,

APPENDICES

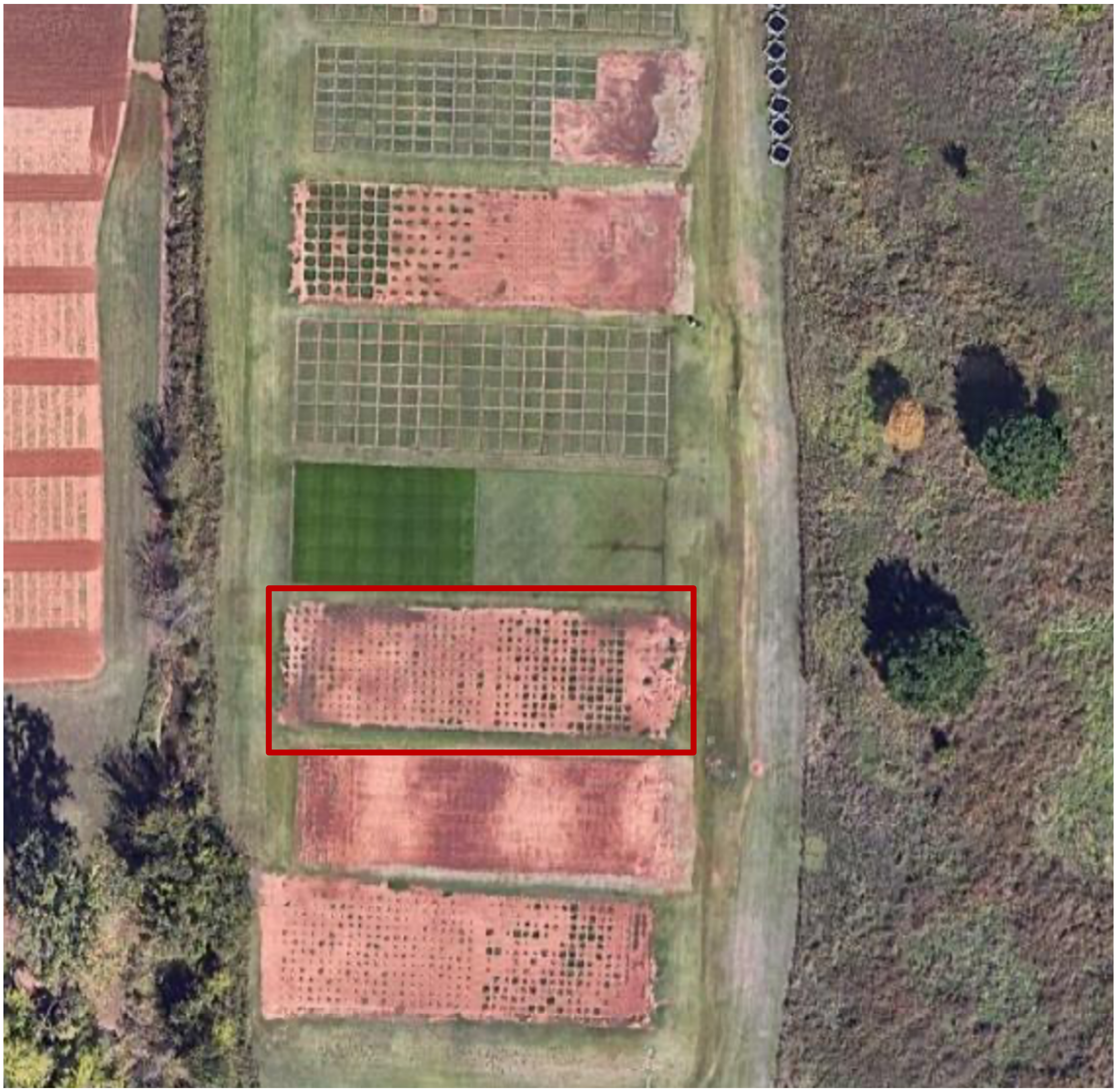


Figure A1 The Bermudagrass (*Cynodon* species) nursery located at Oklahoma State University's Agronomy Farm in Stillwater, Oklahoma. Map was retrieved from Google Maps on September 26, 2021. The Bermudagrass plots used in this study is highlighted in red.

Table A1 Details and information on the turfgrass research at the Turf Research Center (B28 field). The table also demonstrates an example on turfgrass ratings. Information is provided by turfgrass expert and evaluator.[1]

		1	2	3	4	5	6	7	8	9			
Row	**28**	3	3	4	5	5	5	4	4	5	REP 3	**ZOYSIAGRASS**	181 feet (54.3 m) of total length, 68.4" (1.74 m) per plot
	27	3	3	4	5	5	5	5	4	5			
	26	4	4	5	4	6	6	6	4	5			
	25	4	3	4	5	5	4	6	5	4			
	24	4	4	5	4	5	5	4	4	5			
	23	4	5	5	4	4	5	5	5	6	REP 2		
	22	4	5	4	5	5	5	5	5	4			
	21	4	5	5	5	5	4	5	5	4			
	20	5	5	5	3	4	4	3	2	4			
	19	5	5	4	3	5	4	4	4	4			
	18	4	3	5	2	3	4	6	5	4	REP 1		
	17	5	3	4	5	5	5	5	5	4			
	16	5	5	5	5	5	4	5	5	5			
	15	5	5	4	5	5	4	5	5	5			
	14	4	4	5	2	4	5	4	4	6			
	13	5	5	5	7	5	2	4	5	5	REP 3	**BERMUDAGRASS**	
	12	1	7	4	3	6	5	6	5	3			
	11	5	6	5	5	7	5	5	6	3			
	10	5	5	6	5	5	5	5	5	5			
	9	4	3	4	5	3	4	2	5	5	REP 2		
	8	5	6	5	6	5	5	5	5	5			
	7	5	5	3	6	5	5	7	2	5			
	6	6	5	3	5	6	5	5	5	5			
	5	2	5	5	4	5	2	5	5	4			
	4	5	6	7	5	5	4	5	4	2	REP 1		
	3	2	1	6	6	5	5	5	5	5			
	2	4	3	6	5	5	5	6	5	4			
	1	5	4	4	4	5	5	5	5	1			
W S N E		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	W S N E		
		Column											
		46 feet (14.0 m) of total width, 51.8" (1.31m) per plot											

[1] Base fertilizer was applied and incorporated into soil at a rate of 400 lbs per acre before transplanting. Bermudagrass and zoysiagrass were planted on June 24, 2020, and July 15, 2020, respectively. After full establishment, nine-inch clean alleys were maintained by periodic applications of Roundup herbicide to preserve the fidelity of adjacent plots. No N application is given when grass is under drought stress.

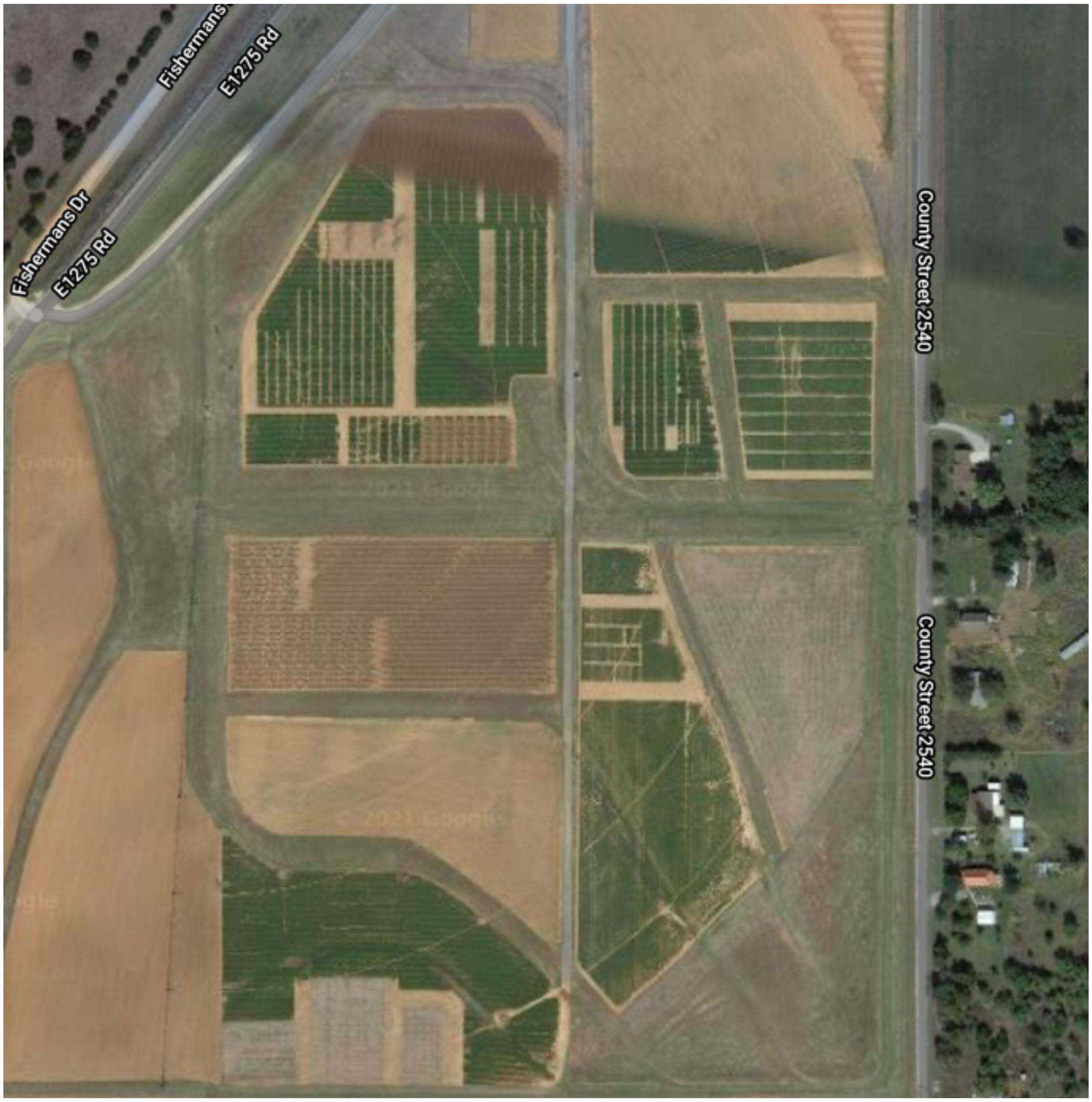


Figure A2 Fort Cobb peanut field, Caddo County, Oklahoma. Map was retrieved from Google Maps on September 26, 2021. The peanut plant plots used in this study is highlighted in red.

VITA

Peyman Nematzadeh

Candidate for the Degree of

Doctor of Philosophy

Dissertation: DEVELOPMENT OF IN-FIELD DATA ACQUISITION SYSTEMS AND MACHINE LEARNING-BASED DATA PROCESSING AND ANALYSIS APPROACHES FOR TURFGRASS QUALITY RATING AND PEANUT FLOWER DETECTION

Major Field: Biosystems Engineering

Biographical:

Education:
Completed the requirements for the Doctor of Philosophy in Biosystems Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2022.

Completed the requirements for the Master of Science in Mechanical Engineering of Biosystems at University of Tehran, Tehran, Iran in 2015.

Completed the requirements for the Bachelor of Science in Agricultural Engineering at University of Tabriz, Tabriz, Iran in 2012.

Experience:
I was a moderator at the ASABE 2021 Annual International Meeting for the session Robotics and Co-Robotics for Agriculture and Natural Resources on July 14, 2021.
Professional Memberships:
President of the Biosystems and Agricultural Engineering Graduate Student Association (BAEGSA) at Oklahoma State University in the academic year of 2020-2021.
Member of the American Society of Agricultural and Biological Engineers (ASABE) since 2020.

Being awarded as Outstanding Graduate Student for Biosystems and Agricultural Engineering at Oklahoma State University in April, 2022.