DETECTION OF SOYBEAN PESTS AND DISEASES THROUGH MACHINE LEARNING TECHNIQUES
DETECTION OF SOYBEAN PESTS AND DISEASES THROUGH MACHINE LEARNING TECHNIQUES

Master dissertation presented to the Federal University of Lavras, in fulfilment of the requirements for the degree of Master of Science in the Graduate Program of Computer Science.

Prof. DSc. Renata Lopes Rosa
Orientadora

LAVRAS – MG
2023
Omole, Oluwatoyin Joy.
56 p.

Orientadora: Prof. DSc. Renata Lopes Rosa.

Bibliografia.

I. Lopes Rosa, Renata. II. Titulo.
DETECTION OF SOYBEAN PESTS AND DISEASES THROUGH MACHINE LEARNING TECHNIQUES

DETECÇÃO DE PRAGAS E DOENÇAS DA SOJA ATRAVÉS DE TÉCNICAS DE APRENDIZADO DE MÁQUINA

Master dissertation presented to the Federal University of Lavras, in fulfilment of the requirements for the degree of Master of Science in the Graduate Program of Computer Science.

APPROVED on July 3rd, 2023.

Prof. Dante Coaquira Begazo USP
Prof. Vinicius Vitor dos Santos Dias UFLA

Prof. DSc. Renata Lopes Rosa
Orientadora

LAVRAS – MG
2023
I would like to dedicate this project to my beloved brother, whose unwavering support and encouragement were instrumental in every step of my journey. His belief in my abilities inspired me to strive for excellence beyond my limits. Furthermore, I dedicate this work to advancing sustainable agriculture, particularly in the field of soybean cultivation. With a growing global population and increasing environmental challenges, sustainable agricultural practices are crucial for the future of our planet. Through this work, I aim to contribute to developing and promoting sustainable techniques that will ensure the long-term viability and productivity of soybean farming.
THANKS

I would like to express my deepest gratitude to God, who has been my constant source of strength throughout this journey, who has uplifted me during challenging times and filled my heart with hope and resilience.

I would like to extend my heartfelt appreciation to my advisor Renata, whose wisdom, expertise, and guidance have been invaluable. Your mentorship has shaped my thinking and helped me to stay on the path. I am truly grateful for your dedication and belief in my abilities.

To my parents, I am forever indebted to your unconditional love, unwavering encouragement, and relentless sacrifices. Your constant support and belief in my dreams have been the cornerstone of my success. I cannot thank you enough for always being there for me, providing a nurturing environment, and instilling in me the values of perseverance and hard work.

To my sibling, you have been my pillar of strength, my confidant, and my source of inspiration. Your faith in me and your endless encouragement has motivated me to pursue my goals with determination. Thank you for always being there, cheering me on, and reminding me of my potential.

I would also like to express my appreciation to my friends, whose unwavering support and encouragement have made this journey more joyful and fulfilling. Your friendship has brought light to my life, and I am grateful for the memories we have created together. Thank you for standing by me, believing in me, and being a source of laughter and support.

Finally, I would like to acknowledge the support of the CAPES Scholarship program. The financial assistance and recognition provided by this esteemed organization have enabled me to pursue my academic and research goals. I am honoured to have been selected as part of the beneficiaries of this scholarship, and I am grateful for the opportunities it has afforded me.

To all those mentioned above, your contributions have played an integral role in shaping my journey and making this achievement possible. Thank you for your unwavering belief in me, your support, and your love. I am forever grateful for every one of you.
RESUMO

A soja, fonte vital de proteína e óleo vegetal, desempenha um papel significativo no crescimento econômico dos países produtores. No entanto, doenças e infestação de pragas representam uma ameaça substancial para a produtividade da soja. A detecção precoce desses problemas nas folhas de soja é crucial para prevenir ou reduzir perdas de produção. Os métodos de aprendizado de máquina e aprendizado profundo mostraram-se promissores na detecção de doenças da soja. Neste estudo, investigamos modelos comumente usados para classificação de imagens de plantas, com foco em doenças da soja e identificação de pragas. Seis modelos foram selecionados, incluindo três modelos simples de aprendizado de máquina e três modelos de aprendizado profundo. Para avaliar seu desempenho, empregamos validação cruzada de 10 vezes e avaliamos a precisão da classificação, precisão, recuperação e métricas de medida F. Nossos resultados superaram os de estudos anteriores, alcançando maior precisão na detecção de doenças e pragas da soja. Notavelmente, o conjunto de dados de doenças superou o conjunto de dados de pragas, apesar do último ser maior. Entre os algoritmos testados, o VGG-16, um algoritmo de aprendizado profundo, demonstrou desempenho superior. A seguir estão as precisões de classificação alcançadas para os conjuntos de dados de pragas e doenças, respectivamente, usando diferentes algoritmos: Support Vector Machine (88% e 92%), Random Forest (83% e 95%), K Nearest Neighbors (76% e 74%), VGG-16 (95% e 99%), VGG-19 (94% e 98%), e nosso algoritmo CNN personalizado, ViewNet (89,5% e 75%). Ao alavancar a validação cruzada de 10 vezes, uma técnica amplamente utilizada em aprendizado de máquina, garantismos avaliações confiáveis e robustas dos modelos. Essas descobertas contribuem para o avanço das práticas agrícolas, fornecendo informações sobre aprendizado de máquina eficaz e abordagens de aprendizado profundo para doenças da soja e detecção de pragas.

ABSTRACT

Soybean, a vital source of protein and vegetable oil, plays a significant role in the economic growth of producing countries. However, diseases and pest infestation pose a substantial threat to soybean yield. Early detection of these issues on soybean leaves is crucial for preventing or reducing production losses. Machine learning and deep learning methods have shown promise in detecting soybean diseases. In this study, we investigated commonly used models for plant image classification, focusing on soybean disease and pest identification. Six models were selected, including three simple machine-learning models and three deep-learning models. To evaluate their performance, we used classification accuracy, precision, recall, and F-measure metrics. Our results surpassed those of previous studies, achieving improved accuracy in detecting soybean diseases and pests. Notably, the disease data set outperformed the pest data set, despite the latter being larger. Among the algorithms tested, VGG-16, a deep learning algorithm, demonstrated superior performance. The following are the classification accuracies achieved for the pest and disease data sets, respectively, using different algorithms: Support Vector Machine (88% and 92%), Random Forest (83% and 95%), K Nearest Neighbors (76% and 74%), VGG-16 (95% and 99%), VGG-19 (94% and 98%), and our custom CNN algorithm, ViewNet (89.5% and 75%). By leveraging 10-fold cross-validation, a widely used technique in machine learning, we ensured reliable and robust evaluations of the models. These findings contribute to the advancement of agricultural practices by providing insights into effective machine learning and deep learning approaches for soybean disease and pest detection.

### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Soybean Leaf with Asian Soybean Rust Disease</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Soybean Leaf with Frog-eye Leaf Spot Disease</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Soybean Leaves Damaged by Caterpillars</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>Actions of Diabrotica Speciosa on Soybean Leaf</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>Images of Soybean Caterpillar and Diabrotica Speciosa</td>
<td>21</td>
</tr>
<tr>
<td>2.6</td>
<td>How SVM Classifies Data</td>
<td>23</td>
</tr>
<tr>
<td>2.7</td>
<td>Decision Tree Classifier</td>
<td>24</td>
</tr>
<tr>
<td>2.8</td>
<td>Neural Networks</td>
<td>26</td>
</tr>
<tr>
<td>2.9</td>
<td>Neural Networks</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Sample Images from the Pest Data Set</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Sample Images from the Disease Data Set</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Number of Images per Fold (Pest Data)</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Deep Learning Architecture</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Deployment Home Page</td>
<td>40</td>
</tr>
<tr>
<td>3.6</td>
<td>Prediction/Result Page</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Training accuracies for the deep learning models</td>
<td>44</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 3.1 – Number of images for Pest Data set ........................................ 32
Table 4.1 – Result Comparison for Deep Learning Algorithms .................... 43
Table 4.2 – Result Comparison for Simple Machine Learning Algorithms .......... 45
Table 4.3 – Result Comparison with Previous Works .................................... 46
LIST OF FRAMES

Frame 2.1 – Comparison of Machine Learning Algorithms . . . . . . . . . . . . . . . . . . . . . . 27
## CONTENTS

1 INTRODUCTION .............................................................................. 12
  1.1 Objectives ................................................................................. 13
  1.2 Justification ............................................................................... 14
2 THEORETICAL REVIEWS AND RELATED WORKS ......................... 16
  2.1 The Soybean Plant .................................................................. 16
     2.1.1 Soybean Diseases ................................................................. 16
     2.1.1.1 Asian Soybean Rust (ASR) ............................................... 16
     2.1.1.2 Soybean Frog-eye Leaf-spot ......................................... 18
     2.1.2 Soybean Pests [Caterpillars] .............................................. 18
     2.1.3 Soybean Pests (Diabritica Speciosa) .................................. 20
  2.2 Machine Learning and Deep Learning Algorithms for Pest Diseases Detection ........................................... 22
     2.2.1 Support Vector Machine (SVM) .......................................... 22
     2.2.2 Naive Bayes (NB) ................................................................. 22
     2.2.3 Decision Trees (DT) .............................................................. 23
     2.2.4 Random Forest ................................................................. 24
     2.2.5 Artificial Neural Network (ANN) ...................................... 25
     2.2.6 K- Nearest Neighbor (KNN) .............................................. 25
     2.2.7 Convolutional Neural Network (CNN) ................................. 27
        2.2.7.1 Transfer Learning ......................................................... 28
  2.3 Related Works on Soybean Pest and Disease ............................... 29
3 METHODOLOGY ........................................................................... 32
  3.1 Data-set ...................................................................................... 32
     3.1.1 Pest Data ........................................................................... 32
     3.1.2 Disease Data-set ................................................................. 33
  3.2 Data Preprocessing .................................................................... 34
  3.3 Model Training, Validation and Testing ....................................... 34
     3.3.1 Deep Learning Model ......................................................... 35
     3.3.2 Machine Learning ............................................................... 37
     3.3.3 Tools and Libraries Used ................................................... 37
  3.4 Model Evaluation ....................................................................... 38
  3.5 Model Deployment with Flask ..................................................... 39
4 RESULTS AND DISCUSSION ................................................................. 42
1 INTRODUCTION

Plants play important roles in life sustenance, aside from the fact that the food we consume comes from them, they also help in providing oxygen, shelter and even clothing. According to The Institute of Food and Agricultural Sciences (2019), the only source of food and oxygen are plants, no animal alone can supply these. Therefore, plants can be considered a source of every basic need of humans.

The soybean plant is popular for its high protein source and wide variety of use, it is in fact known as the plant that produces the most protein per cultivated area (LIU, 2012). The soybean plant also accounts for 48% of the world oil crop market (XIAOMING; QIONG, 2018). It serves as food for both humans and animals, it can be processed into different food products, its also rich in nutritional contents that are good for the heart and for diabetic patients. Recently, the average annual yield loss of the soybean crop accounts for between 8% and 25% in the United States only due to diseases (BEVERS; SIKORA; HARDY, 2022) and in Brazil, which is the second highest producer of soybean, Soybean rust reduced soybean up to 13-33% (FATTORI; SENTELHAS; MARIN, 2022). Other factors like pests, environmental conditions, cultivar selection, and management practices also affect the yield and quality of soybean and increase their susceptibility to diseases (ALLEN et al., 2017).

Pests and diseases have continually been a menace to optimum productivity in agriculture, according to Meshram et al. (2021) detection of disease is an important task that saves crops from major losses. when it is done quickly and accurately, such losses are reduced. Deep Learning (DL) and Machine Learning (ML) models have proven effective as a means of plant disease detection. Machine Learning involves applying algorithms that learn from problem-specific training data, this allows machines to find hidden insights and complex patterns without explicitly being programmed (JANIESCH; ZSCHECH; HEINRICH, 2021). Machine Learning (ML) models have been effective as a means of plant disease detection, either with the supervised learning approach or with the unsupervised learning approach.

Supervised learning in disease detection is when the machine is taught and trained using a well-labelled image data set of diseased pair. That is the data already tagged with the correct disease classifications or the absence. The larger the data set, the more accurately the machine learns from it (BARBEDO, 2016). In the supervised setting, the acquired expertise (trained model) is used to predict the outputs (labels) for the test data. In unsupervised learning, however, there is no distinction between training and test sets with data being unlabeled. The learning algorithm processes input data with the goal of discovering hidden patterns (LIAKOS et al., 2018).

There are various learning algorithms with which the ML process can be achieved. Some of the ML models that have been successfully adopted for detecting plant disease include Support Vector Machine (SVM), Artificial Neural Networks (ANN), Decision Trees, K-means and Random Forest (GUI;
A type of ML based on Artificial Neural Networks (ANN) where a large number of processing layers are organized in deeply nested network architectures is called Deep Learning (DL) (JANIESCH; ZSCHECH; HEINRICH, 2021). The DL method that was utilized in this study is the Convolutional Neural Network (CNN) as it is considered one of the most powerful techniques in modelling complex processes and performing pattern recognition in applications with a large amount of data (FERENTINOS, 2018).

These models are as well effective in the classification or detection of plant pests. Kasinathan, Singaraju e Uyyala (2021) used SVM, ANN, K-Nearest Neighbor (KNN), Naive Bayes (NB), and CNN to classify and detect the insects in corn, soybean, and wheat. However, many machine learning methods are yet to be robust enough to fully replace traditional methods of plant disease detection, though these traditional methods of identifying plant diseases are slow, laborious and can also be inaccurate (KABIR; OHI; MRIDHA, 2021), but often, due to the choice of process or its specificity to one symptom, they are sometimes required to back or confirm algorithm results (BARBEDO, 2016; BARBEDO, 2018). Deep learning on the other hand has been confirmed to outperform humans in many image recognition tasks (NANNI et al., 2021). One of the most difficult challenges is that plant species have some diseases with a significant degree of similarity and can appear simultaneously on a single plant. This affects not only machine learning methods but equally human experts as well. Thus, it is important to pick machine learning methods based on the target diseases (ALIYU; MOKJI; SHEIKH, 2020).

Different studies have tried to classify plant diseases using different deep learning and machine learning algorithms, and this has also yielded different classification accuracies, this study, however, included the action of plant pests in the classification categories, the performance of the chosen deep learning and machine learning algorithms were evaluated based on the accuracy scores, precision, recall and F1 measure obtained in multi-class classification of the diseases, pests and healthy plants. VGG16 and VGG19 gave significantly higher results than in the compared previous works.

1.1 Objectives

The main objective of this study is to compare the performance of CNN algorithms with the other 3 machine learning algorithms and to highlight the importance of early Disease and Pest action on leaf detection.

The following specific objectives will help in achieving the main goal of this study:

a) To identify major diseases and pests affecting the soybean plant;

b) To identify the most frequently used ML and DL algorithms for detecting crop pests and diseases;
c) Select relevant data sets to work with based on the list of the major pests and diseases affecting the soybean plant;
d) To train the selected models with collected data and predict soybean diseases using deep learning and different machine learning algorithms;
e) To deploy the most efficient model based on classification accuracy, precision, recall, and F-measure scores with Flask for web testing.

1.2 Justification

Based on the importance of the soybean plant as a major source of protein for both humans and livestock, it is important to quickly and efficiently detect disease and pest infestation. Early detection improves control and this in turn improves production and availability of the crop. Several works have been done in detecting the presence of different Soybean diseases but compared to these works, there are fewer studies on Soybean pest detection or classification. However, both plant phenomenons are equally important for sustainable and effective plant management. A study carried out by Hampf et al. (2021a) showed that a minimum of 11.67% yield loss are caused by different soybean pests.

In general, early diagnosis of both plant and disease with ML techniques is of high significance in research because of its benefits which include:

a) Efficient resource utilization: Early detection allows for targeted and precise interventions, such as applying pesticides or implementing disease management strategies, only where and when necessary. This reduces the overall use of chemical inputs, minimizing environmental pollution and the negative impacts on beneficial organisms, including pollinators and natural predators of pests;
b) Reduced pesticide usage: ML-based early diagnosis enables farmers to apply pesticides judiciously and in a more targeted manner. By accurately identifying the presence of pests or diseases at an early stage, farmers can employ integrated pest management practices, including biological controls and cultural practices, as alternatives to or in combination with chemical treatments. This reduces the reliance on pesticides, promoting sustainable agricultural practices and minimizing the harmful effects on ecosystems and human health;
c) Enhanced crop productivity: Early diagnosis helps prevent or minimize yield losses caused by pests and diseases. By detecting issues at their earliest stages, farmers can take proactive measures to manage or mitigate the problems effectively. This leads to healthier plants, improved crop growth, and increased overall productivity. As a result, more food can be produced from the same amount of agricultural land, contributing to addressing
global food security and reducing pressure to expand agricultural areas into natural habitats;

d) Sustainable agricultural practices: ML-based early diagnosis supports the adoption of sustainable agricultural practices, such as precision agriculture and site-specific management. By optimizing resource allocation, minimizing chemical inputs, and maximizing crop health, these practices promote environmental sustainability, conserve natural resources and reduce the ecological footprint of agriculture.

This study has achieved higher classification accuracies than compared to previous works which shows an improvement in the goals of sustainable agriculture through the use of technology. Many future researchers can also leverage the findings of this research in creating better platforms for soybean pest and disease detection.

Following the introduction, this project document is organized into several sections starting with the theoretical reviews and related works, followed by the methodology, then the results and discussion and finally the conclusion.
2 THEORETICAL REVIEWS AND RELATED WORKS

This section provides essential information about the soybean plant, exploration of the selected diseases and pests that commonly affect soybean plants, and highlights their impact on crop yield.

In the context of plant disease and pest detection, we discuss different machine-learning algorithms that are commonly used, and finally related works in the field of pest and disease detection.

2.1 The Soybean Plant

Soybean is a legume crop grown worldwide for its protein and oil content (MEDIC; ATKINSON; HURBURGH, 2014). Though the crop is known to have its origin in China, the US and Brazil supply over 80% of the global exports (GALE; VALDES; ASH, 2019). The soybean plant is considered one of the most important crops worldwide because the seeds are important for both protein meal and vegetable oil (HARTMAN; WEST; HERMAN, 2011). Soybean yields have been suppressed by both pests and diseases. Soybean pests cause yield losses of up to 11% and these losses usually come in the form of a reduction in yield and grain quality (HAMPF et al., 2021b). Similarly, the yield loss due to soybean diseases can also result in up to 11% yield loss in the top producing countries, and when these diseases are considered individually, a mean loss of above 51% was recorded (BANDARA et al., 2020; WRATHER; KOENNING, 2009; RUPE; LUTTRELL, 2008). Thus, to reduce the losses caused by pests and diseases, it is important to detect their infection or presence early as this can lead to a quick response in mitigating the loss and this detection has been carried out successfully with both deep learning and machine learning techniques (SINGH et al., 2020) and (LIU; WANG, 2021).

2.1.1 Soybean Diseases

Soybean diseases can be classified into 3 namely root diseases, stem diseases and leaf diseases (MARKELL; MALVICK, 2018). The diseases affect different parts of the plant as the name is attached. Of the many diseases that affect the soybean plant, this study will be focusing on the soybean Leaf Spot disease and the Asian Rust diseases because they are part of the important economically significant soybean diseases and there are more public data available for them. (GUI; MBAYE, 2019; JADHAV, 2019; HAMPF et al., 2021b).

2.1.1.1 Asian Soybean Rust (ASR)

Asian Soybean rust is a fungus disease and has been rated one of the most deadly diseases of the soybean plant. In Brazil, Asian soybean rust causes yield loss of up to 90% (GODOY et al., 2016). The
fungus is able to survive all year round on the host plant and has been known to affect other plants apart from the soybean plant (PRIMIANO et al., 2017). Generally, the mature leaves are usually affected by rust because the younger plants are less susceptible to rust diseases due to stomata development, but in the case of soybean Asian rust, the fungus does not use the stomata for penetration which leaves good chances for the younger plants to be affected also (FURTADO et al., 2009; PRIMIANO et al., 2017). The major action of the disease in causing yield loss is that it increases the reduction in the leaf area along with an early maturation, which in turn leads to losses in the grains’ weight and quality (GABRIEL et al., 2018; FATTORI; SENTELHAS; MARIN, 2022). The symptoms of Asian Soybean Rust are leaf lesions containing several individual uredia and early defoliation of soybean plants due to the destruction of the healthy leaf area by rust severity. The symptoms first appear as a yellow tiny spot on the upper side of the leaves as shown in figure 2.1 (REIS; ZANATTA; REIS, 2019; PRIMIANO et al., 2017).

Figure 2.1 – Soybean Leaf with Asian Soybean Rust Disease

Source: Sikora (2021)
2.1.1.2 Soybean Frog-eye Leaf-spot

The soybean Frog-eye Leaf Spot (FLS) disease according to Soybean Research and Information Network is the most foliar disease of soybean. Frog-eye leaf spot of soybean is caused by the pathogen Cercospora sojina, its been found worldwide with the potential to cause yield loss of up to 60% (PHIL-LIPS; KANDEL; MUELLER, 2021). Young soybean leaves are most susceptible, while older ones are more resistant. Leaf symptoms are first noticeable after plants begin to bloom. Infection occurs during warm, humid weather with cloudy days and frequent rain. When these conditions persist, the infection can spread from leaves to pods, seeds, and stems (BRADLEY et al., 2021). The symptoms of soybean FLS diseases are:

a) FLS appears as circular, tan to grey spots on the leaves, surrounded by very pronounced dark purple margins. They are most often the size of the end of a pencil eraser, about ¼ inch in diameter. Smaller lesions can coalesce into larger lesions as seen in figure 2.2 below;

b) When disease pressure is high on susceptible varieties, spots can also be found on the stems, pods, and seeds. Lesions on pods are circular to elongate and slightly sunken, with a reddish-brown colour. As pod lesions age, they become brown to light grey with narrow dark-brown borders. The pathogen can penetrate through the pod wall and infect the developing seeds;

c) Symptoms on seeds appear as conspicuous light to dark grey or brown areas that can range from specks to large blotches covering the entire seed coat, and the seed coat may crack or flake.

When any of these symptoms of FLS is noticed, or even during the seasons when this disease is more prevalent, it is important to test or scout the field for FLS disease so it can be quickly controlled in case of its before it spreads so much in the field. It is, therefore, necessary to deploy machine learning algorithms in the early and accurate detection of this disease.

2.1.2 Soybean Pests [Caterpillars]

According to Grande e Rando (2018) farmers reported that bedbugs and caterpillars are the main pests affecting their crops. Soybeans can be attacked by pests at any stage from seedlings to harvest, they are more tolerant of insect damage than most other grain legumes and noticeable damage (particularly leaf damage) does not necessarily result in yield loss. During the vegetative stage, 33% leaf defoliation can be tolerated without yield loss, although this falls to 16% during pod set (GRDC, Grains Research and Development Corporation, 2016).
Caterpillars are pests that attack leaves, stems, pods, and grains, depending on their gender. In some cases, they can attack more than one item (HODGSON et al., 2021). There are different kinds of soybean caterpillars with unique actions on the plants. An example is the velvet bean caterpillar that has soybean as its primary host, they are the fastest defoliating pest of the soybean plant, and they are able to strip the field in 5 to 7 days (HODGSON et al., 2021). Another dangerous soybean caterpillar is the Soybean Looper. They cause defoliation exceeding 20% from pod initiation to pod fill and can result in significant yield loss (CHEN et al., 2018). Caterpillars cause defoliation in the leaves as shown in figure 2.3.

The data set for this study contains images of the actions of different caterpillars on the soybean plant. It contains more images of the actions of Spodoptera which cause damage to seedlings, reducing the growing stand, but they have also been reported as defoliators in the reproductive stage and feed on pods (MIGNONI et al., 2022). We will be detecting the actions of these caterpillars on the plant before it gets too late to recover.
2.1.3 Soybean Pests (Diabritica Speciosa)

This pest is considered one of the most important agricultural pests in Latin America. It is one of the three neotropical species of the Diabritica spp. that is considered an agricultural pest in South America (WALSH et al., 2020; COSTA et al., 2018). The adult and larva of this pest have different food preferences, in their adult stage, they are defoliators and can cause damage to the plants’ pods and flowers (MIGNONI et al., 2022; ÁVILA; BITENCOURT; SILVA, 2019). They are regarded as pests of seedlings and young plants and they attack a wide variety of crops of which soybean is an important one (WALSH et al., 2020). Ávila, Bitencourt e Silva (2019) confirmed in their study that the adults had better development and preference for bean plants, they feed on the softer leaves making small round holes on them, they also make incisions on the edges of the leaves (MIGNONI et al., 2022; WALSH et al., 2020). The action of this pest on the soybean is shown in Fig. 2.4 below.

The adult Diabrotica Speciosa measures about 8 to 10 millimetres in length. The beetle’s colouration can vary, but it commonly exhibits a bright yellow or greenish-yellow hue as shown in figure 2.5. On the other hand, The soybean caterpillar has a distinct physical appearance that changes as it progresses through different life stages. In its early larval stage, it typically has a green colouration with a cylindrical body covered in fine hairs as shown in figure 2.5. As it matures, the caterpillar develops a brownish hue and grows up to 3 centimetres in length.
Figure 2.4 – Actions of Diabrotica Speciosa on Soybean Leaf

Figure 2.5 – Images of Soybean Caterpillar and Diabrotica Speciosa

Caterpillar
Diabrotica Speciosa

Source: Mignoni et al. (2022)
2.2 Machine Learning and Deep Learning Algorithms for Pest Diseases Detection

Machine learning algorithms are mathematical model mapping methods used to learn or uncover underlying patterns embedded in the data (ABRAHAM; JAYANTHI; BHASKaran, 2020). These algorithms have been applied to improve quality and precision in different fields of study of which agriculture is an important part. Machine learning algorithms work on different types of data such as numerical data, categorical data, texts and time series. Image data sets will be used for this study, therefore only machine learning algorithms common for image classification will be discussed in this section.

2.2.1 Support Vector Machine (SVM)

SVM is a supervised learning algorithm that can be used for both classification and regression problems. SVM algorithm can classify both linear and non-linear data. SVM first maps each data item into an n-dimensional feature space where n is the number of features. Then identifies the hyperplane that separates the data items into two classes while maximizing the marginal distance for both classes and minimizing the classification errors (SCHOLKOPF, 1999). The marginal distance for a class is the distance between the decision hyperplane and its nearest instance which is a member of that class. More formally, each data point is plotted first as a point in an n-dimension space (where n is the number of features) with the value of each feature being the value of a specific coordinate. To perform the classification, we then need to find the hyperplane that differentiates the two classes by the maximum margin (UDDIN et al., 2019). The hyperplane classification sometimes might not come as easy and direct, which is why it might need some tuning in parameters. SVM might not perform optimally in a case of a large data set because it will require a higher training time, and also when the data set has its target classes overlapping.

Similar to neural networks, SVM possesses the ability to be a universal approximator of any multivariate function to any desired degree of accuracy (WANG, 2005). SVM implements the strategy of keeping the value of the training error or approximation error fixed and minimizing the confidence interval.

2.2.2 Naive Bayes (NB)

This is also a supervised machine learning algorithm. It uses the Bayes theorem of probability to predict the class of unknown data sets. This classifier simplifies learning by assuming that features are independent given class (RISH et al., 2001). It assumes that a particular feature in a class is not
directly related to any other feature although features for that class could have interdependence among themselves (UDDIN et al., 2019).

\[
P(A|B) = \frac{P(B|A)P(A)}{P(B)}
\]  

(2.1)

Where:
- A, B = events
- P(A|B) = probability of A given B is true
- P(B|A) = probability of B given A is true
- P(A), P(B) = the independent probabilities of A and B

The Naive Bayes algorithm is fast and does not require a large training data set. Naive Bayes can be used in real-time prediction, multi-class prediction, text classification and recommendation systems. A major limitation of the algorithm is the assumption of independent predictors.

### 2.2.3 Decision Trees (DT)

This is a classification and regression algorithm with a form of flowchart structure. The observations are the branches and the decisions are the leaves. A decision tree models the decision logic i.e., tests and corresponds outcomes for classifying data items into a tree-like structure. The nodes of a decision tree normally have multiple levels where the first or top-most node is called the root node (UDDIN et al., 2019). All internal nodes (i.e., nodes having at least one child) represent tests on input variables or attributes. Depending on the test outcome, the classification algorithm branches towards the appropriate child node where the process of test and branching repeats until it reaches the leaf node (UDDIN et al.,
There are different types of decision tree algorithms. Some examples are CHAID (Chi-squared Automatic Interaction Detector), CART (Classification and Regression Tree), ID3 (Iterative Dichotomiser 3), C4.5, Hunts Algorithm, etc. It has applications in business, intrusion detection, energy modelling, e-commerce, image processing and in medicine. (BRIJAIN et al., 2014).

**Figure 2.7 – Decision Tree Classifier**

```
\hat{Y} = \text{RF}(X) = \frac{1}{N} \sum_{i=1}^{N} T_i(X)
```

Where:

- \( \hat{Y} \) represents the predicted output
- X represents the input features
- RF represents the random forest model
- $T_i$ represents the individual decision tree in the forest.

The equation calculates the average prediction of all the decision trees in the random forest ensemble to obtain the final prediction, where $(N)$ is the total number of decision trees in the random forest.

### 2.2.5 Artificial Neural Network (ANN)

Artificial Neural networks are supervised learning algorithms. They are inspired by the functioning of the neural networks of the human brain. They were first proposed by McCulloch e Pitts (1943). An ANN algorithm can be represented as an interconnected group of nodes. The output of one node goes as input to another node for subsequent processing according to the interconnection. Nodes are normally grouped into a matrix called a layer depending on the transformation they perform. Apart from the input and output layer, there can be one or more hidden layers in an ANN framework (UDDIN et al., 2019). ANN typically start out with randomized weights for all their neurons. This means that initially they must be trained to solve the particular problem for which they are proposed (NIKAM, 2015). Learning of neural network is performed by adjusting the weight of the connection, by updating the weight iteratively performance of the network is improved. On the basis of connection, ANN can be classified into two categories: feed-forward network and recurrent network (BALA; KUMAR, 2017).

ANN has many advantages such as its tolerance to noisy data, its comprehensive, and learning well from examples but it also has some disadvantages like a long training time, high computational cost, and adjustment of weight.

### 2.2.6 K-Nearest Neighbor (KNN)

K-Nearest Neighbors (KNN) algorithm is a simple yet effective machine learning algorithm used for both classification and regression tasks. It operates on the principle of similarity, where the prediction for a new data point is based on its proximity to its K nearest neighbours in the training data set. The KNN algorithm is relatively straightforward and easy to understand. However, it can be computationally expensive for large data sets since it requires calculating distances between the new data point and all training data points. Nevertheless, KNN is a versatile algorithm that can be applied to various domains and serves as a good baseline for comparing with other more complex machine learning methods.

KNN equation for Classification
Figure 2.8 – Neural Networks

\[ \hat{Y} = \underset{c}{\arg\max} \left( \sum_{i=1}^{K} I(y_i = c) \right) \]  

(2.3)

Where:
- \( \hat{Y} \) represents the predicted output,
- \( K \) is the number of nearest neighbours to consider,
- \( y_i \) denotes the target value of the \( i \)-th nearest neighbor, and
- \( I(\cdot) \) is the indicator function that evaluates to 1 if the condition inside the parentheses is true and 0 otherwise.

KNN equation for Regression
\[ \hat{Y} = \frac{1}{K} \sum_{i=1}^{K} y_i \]  

(2.4)

Where:
- \( \hat{Y} \) represents the predicted output,
- \( K \) is the number of nearest neighbours to consider, and
- \( y_i \) denotes the target value of the \( i \)-th nearest neighbor.

All the mentioned algorithms have their different advantages and disadvantages and each one is more applicable in different situations. Taunk et al. (2019) gave a number of comparisons among these classification algorithms as in frame 2.1 below.
Frame 2.1 – Comparison of Machine Learning Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>- Effective in high-dimensional spaces</td>
<td>- Less efficient for large data sets</td>
</tr>
<tr>
<td></td>
<td>- Suitable for small to medium-sized data sets</td>
<td>- Kernel selection can impact performance</td>
</tr>
<tr>
<td></td>
<td>- Handles non-linear decision boundaries</td>
<td>- Can be sensitive to noise and outliers</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>- Fast and efficient</td>
<td>- Assumes independence between features</td>
</tr>
<tr>
<td></td>
<td>- Works well with high-dimensional data</td>
<td>- May not capture complex relationships</td>
</tr>
<tr>
<td></td>
<td>- Handles categorical features well</td>
<td>- Sensitive to correlated features</td>
</tr>
<tr>
<td>Decision Trees</td>
<td>- Easy to interpret and visualize</td>
<td>- Prone to overfitting without proper pruning</td>
</tr>
<tr>
<td></td>
<td>- Handles feature interactions automatically</td>
<td>- Can be sensitive to small variations in data</td>
</tr>
<tr>
<td></td>
<td>- Can handle both numerical and categorical data</td>
<td>- May create biased trees with imbalanced data</td>
</tr>
<tr>
<td>Random Forest</td>
<td>- Reduces overfitting with an ensemble of trees</td>
<td>- Less interpretable compared to single decision trees</td>
</tr>
<tr>
<td></td>
<td>- Handles high-dimensional data well</td>
<td>- Can be computationally expensive</td>
</tr>
<tr>
<td></td>
<td>- Robust to outliers and noise</td>
<td></td>
</tr>
<tr>
<td>Neural Networks</td>
<td>- Powerful for complex patterns</td>
<td>- Requires more training data</td>
</tr>
<tr>
<td></td>
<td>- Can handle large amounts of data</td>
<td>- Longer training time for deep networks</td>
</tr>
<tr>
<td></td>
<td>- Non-linear modeling capabilities</td>
<td>- Requires careful tuning of hyperparameters</td>
</tr>
<tr>
<td>KNN</td>
<td>- Simple and intuitive</td>
<td>- Requires more computational time</td>
</tr>
<tr>
<td></td>
<td>- Effective for small data sets</td>
<td>- Performance can degrade with high-dimensional data</td>
</tr>
</tbody>
</table>

Source: Author (2023)

2.2.7 Convolutional Neural Network (CNN)

Convolutional neural networks (CNN) are designed specifically for image classification and recognition tasks (ZHANG et al., 2019; O’SHEA; NASH, 2015; PATEL, 2020). They are composed of layers of interconnected nodes, where each layer is responsible for learning that can be applied to the input data. These filters are designed to detect specific patterns or features in the input data, such as edges, corners, or specific shapes (O’SHEA; NASH, 2015; KRIZHEVSKY; SUTSKEVER; HINTON, 2017).

Convolutional Networks are inspired by the structure and function of the brain, which processes visual information. They consist of an input layer, one or more hidden layers, and an output layer. The hidden layers are composed of convolutional layers, pooling layers, and fully connected layers (LECUN et al., 1998; KRIZHEVSKY; SUTSKEVER; HINTON, 2017; PATEL, 2020). The Convolutional layers apply filters to the input data to extract features, pooling layers downsample the feature maps to reduce the spatial dimensions, and fully connected layers combine the features to make a prediction.

CNN has been successfully applied to a wide range of image recognition and classification tasks, including object recognition, face recognition, and scene classification (ALZUBAIDI et al., 2021). They
are also used in majority of the computer vision tasks which include image detection, image tagging, image recognition, image classification, image analysis, video analysis, and natural language processing. Deep Learning has gradually become the most widely used computational approach in the field of ML, thus achieving outstanding results on several complex cognitive tasks, matching or even beating those provided by human performance. CNN also has applications in various fields such as Medicine, Agriculture, Mathematics etc (BHATT et al., 2021; YAMASHITA et al., 2018; ALZUBAIDI et al., 2021). Several studies (KRIZHEVSKY; SUTSKEVER; HINTON, 2017; SIMONYAN; ZISSERMAN, 2014) have been done to improve the CNN algorithm to achieve better results in its various fields of application, we are progressing studies by achieving better results for pest and detection both with a custom algorithm and with transfer learning.

Figure 2.9 – Neural Networks

2.2.7.1 Transfer Learning

Transfer learning is a machine learning technique that involves using knowledge learned from one task to improve the performance of a different but related task. It allows a model trained on one data set to be fine-tuned or adapted for use on a different data set, without the need to start from scratch and train a new model from scratch (PAN; YANG, 2010; TAN et al., 2018) Knowledge learned from a larger, related data set can be useful in tasks that have smaller data sets through transfer learning, it solves the problem of insufficient training data and forfeits the claims that training and testing data must be taken
from the same domain (WEISS; KHOSHGOFTAAR; WANG, 2016; TAN et al., 2018). There are different kinds of transfer learning (TAN et al., 2018; WEISS; KHOSHGOFTAAR; WANG, 2016) but for this study, we used the VGG algorithms and compared them with an algorithm we developed. VGG is the short form for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. It was developed by Simonyan and Zisserman (SIMONYAN; ZISSERMAN, 2014). It has smaller filters 3X3 and greater depths which contributes to its excellent performance (ALOM et al., 2019). There are two common variations of the VGG architecture, one consisting of 13 convolutional layers which is called VGG16 and VGG16 which consist of 16 convolutional layers (LAVIN; GRAY, 2016; SIMONYAN; ZISSERMAN, 2014). The architecture of the VGG algorithm consists of the following:

a) 2 convolutional layers with 3X3 layers, both of which uses ReLU activation functions;

b) max pooling layer and several fully connected layers that uses the ReLU activation function;

c) the softmax layer for classification which is the last layer.

2.3 Related Works on Soybean Pest and Disease

This section aims to explore and compare various approaches that have been used in the field of plant pests and disease detection using different machine learning algorithms.

A study by (JADHAV; UDUP; PATIL, 2019) proposes a novel system for detecting and classifying soybean diseases, such as Blight, Frogeye leaf spot, and Brown Spot, using multiclass support vector machine (SVM) and K-nearest neighbour (KNN) classifiers. The system utilizes colour images of diseased leaf samples and achieves an accuracy of 87.3% and 83.6% respectively for disease classification. SVM also produced better results than other simple machine learning algorithms in the study by (SUJATHA; MAHALAKSHMI, 2020) where they compared the performance of machine learning (ML) and deep learning (DL) methods for plant leaf disease detection. The authors experiment with various ML (Support Vector Machine, Random Forest, Stochastic Gradient Descent) and DL (Inception-v3, VGG16, VGG16) models. However, the results indicate that DL methods, particularly VGG16, outperform ML methods, achieving a disease classification accuracy of 89.5%.

The study of (PAJJURI; KUMAR; THOTTOLIL, 2022) evaluates various state-of-the-art transfer learning architectures, including GoogLeNet, AlexNet, VGG16, and ResNet50V2, for plant disease detection and classification. The models are tested on different plant disease data sets. The results indicate that VGG16 achieves the highest accuracy of 96.6%, 98.5%, and 89% on different data sets, surpassing other models.
However, VGG16 gave a lower accuracy than ResNet50 and VGG19 in the study by (SAHU; PANDEY; GEETE, 2021) whose work focuses on the classification of soybean leaf diseases using deep learning algorithms and transfer learning. The authors compare the performance of six pre-trained deep convolutional neural networks (CNNs) with traditional machine learning methods. The results show that the fine-tuned ResNet50 network achieves a classification accuracy of 93%, outperforming other models.

The authors (GUI; MBAYE, 2019) propose a Deep Convolutional Neural Network (CNN) based on LeNet for the recognition and classification of soybean leaf spot diseases. The affected areas of disease spots are segmented from leaf images using the Unsupervised Fuzzy Clustering algorithm. The proposed model achieves a testing accuracy of 89.84%. However, it shows poor per-class recognition results with 1378 misclassified images and 1271 correctly classified images. Comparatively, the VGG16 model achieves better performance with a success rate of 93.54% and improved per-class recognition results, misclassifying 1245 images and correctly classifying 1404 images. Also, the study by (BEVERS; SIKORA; HARDY, 2022) focuses on the development of an automated classifier for soybean disease identification using convolutional neural networks (CNNs). The authors acquire more than 9,500 original soybean images representing eight distinct disease and deficiency classes. They experiment with various approaches to transfer learning, data engineering, and data augmentation to enhance model training. The best-performing model is based on the DenseNet201 architecture, achieving an overall testing accuracy of 96.8%. Another model, VGG16, achieves an accuracy of 90.8%. The study provides insights into data set composition and augmentation techniques for similar applications.

In their study on the practicality and accuracy of detecting defoliation of soybean from canopy-level images, Walker (2021) evaluated the performance of a modified Faster R-CNN with a VGG16 feature extraction network on a canopy-level database. The precision and recall values achieved were 25.16% and 65.00% respectively, indicating no detection occurred. The authors attributed this outcome to the use of canopy images of soybean leaves, rather than canopy images, while in our study, we primarily utilized single leaf images to enhance the performance of our models.

In another research conducted by Tetila et al. (2020), various machine learning models were compared to the deep learning models proposed for detecting and classifying soybean pests. The models included Support Vector Machine, Artificial Neural Network, Naive Bayes, K-Nearest Neighbors, Adaboost, and deep learning architectures. Among these, Support Vector Machine exhibited the highest accuracy of 60.49%. On the other hand, the deep learning approach utilizing Resnet-50 achieved an accuracy as high as 93.82%. Our study also compares deep learning and machine learning algorithms, aiming to achieve even higher accuracy scores for each model than those reported in previous studies.
These studies contribute to the field of plant disease detection and pest identification by leveraging machine learning techniques, particularly CNNs, to achieve accurate classification and automated diagnosis. The use of transfer learning, data engineering, and augmentation techniques demonstrates their potential in improving model performance and expanding the accessibility of disease diagnostics and pest management strategies in agriculture. They have also shown the superiority of deep learning models, such as VGG16, VGG16 and ResNet in these tasks.
3 METHODOLOGY

This section describes the data sets that were used for the study, the tools and materials, the different algorithms compared and the research design.

3.1 Data-set

The data sets that were used in this study were gotten from different sources online. The pest data set was readily available and free from a single source while the disease data sets were sourced from different sites that hold free data sets, some of which require permission to access.

3.1.1 Pest Data

For this study, the data set available in Mignoni et al. (2022) was used. There are 3 folders in the data set which are named, Caterpillar, Diabrotica and Healthy respectively. We added more images for healthy soybean images from Plant Village\(^1\) to increase the number as there are only 896 healthy images in (MIGNONI et al., 2022). The resulting number of images that were used in the analysis is shown in Table 3.1 below:

<table>
<thead>
<tr>
<th>Folder</th>
<th>Number of Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caterpillar</td>
<td>3,309</td>
</tr>
<tr>
<td>Diabrotica</td>
<td>2,205</td>
</tr>
<tr>
<td>Healthy</td>
<td>4,985</td>
</tr>
<tr>
<td>Total</td>
<td>10,499</td>
</tr>
</tbody>
</table>

Source: Author (2023)

According to Mignoni et al. (2022), the images were captured in January 2021, so they are very recent data. The capture took place at various stages of the plants, from budding to the adult stage, but before starting the yellowing phase of the leaves. As for the environment, the images were collected in adverse conditions, such as cloudy, sunny, drizzle, cloud shadows, and windy weather. The height of the image capture varied between 20 cm and 1 meter away from the plant. The data source already did some pre-processing on the images such as; Data annotation, Data Split, Image size standardization and Data augmentation. The image dimension is 500x500 and the majority of the leaf images were captured in the bunch as shown in Fig. 3.1 below.

\(^1\) https://plantvillage.org/
3.1.2 Disease Data-set

The data set for the disease classification was gotten from Plant Village\(^2\) and Forestry Images\(^3\) as was used in Gui e Mbaye (2019). These data sets’ repositories hold recent images of different soybean diseases. The data sets retrieved from these sites are not as many as in the pest data sets but they were individual leaf images finely showing the disease features. The image dimension varies from leaf to leaf, we downloaded each one at the highest resolution possible. For the disease classification, we used 600 images with each class, i.e., FLS, Asian Rust and the healthy class having 200 images each. The image examples are shown in Fig. 3.2

\(^2\) https://plantvillage.org/
\(^3\) https://www.forestryimages.org/
3.2 Data Preprocessing

To enhance the quality and consistency of the data that was gotten from different sources, different steps of data preprocessing were implemented to make it fit the algorithms that were implemented. The following tools and methods were used in preparing the data:

a) Data Sampling: We pulled in some images at random from the different folders to check and confirm the image properties for the data that came from different sources. Though the pest images from Mignoni et al. (2022) were all of dimension 500x500, the data from other sources vary from size to size. The step helped us to detect the diversities in the data set, and to have a detailed view of the images;

b) Data Transformation and Normalization: We transformed the data using CV2 to read and resize the images. All images were resized to 256x256 for suitability to practice with pre-trained models. We also used "matplotlib.pyplot" to convert the image colour to grayscale; The images in the folders were converted into a list data structure and the append function was used to add more data into the list. This was done so as to make the data fit the algorithms that will be applied to both deep and machine learning. The data was also labelled based on the different plant conditions that we are considering in the study;

c) Data Splitting: For this study, the K-fold cross-validation procedure was used on the data. The K value was 10 and this was done on the Train and the Validation Data sets. This was done to ensure inclusiveness, avoid over-fitting and also reduce the variance of accuracy. For each fold, in the pest data set, the training images were 7,778 and the validation images were 778 with 3 classes for each. i.e. (Caterpillar, Diabrotica and Healthy). While for the disease data set, the number of training and validation images were 488 and 44 images respectively for each fold. The number of test images was 1,943 and 108 for the Pest and Disease data sets respectively. Figure 3.3 shows an example of the first fold for the pest data set.

3.3 Model Training, Validation and Testing

This step is divided into two phases, training with Deep learning models and training with ordinary machine learning models.
3.3.1 Deep Learning Model

We implemented a custom CNN algorithm, 2 pre-trained algorithms, and 3 machine-learning algorithms for the disease and pest action classification. For all the algorithms, the data-splitting methods were the same, and we only changed the model used each time. The custom CNN algorithm was structured like what was used in (SAKIB et al., 2020). The layers of the CNN model include the Convolutional layers which is the main building block of the CNN algorithm. The model architecture consists of multiple layers of convolutional, pooling, and dropout layers, followed by fully connected dense layers with ReLU activation functions, and a final softmax activation layer for multi-class classification. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. For this study, we named this CNN architecture ViewNet. The model architecture is broken down as follows:

a) Input Layer: The input layer receives the images which are in the shape of (image rows, image columns, 3), where image rows and image columns represent the height and width of the input images, respectively, and 3 represents the number of colour channels (RGB);

b) Convolutional Layers: The model starts with a 3x3 convolutional layer with 64 filters and ReLU activation. Followed by another 3x3 convolutional layer with 64 filters and ReLU activation. For this study we have used the 2D convolutional layer which is most suitable for image data (KIRANYAZ et al., 2021). These Convolutional layers systematically apply learned filters to the input images in order to create feature maps that summarize the presence of the features in the input.

We then added a max pooling layer with a pool size of (2, 2) to reduce the spatial dimensions by half. The Max pooling layer calculates the maximum value for each patch of the
feature map and the purpose of this layer is to reduce the data to exactly what is needed in other to speed up computation. A dropout layer with a rate of 0.25 is added to randomly set 25% of the input units to 0 during training to prevent over-fitting;

c) Flatten Layer: The feature maps are flattened into a 1-dimensional vector so it can be inputted into the next layer;

d) Dense Layers: The flattened vector is passed through a fully connected dense layer with 16 units and ReLU activation, followed by a dropout layer with a rate of 0.1. Then, another dense layer with 32 units and ReLU activation is added, followed by a dropout layer with a rate of 0.2. This is repeated with another dense layer with 32 units and ReLU activation, followed by a dropout layer with a rate of 0.2. Finally, a dense layer with a softmax activation function is added with the number of units equal to the number of classes in the classification task (represented by the length of the class Labels list;

e) Output Layer: The model is compiled with the Adam optimizer, categorical cross-entropy loss function for multi-class classification, and accuracy as the evaluation metric.

The batch size is set to 64, and the model is trained for 50 epochs. The activation function used in the convolutional and dense layers is ’ReLU’.

The model architecture is as shown below;

Figure 3.4 – Deep Learning Architecture

Aside from this custom architecture, we used VGG16 and VGG16 pre-trained models on our data so as to achieve the goal of comparing the results.
The tools and libraries used include Jupyter Notebook, TensorFlow, Keras and SKLearn, Matplotlib etc.

### 3.3.2 Machine Learning

The ML algorithms were applied to the data sets after the deep learning algorithms, the data preparation process was the same, we only change the algorithms. The shape of the images however was flattened into a vector form to accommodate the regular machine-learning algorithms. Every other procedure remained the same. We used Support Vector Classifier (SVC), Random Forest Classifier (RFC) and KNN which were also imported from SKLearn. We also implemented the 10-fold cross-validation on the data set to split it into the training and testing sets for evaluating the different classifiers that were used on them.

### 3.3.3 Tools and Libraries Used

Below is the list of all the tools and libraries used in the classification experiment;

- **a)** Numpy: A library for numerical computing in Python, used for various array operations;
- **b)** Scikit Learn: Scikit-Learn, commonly abbreviated as SKLearn, is a popular and widely used open-source machine learning library for Python. It provides a broad range of tools for data preprocessing, feature engineering, model selection, and evaluation. We imported K-Fold and the 3 ML algorithms from this library.

  - SKLearn.metrics: Part of sci-kit-Learn, used for evaluating various classification metrics such as accuracy, F1-score, precision, recall, and confusion matrix.
  - SKLearn.model selection: Part of Scikit Learn, used for creating a StratifiedKFold object for cross-validation;
- **c)** PIL (Python Imaging Library): A library for image processing in Python, used for image-related operations;

  - ImageFile: Part of the PIL library, used for dealing with images that might raise decompression bomb warnings.
- **d)** Random: A Python module used for generating random numbers and performing random shuffling;

  - TensorFlow.Keras: The high-level Keras API provided by TensorFlow for building deep learning models. We imported VGG16 and VGG19 from this library;
f) Shutil: A Python module used for high-level file operations like moving and copying files and directories;

g) ImageDataGenerator: Part of the Keras library, used for real-time data augmentation during model training;

h) OS: A Python module used for interacting with the operating system, used for handling file and directory operations;

i) Warnings: A Python module used for managing warning messages in the code;

3.4 Model Evaluation

The evaluation of the effectiveness of each classifier in the machine learning algorithms will be performed by measuring the results of accuracy, recall, precision, and F-measure. According to Hossin e Sulaiman (2015), Juba e Le (2019) these evaluation metrics mean the following:

a) Accuracy: the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated. Accuracy is the most used evaluation metric either for binary or multi-class classification problems. An advantage is that it is easily computed, used and understood by humans while a major disadvantage is that produces less distinctive values. The equation for accuracy is represented in equation 3.1 below:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}
\]

The values for the variables True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), are extracted by a confusion matrix;

b) Recall: Recall is used to measure the fraction of positive patterns that are correctly classified. It is a better way of evaluating performance in cases of data imbalance. The equation reflects the ratio of True positives (TP) to all that is positive in the data set:

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{3.2}
\]

c) Precision: Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class. Just like in the case of Recall, it is also very useful with imbalanced data sets;

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{3.3}
\]
d) **F- Measure**: F- measure represents the harmonic mean between recall and precision values. It is also called F1 score. Since the harmonic mean skews towards the least element in the data, it helps to mitigate outliers in the data set;

\[
F\text{-measure} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}
\]

(3.4)

e) **ROC Curve AUC**: The ROC (Receiver Operating Characteristics) curve is gotten by plotting the true positive rate against the false positive rate. The AUC (Area under the Curve) reflects the overall ranking performance of a classifier, the higher the AUC, the better the classifier.

### 3.5 Model Deployment with Flask

Deploying a machine learning model makes its results accessible to users beyond the notebook environment, this process is commonly referred to as placing the model in a production environment. To achieve this, we leveraged the Flask framework, a web application framework written in Python (LAKSHAY, 2020).

Yaganteeswarudu (2020) proposed a multi-disease detection platform using the Flask framework, wherein each model’s behaviour was stored in Python pickle files. These pickles were then utilized to design the Flask API. On the other hand, Garg e Pundir (2021) developed a Graphical User Interface (GUI) using HTML and CSS, which interacted with the Flask framework as a query. The Flask framework, in turn, employed the trained model to predict values or labels based on input parameters, and the results were displayed on the website. This demonstrates the flexibility of the Flask framework and the various approaches that can be employed to implement it effectively.

Following a similar approach, we deployed our exceptional model to production. Additionally, we ensured to save all tested models, providing a robust backup for future changes or references.

We initialized a Flask application with Python after importing the necessary libraries (Flask, OS, TensorFlow, PIL and NumPy), we then defined the path to the folder containing the saved models.

We loaded the saved pest and disease models, We then defined a function to pre-process the uploaded image, resizing it to 256x256 pixels and normalizing its pixel values. We created HTML templates for the form that accepts the image to be predicted which is the home page, then the prediction page where the prediction is displayed.
We added an image background to the HTML templates for aesthetics. The list of classes (3) for the pests and diseases was provided which matches the classes used to train the models. The app was run on port 5000 as seen in figures 3.5 and 3.6 below.

We downloaded other images of the pest and disease condition that were not part of our initial data set to try the app and it gave the correct prediction each time. We were able to validate our trained models through this experiment because they gave the right classification for new images each time.

Figure 3.5 – Deployment Home Page

Source: Author (2023)
Figure 3.6 – Prediction/ Result Page

Uploaded Image prediction: Caterpillar

Source: Author (2023)
4 RESULTS AND DISCUSSION

This study compares the different machine learning algorithms in the detection of two of the most common pests and diseases of the soybean plant noting that early detection helps to reduce the damage level and production loss. We discovered that Soybean Frog-eye leaf spot disease and Soybean Asian Rust are one of the most common and significant diseases of the Soybean plant (GU et al., 2021; GODOY et al., 2016). Also, we were able to establish that Caterpillar and Diabritica Speciosa which were considered in this study are part of the economically significant pests that affect soybean (ALVARENGA et al., 2019; BERTI; GUERRA, 2016).

For early detection of diseases and pest actions in plants, different ML and DL algorithms were deployed, but as part of this study, we researched the most commonly used algorithms for pest and disease detection out of which we picked out SVM, KNN, RF and CNN. At least one of these algorithms was used in all cited literature for detecting plant pests or diseases. With these algorithms, we have been able to achieve higher classification accuracies than in some previous works.

4.1 Deep Learning Algorithms Results

The CNN models were chosen for being versatile producing great results in classification problems. The pre-trained VGG models performed better than the custom-made CNN architecture. We recorded the Accuracy, recall, precision and F-measure which to a large extent revealed the strength and performance of each algorithm.

The accuracy shows the proportion of correctly classified instances out of the total number of instances. The recall shows the proportion of true positive instances that were correctly identified out of all positive instances in the data set. The precision shows the proportion of true positive instances that were correctly identified out of all instances that were classified as positive by the algorithm. Finally, the F-measure column shows the weighted average of precision and recall.

For all 3 architectures, we implemented a Stratified K-Fold cross-validation approach for training and evaluating the models using the Keras library. We defined the number of splits to be 10, we then iterate over each fold, the images and their labels to load the training and validation data from the stated file directories. We trained the model using model.fit() on the training data with 50 epochs. To predict the labels for the validation data we used model.predict() and then calculate the performance metrics (accuracy, precision, F1-score, recall) using a custom-defined function. Also, each model architecture is defined using the getModel() function, which returns a Keras Sequential model with convolutional
and dense layers, configured with the Adam optimizer and categorical cross-entropy loss for multi-class classification.

For the custom model, a convolutional neural network (CNN) model for image classification using the Keras library was implemented. The model architecture consists of multiple layers of convolutional, pooling, and dropout layers, followed by fully connected dense layers with ReLU activation functions, and a final softmax activation layer for multi-class classification. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. The result of the experiment on the pest and disease data are shown in Table 4.1

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViewNet</td>
<td>0.89</td>
<td>0.89</td>
<td>0.90</td>
<td>0.89</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.95</td>
<td>0.96</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>VGG19</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>ViewNet</td>
<td>0.75</td>
<td>0.80</td>
<td>0.76</td>
<td>0.75</td>
</tr>
<tr>
<td>Diseases</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG16</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Source: Author (2023)

For detecting pests’ actions on the soybean leaves, our experiments show that the VGG16 and VGG19 algorithms achieved higher accuracy, recall, precision, and F-measure scores than the custom deep learning algorithm that we developed. Specifically, VGG16 achieved an accuracy of 0.95, a recall score of 0.96, a precision score of 0.95, and an F-measure score of 0.95, while VGG19 achieved an accuracy of 0.94, a recall score of 0.94, a precision score of 0.94, and an F-measure score of 0.94. In contrast, the ViewNet algorithm achieved an accuracy of 0.89, a recall score of 0.89, a precision score of 0.90, and an F-measure score of 0.89.

For detecting plant diseases, our experiments show that the VGG16 algorithm achieved the highest scores for all evaluation metrics, with an accuracy of 0.99, a recall score of 0.99, a precision score of 0.99, and an F-measure score of 0.99. VGG19 also achieved high scores, with an accuracy of 0.98, a recall score of 0.98, a precision score of 0.99, and an F-measure score of 0.98. The ViewNet algorithm also achieved lower scores, with an accuracy of 0.75, a recall score of 0.80, a precision score of 0.76, and an F-measure score of 0.75.

Overall, the results suggest that the VGG16 and VGG19 deep learning algorithms are more effective than the ViewNet algorithm for detecting pests and diseases in soybean plants, which supports the findings of (MKONYI et al., 2020) in their studies where VGG16 gave better performance than the other algorithms they were compared with.
The trend of the classification accuracy for each fold of the training and validation is visualized in figure 4.1.

Figure 4.1 – Training accuracies for the deep learning models

![Figure 4.1](image_url)

Source: Author (2023)

Figure 4.1 presents the performance of the three models (VGG16, VGG16, and ViewNet) on the pest data set, evaluated using 10-fold cross-validation. Each fold represents a separate evaluation of the models on different subsets of the data. The "Accuracy" column shows the accuracy of each model in percentage (%) for each fold. Overall, VGG16 achieved the highest accuracy, ranging from 96% to 100%, across all folds, demonstrating its consistent and superior performance compared to VGG16 and ViewNet. VGG16 exhibited the second-best performance, with accuracy varying between 94% and 99%, while ViewNet had the lowest accuracy, ranging from 87% to 95%. These results provide valuable insights into the models' performance, indicating that VGG16 is the most effective model for the data set, while VGG16 also performs well, and ViewNet lags slightly behind in accuracy.

4.2 Simple Machine Learning Algorithm Results

The ML algorithm also produced good classification scores for detecting the selected pests and diseases of the soybean plant. Using the same data sets and data transformation procedure, we applied three different machine learning algorithms (Support Vector Machine [SVM], Random Forest [RF], and K-Nearest Neighbors [KNN]) to the two different data sets (Pests and Diseases) we then recorded their accuracy, recall, precision, and F-measure.

For the Pests data set, the SVM algorithm achieved the highest accuracy, recall, precision, and F-measure, with a value of 0.88 for each metric. The RF algorithm had the second-highest performance, with an accuracy of 0.85, a recall of 0.85, a precision of 0.86, and an F-measure of 0.85. The KNN
algorithm had the lowest performance, with an accuracy of 0.76, a recall of 0.76, a precision of 0.77, and an F-measure of 0.76.

For the Diseases data set, the RF algorithm achieved the highest accuracy, recall, precision, and F-measure, with a value of 0.95 for each metric. The SVM algorithm had the second-highest performance, with an accuracy of 0.92, a recall of 0.92, a precision of 0.92, and an F-measure of 0.92. The KNN algorithm had the lowest performance, with an accuracy of 0.74, a recall of 0.74, a precision of 0.82, and an F-measure of 0.74.

Table 4.2 – Result Comparison for Simple Machine Learning Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>RF</td>
<td>0.85</td>
<td>0.85</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>KNN</td>
<td>0.76</td>
<td>0.76</td>
<td>0.77</td>
<td>0.76</td>
</tr>
<tr>
<td>SVM</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Diseases</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>KNN</td>
<td>0.74</td>
<td>0.74</td>
<td>0.82</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Source: Author (2023)

In our experiment, we observed that the disease data sets consistently outperformed the pest data set in both the training and testing phases, despite the general assumption that a larger data set would yield better results. To investigate this, we reduced the data set size to match that of the disease data set, and interestingly, the classification results decreased as well, aligning with findings from previous research (POWER et al., 2018; TAHA et al., 2021). Several factors could explain why a model with fewer data performed better, such as overfitting, data noise, and sampling bias (HE et al., 2016; GONG et al., 2019; PENG et al., 2020).

To gain further insights, we compared five random samples from the disease data set and the pests data set. It became evident that the disease data set’s samples had more distinct features compared to those from the pest data set. Notably, many images in the pest data set depicted full soybean plants, while the disease data set comprised individual leaf images with distinct features. Although we cannot draw definitive conclusions from our pest data set, it is worth considering that having more pest data actually led to improved classification results.

Throughout this study, we extensively tested and validated popular image recognition algorithms with the aim of achieving better classification results compared to prior works. Additionally, we explored a relatively new data set of pest actions on the soybean plant (MIGNONI et al., 2022). This work underscores the scarcity of research on soybean pests, which pose a significant threat to production, compared to the abundance of studies on soybean diseases. We conducted a comparison of our results
with existing works on plant diseases, but we could not find any previous research specifically focused on pest actions on plants like the data set we utilized. Our study presents a valuable addition to the literature on identifying plant pests as well as diseases, and Table 4.3 showcases our higher accuracies compared to various studies in the field.

Table 4.3 – Result Comparison with Previous Works

<table>
<thead>
<tr>
<th>Article Title</th>
<th>Authors</th>
<th>Plant Phenomenon</th>
<th>Model Used</th>
<th>Result</th>
<th>Results from this Study</th>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(PAJURI, KUMAR, THOTTOLIL, 2022)</td>
<td>Plant Disease</td>
<td>VGG 16</td>
<td>96.6%</td>
<td>99.7%</td>
<td>PlantVillage Dataset</td>
</tr>
<tr>
<td></td>
<td>(SUJATHA et al., 2021)</td>
<td>Plant Disease</td>
<td>VGG 16</td>
<td>89.5%</td>
<td>99.7%</td>
<td>Manually gathered dataset</td>
</tr>
<tr>
<td></td>
<td>(GUI, MBAYE, 2019)</td>
<td>Soybean Disease</td>
<td>VGG 19</td>
<td>93.5%</td>
<td>99.7%</td>
<td>Online data sources: Plant Village and Forestry image</td>
</tr>
<tr>
<td></td>
<td>(BEVERS, SIKORA, HARDY, 2022)</td>
<td>Soybean Disease</td>
<td>VGG 16</td>
<td>90.8%</td>
<td>99.7%</td>
<td>Original field images</td>
</tr>
<tr>
<td></td>
<td>(TETILA et al., 2020)</td>
<td>Soybean Pest</td>
<td>VGG 16</td>
<td>91.8%</td>
<td>95.2%</td>
<td>Real field condition</td>
</tr>
<tr>
<td></td>
<td>(SIVAKUMAR et al., 2020)</td>
<td>Soybean Disease</td>
<td>RFC</td>
<td>83.3%</td>
<td>95.2%</td>
<td>Low-altitude UAV imagery</td>
</tr>
<tr>
<td></td>
<td>(ALATAWI et al., 2022)</td>
<td>Plant Disease</td>
<td>VGG16</td>
<td>95.2%</td>
<td>99.7%</td>
<td>PlantVillage</td>
</tr>
<tr>
<td></td>
<td>(WALKER, 2021)</td>
<td>Soybean Pest</td>
<td>VGG 16</td>
<td>89% Precision</td>
<td>95% Precision</td>
<td>Original UAV images</td>
</tr>
</tbody>
</table>

Source: Author (2023)

Each row in the table 4.3 represents a different research article, showcasing essential details such as the article title, authors, the plant phenomenon investigated, the machine learning model employed, and the corresponding accuracy results. This table allows a quick assessment of the performance of the different models we have used in comparison to how they were used in previous studies. Additionally, it highlights the data sources used in each study, offering valuable insights into the data sets utilized for
training and evaluating the machine learning models. None of the studies with originally sourced data has their data accessible online. This is a valid challenge of plant image classification.
5 CONCLUSION

This study compares the performance of various machine learning (ML) algorithms for detecting pest actions and diseases in soybean plants. Soybean Frog-eye leaf spot disease and Soybean Asian Rust were chosen as the common and significant diseases of the Soybean plant, while Caterpillar and Diabrotica Speciosa were chosen as the economically significant pests that affect soybean. The ML algorithms used for detecting pests and disease actions were Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and Convolutional Neural Network (CNN).

One of the goals of this study was to emphasise the importance of early detection in the soybean field and not just disease detection alone but also the destructive actions of the pest. To achieve this aim, we were able to train ML models on both pest and disease data sets separately. The outcome of which we hope reduces the different effects of uncensored use of herbicides and pesticides which include:

a) Accumulation of residues on the crop which may affect their nutritional organic status;
b) Depletion of soil nutrients thereby making reduced nutrients available for plant growth and eventually reducing the yield from the plant;
c) Genetic Modification: Studies have shown that some herbicides are potent enough to affect the genetic nature of some crops.

Another objective of this study was to do a comparative analysis of the different ML techniques that were chosen. CNN models were found to be versatile and produced better results than other algorithms for classification problems. Three architectures of CNN models were implemented, a custom deep learning algorithm called ViewNet, VGG16, and VGG19. A Stratified K-Fold cross-validation approach was applied for training and evaluating the models, and the performance was evaluated using accuracy, recall, precision, and F-measure. VGG16 and VGG19 algorithms performed better than the custom deep learning algorithm.

VGG16 achieved an accuracy of 0.95, a recall score of 0.96, a precision score of 0.95, and an F-measure score of 0.95 for detecting pests actions, and an accuracy of 0.99, a recall score of 0.99, a precision score of 0.99, and an F-measure score of 0.99 for detecting plant diseases. VGG19 achieved an accuracy of 0.94, a recall score of 0.94, a precision score of 0.94, and an F-measure score of 0.94 for detecting pests actions, and an accuracy of 0.98, a recall score of 0.98, a precision score of 0.99, and an F-measure score of 0.98 for detecting plant diseases.

The ViewNet algorithm achieved an accuracy of 0.89, a recall score of 0.89, a precision score of 0.90, and an F-measure score of 0.89 for detecting pests’ actions, and an accuracy of 0.75, a recall score of 0.80, a precision score of 0.76, and an F-measure score of 0.75 for detecting plant diseases.
Our proposed custom deep learning algorithm, ViewNet, demonstrated strong performance in the
detection of both pests and disease data sets. These results have shown to demonstrate the effectiveness
and reliability of ViewNet in accurately identifying and classifying both pests and diseases in the given
datasets.

In comparison to other state-of-the-art deep learning models, such as VGG16 and VGG19, Vi-
iewNet exhibits competitive performance though it has lower classification results than them.

The high recall and precision values achieved by ViewNet demonstrate its ability to minimize
false negatives and false positives, respectively, thereby enhancing its suitability for the precise detection
of pests and diseases in agricultural environments. These results underscore the potential of our custom
ViewNet algorithm to make a valuable contribution to the field of plant disease and pest detection.

In conclusion, our ViewNet algorithm has shown promising performance, further development
and fine-tuning of ViewNet could potentially lead to even higher accuracy and reliability, making it a
valuable tool for agricultural applications in the future.

For future works, we hope that other studies further enhance and expand this study on plant
disease and pest detection using deep learning algorithms. To do this action can be taken in the following
areas:

a) Model Fine-tuning: Fine-tuning the existing deep learning models, including ViewNet, on
specific plant diseases and pests holds the potential to significantly improve their accuracy
and detection performance. Tailoring the models to the unique characteristics of different
plant species and diseases can lead to more precise and reliable predictions;

b) Sharing Research Images: Studies that utilized original image datasets can contribute to
the community by making these datasets publicly available in online repositories. Sharing
research data fosters collaboration and allows other researchers to build upon the work and
advance the field;

c) Real-time Deployment: Optimizing deep learning models for real-time deployment on
edge devices or embedded systems is crucial for practical agricultural applications. This
will enable swift and efficient plant disease and pest detection in the field, facilitating
timely intervention measures;

d) Disease and Pest Diversity: To validate the generalizability of the models, future studies
should conduct experiments on a larger variety of soybean pests and diseases. Exploring
a wider range of pests and diseases will provide more robust evidence of the model’s
effectiveness in diverse environmental conditions environments and climates;
e) Focus on Soybean Pests: This study highlights the prevalence of research on soybean disease classification compared to pest or pest actions on leaves classification. Future studies can address this imbalance by focusing on including pests in plant image classification works. Understanding and classifying pests is vital, as they can significantly impact crop yield and productivity;

f) User-friendly Interfaces: For wider adoption and practical usage, deploying the models into web applications with user-friendly interfaces can greatly benefit farmers and agricultural practitioners. User-friendly interfaces will allow easy interaction with the models and enhance usability.

By considering these recommendations, future research in plant disease and pest detection using deep learning can advance the field, leading to more accurate, efficient, and practical solutions that benefit the agricultural industry and contribute to global food security.
REFERENCES


MUELLER, D. Iowa State University. 2012.


SIKORA, E. 2021.


