UNIVERSITY OF CAPE TOWN

Application of CNN-gcForestCS to cassava leaf image classification

Student:
Liam Carew
CRWLIA001

Supervisor:
Mr Stefan S Britz

Minor dissertation for M.Sc. Data Science

at the

DEPARTMENT OF STATISTICAL SCIENCES

UNIVERSITY OF CAPE TOWN

Abstract

Faculty of Science
Department of Statistical Sciences

M.Sc. Data Science

Application of CNN-gcForestCS to cassava leaf image classification

by Liam CAREW

Cassava is one of the most consumed carbohydrates in the world, providing a reliable source of income and nutrition to inhabitants of Latin America, Africa and Asia. However, its production is greatly affected by pathogenic infection with cassava mosaic disease (CMD) posing the greatest threat to cassava farmers in Africa and Asia. Given that developing nations are estimated to be hit hardest by climate change and projected to have the largest population increases in coming decades, optimisation of cassava yield in these areas is imperative to ensure food security. Traditionally, crop health is determined by manual inspection which can be laborious, error-prone and require technical expertise. This produces a costly barrier of entry for smallholding farmers who make up majority of global cassava production.

Development of automated disease detection systems using convolutional neural networks (CNNs) deployable on mobile phones have shown to be a cost-efficient and effective method for cassava monitoring, mainly owing to their advanced feature extraction capabilities. However, CNNs require complex hyperparameter tuning and can be computationally intensive to train. GcForestCS (multi-grained cascade forest with confidence screening) presents an alternative statistical learning method that can be trained using CPU, and requires less complex hyperparameter tuning than deep learning while producing competitive performance for lower-dimensionality datasets. Taking advantage of the feature extraction capabilities of CNNs and the competitive performance of gcForestCS for lower-dimensionality datasets, the central aim of this dissertation was to investigate CNN-gcForestCS as an alternative to deep learning for cassava leaf disease detection.

The performance of CNN-gcForestCS was compared to gcForestCS and deep learning where the effect of class balance, CNN feature extraction, CNN feature extractor fine-tuning, pooling after multi-grained scanning, and training set curation were assessed. The results showed that the best DenseNet201-gcForestCS model (86.79%) produced marginally worse performance than the best DenseNet201 model (87.43%), while the best MobileNetV2-gcForestCS model (83.66%) produced marginally better performance than the best MobileNetV2 model (82.87%).

Overall, the results indicate that it is inconclusive whether CNN-gcForestCS is a viable alternative to deep learning for cassava leaf disease detection, especially when considering the high computational cost associated with the CNN-gcForestCS methodology.

Acknowledgements

To begin, I would like to thank my supervisor, Stefan Britz, for his constant calming and guiding presence throughout the duration of this dissertation, and for teaching me invaluable lessons about the art of storytelling and how to use it to communicate your research purpose which I will continue to nurture as I progress in my career.

Secondly, I would also like to acknowledge the University of Cape Town (U.C.T.) Postgraduate Funding Office for providing the funding that made the following research possible. I'd also like to thank the ICTS High Performance Computing team at U.C.T. for maintaining and allowing me access to state-of-the-art computing facilities that were used to train models within this dissertation.

Thirdly, I would like to thank the Artificial Intelligence (AI) lab at Makerere University in Kampala, Uganda for collating the image data used in this work, and making the dataset publicly accessible. I'd also like to acknowledge the hard work of contributors at the National Crops Resources Research Institute (NaCRRI) for providing labels for the images within this dataset.

Lastly, I would like to thank my family, especially my parents Sandi and Martin, for helping me through the emotional ebbs-and-flows I experienced while completing this thesis, and reminding me that I am good enough and can achieve anything I put my mind to.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cassava disease detection

1.1.1 The issue of food insecurity

It is estimated that between 720 and 821 million people are undernourished, with food insecurity being projected to affect approximately 30% of the global population (FAO et al. 2020). Of those affected, 21% belong to African nations —a statistic that has been increasing in recent years (FAO et al. 2020; FAO et al. 2021). The issue of food insecurity, especially in the developing world, is so far-reaching that it directly forms part of the United Nation's (UN) Sustainable Development Goals (SDGs) which the organisation aims to achieve by 2030, specifically Sustainability Goal 2: Eradication of Hunger. Other sustainability goals affected include eradication of poverty (SDG1), providing clean water and sanitation (SDG6), sustainable land use (SDG11), responsible production and consumption (SDG12), mitigating climate change (SDG13), and sustainable life on land and water (SDG14 and SDG15)(Vågsholm, Arzoomand, and Boqvist 2020; UN 2018). The need for food security is projected to increase between 36%–54% in the coming decades, with food production needing to increase by 25%–70% by 2050 to keep up with demand and allow for affordable access to staple foods (Hunter et al. 2017; van Dijk et al. 2021).

1.1.2 Importance of agriculture

One estimate shows that 90% of nutritional calories come from agricultural output, reflecting the importance the agricultural sector plays in ensuring food security (Cassidy et al. 2013). The development of an agricultural sector allows countries to engage in trade with each other, stimulating economic growth, creating jobs, and alleviating the effects of poverty (Irz et al. 2001; van Ittersum et al. 2016; Evans 2019). The agricultural sector is often espoused as a keystone toward economic empowerment of developing nations, reflected by its considerable contribution to gross domestic product (GDP) in countries such as Nigeria and India (Mathur, Das, and Sircar 2006; Ijirshar 2015). Fostering agricultural sectors in developed nations requires a significant level of upfront investment which, if done properly, has shown to encourage economic development in third-world countries (Mathur, Das, and Sircar 2006; Ijirshar 2015).

Given that Sub-Saharan Africa currently accounts for 13% of the global population – another statistic that is likely to grow in the coming decades – food security will become an ever greater concern in this region (van Bavel 2013; Gu, Andreev, and Dupre 2021). In the Sub-Saharan African context, agriculture is dominated by western African nations, which account for 60% of agricultural production (Ayanlade and Radeny 2020). This disparity is seen as being due to scaling of small-holding farms to medium-scale farms (MSFs) in these more land-dominant nations, allowing for larger food export volumes to be achieved

(Deininger and Byerlee 2012; Jayne, Chamberlin, and Headey 2014; Otsuka, Nakano, and Takahashi 2016), as well as the low levels of agricultural expansion in other areas which are correlated with low monetary returns for farmers and land fragmentation from population expansion (Giller 2020). In terms of subsistence farming, 30%–34% of global food production is produced by 'small-holding' farms [1] (Ricciardi et al. 2018), while 'family farms' are said to account for approximately 80% of global crop production (Lowder, Sánchez, and Bertini 2021). Increase in agricultural output in this region in the recent past has mainly been due to cultivation over larger areas of land and not due to more efficient use of current land resources (Sanchez 2002; Grassini, Eskridge, and Cassman 2013).

1.1.3 Challenges in staple crop production

Despite the expansion of agricultural production in certain regions of Sub-Saharan Africa, malnutrition still remains high. Given that climate change is expected to cause further land degradation in years to come, farming one's own food as subsistence farmers – or providing access to food at affordable prices – may become more vital and difficult to attain in the coming years (Mechiche-Alami and Abdi 2020). Along with the effects of climate change and land degradation, the COVID-19 pandemic has placed further pressure on agricultural output of smallholding farmers in Sub-Saharan Africa who are dependent on trade as a source of income (Ayanlade and Radeny 2020). Increases in disease incidence, which is also linked to climate change, also affects food production, with pathogenic infections causing approximately 25% yield reduction worldwide (Oerke and Dehne 2004; Sundström et al. 2014).

Sub-Saharan Africa, and especially western Africa, is a particularly susceptible region to the effects of climate change. A study on the effects of climate change from 2000 to 2009 on sorghum and millet, which are staple crops in West Africa, showed that an approximate $1^{\circ}C$ increase in temperature was associated with an estimated 10%–20% yield reduction in millet and 5%–15% reduction in sorghum (Iizumi et al. 2017). This indicates that farmers will need to alter their methods of farming (and potentially the crops they farm) to reduce the detrimental effects of climate change on production and hunger statistics (UN 2015; Sultan and Gaetani 2016). Given that the agricultural centre of Sub-Saharan Africa is located in western Africa (Ayanlade and Radeny 2020) and that climate change is likely to affect this area most, farmers in these areas will likely need to adapt their farming practices and the types of crops they grow to allow for the surrounding populations to have adequate access to nutrition (Godlee 1991; Butler 2016).

In addition to the effects of climate change on crop cultivation, agricultural land usage has shifted primarily toward livestock production and is expected to increase as changes in global wealth and resultant dietary patterns occur – as more individuals enter the middle class, diets tend to shift more toward animal-based products, fruits and vegetables (Smith et al. 2010; Godfray et al. 2018; Kawabata et al. 2020). These changes in land usage are placing pressure on crop production, leading to ecosystems of the surrounding environment being cleared to produce more farming land. These ecological trade-offs may result in further disturbance to surrounding biomes which tend to act as 'carbon sinks', and therefore clearing these land areas could amplify the effects of climate change (Moyer and Bohl 2019; Yang et al. 2020). In light of the above information, there will be an increasing demand in the coming decades to improve yield efficiency from the limited farming land areas available for agricultural crop production to facilitate sustainable food production (Johnson 1997).

[1] Farms smaller than 2 hectares.

1.1.4 Cassava: challenges & potential solutions

Cassava (*Manihot esculenta* Crantz) presents a potential partial solution to the problem of food insecurity, since it can be easily grown with little maintenance by subsistence farmers and produce adequate yield in spite of adverse biotic constraints such as poor soil quality, drought conditions, and low rainfall (Cock 1982; Hillocks et al. 2001; El-Sharkawy 2006; Chavarriaga-Aguirre et al. 2016). It is mainly grown for its nutrient-dense roots, and is estimated to form part of the diets of 800 million people worldwide (Tomlinson et al. 2018).

Despite the advantages of cultivating cassava in our changing global climate, its production, particularly in Sub-Saharan Africa, is greatly affected by pathogenic spread and disease, which makes crops inedible. Cassava Mosaic Disease (CMD) and Cassava Brown Streak Disease (CBSD) in particular negatively affect yield. Indicators of disease post-infection include changes in plant, and particularly leaf, physiology where differences between diseases are typically subtle, and can lead to mischaracterisations if inspections are done manually.

This setting presents an ideal application for an image-based classification model based on labelled cassava leaf images that can be deployed on a mobile application to be used by field workers. This has previously been done with deep learning models which have shown promising results (Mwebaze and Owomugisha 2017; Enkvetchakul and Surinta 2021; Dhivyaa, Kandasamy, and Rajendran 2022; Huertas-Tato et al. 2022). gcForest (multi-grained cascade forests), as proposed by Zhou and Feng (2017), is a tree-based statistical learning technique that has been postulated as an alternative to deep learning when inputs dimensions are small. This research investigates combining convolutional neural network (CNN) feature extraction from transfer learning models with a gcForestCS (multi-grained scanning cascade forests with confidence screening) classifier.

1.2 Aim

The focus of this research is to determine whether a CNN-gcForestCS image classification model is able to compete with, or outperform, deep transfer learning models for the purposes of detecting disease types present in cassava leaf images. Models are trained and evaluated on 9430 cassava leaf images that contains four disease classes and one healthy class (Mwebaze and Owomugisha 2017). The dataset is discussed in detail in Chapter 3. Given that transfer learning models have performed well for cassava leaf disease detection in previous work, and that CNN-gcForest has previously outperformed deep learning for a multi-class image classification task (Boualleg, Farah, and Farah 2019), CNN-gcForestCS may allow for an improvement over existing deep learning methods for cassava leaf disease detection.

In addition, the effect that different components of the data pipeline (Figure 4.1) have on CNN-gcForestCS model performance are assessed. This includes: class balance; CNN feature extraction and feature map size; CNN feature extractor fine-tuning; addition of a pooling step after multi-grained scanning; and curation of the training set.

1.3 Research questions

In this dissertation, there are 6 research questions with respect to cassava leaf disease detection that are investigated across a variety of model combinations depicted in the data

pipeline (Figure 4.1, and Tables 4.1 and 4.2). Each research question is outlined below, along with a brief explanation of how it is to be explored:

1. **What is the effect of class balance on the performance of CNN-gcForestCS?**

   The original training set is highly imbalanced, which may introduce bias during model fitting. To assess whether this class imbalance affects model performance, equivalent model combinations are trained on either the original imbalanced training set, or a balanced version of the training set (a description of how the training set was balanced is provided in Subsection 3.5.3). The test set performance of these models are compared according to weighted F1-score, with the expectation being that model generalisability will improve when trained on a balanced training set.

2. **What is the effect of CNN feature extraction, and feature map size, on the performance of gcForestCS?**

   To assess whether CNN feature extraction improves model performance of a gcForestCS classifier, a gcForestCS classifier is trained on the resized version of the training images (with shape $224 \times 224 \times 3$) where image processing via a CNN feature extractor has not occurred. These results are compared to CNN-gcForestCS model combinations for different feature map sizes (28×28, 14×14, and 7×7) extracted from different convolutional layers contained within the transfer learning models used. The expectation is that smaller feature map sizes will lead to better CNN-gcForestCS model performance.

3. **What is the effect of CNN feature extractor fine-tuning on the performance of CNN-gcForestCS?**

   To assess whether fine-tuning the weights of CNN feature extractors improves CNN-gcForestCS model performance, model generalisability of equivalent CNN-gcForestCS model combinations that differ based on whether or not the CNN feature extractor had been fine-tuned are compared. The expectation is that fine-tuning will lead to better CNN-gcForestCS model generalisation to unseen data than using the default ImageNet weights.

4. **What is the effect of adding a pooling step after multi-grained scanning on performance of CNN-gcForestCS?**

   The addition of a pooling step after multi-grained scanning allows for dimensionality reduction, which may lead to improved model performance of a gcForestCS classifier given that the cascade module of gcForest (see Subsection 4.2.2) tends to train better when input dimensions, especially height and width, are small (Zhou and Feng 2017). To assess the effect of this, the test performance of equivalent model combinations which only differ based on whether a pooling step was added after multi-grained scanning are compared. The expectation is that there will be an improvement in model performance from adding a pooling step after multi-grained scanning.

5. **What is the effect of training set curation on the performance of CNN-gcForestCS?**

   Upon further investigation of the training set, many images either have poor focus or exposure on the leaves, reducing the quality of the data on which the models are trained. To assess the effect of removing these images, the test performance of model combinations that only differed based on whether they were trained on curated data

are compared, with the expectation being that models trained on curated data will generalise better than those that are not.

6. **Is CNN-gcForestCS a viable alternative to deep learning for this problem?**

   This is the central research question that this dissertation aims to answer. The test set performance of deep learning benchmarks in previous work, as well as the best deep transfer learning model combinations (DenseNet201 and MobileNetV2) trained in this work are compared to the equivalent best CNN-gcForestCS model combinations based on overall accuracy and weighted F1-score (these performance metrics are discussed in Subsections 4.3.1 and 4.3.2).

1.4 Dissertation outline

This chapter provided an introduction to and context around the topic of investigation, as well as the research aims and questions that will be assessed. Chapter 2 outlines the literature related to plant leaf disease detection, cassava, and image classification, while Chapter 3 provides information regarding the dataset used, class distribution, limitations of the dataset, and the modifications made to the dataset to reduce the effect these limitations may have during modelling. Chapter 4 discusses the data pipeline, technical details relating to the CNN and gcForestCS methodologies, the performance metrics used, the hardware & software used during training, and the hyperparameter search spaces used during gridsearches for each model combination. Chapter 5 presents and assesses the results with respect to the research questions outlined above, after which Chapter 6 concludes the dissertation, providing answers to the research questions, considerations for future research, and final remarks.

Chapter 2

Literature Review

Chapter 2 provides a collation of the relevant literature on the topic of cassava leaf disease detection. Section 2.1 will outline literature related to plant leaf disease detection which will begin by describing the mechanics of pathogen transmission and infection (Subsection 2.1.1) and how this relates to plant immunity and the environment in the 'disease triangle' (Subsections 2.1.2 & 2.1.3). The impacts and mitigation of plant disease development will then be discussed (Subsections 2.1.4 & 2.1.5) before outlining literature specific to cassava as a crop and the consequences of disease development and detection (Section 2.2).

Following on, Section 2.3 will outline the relevant literature relating to image classification which will begin by briefly discussing the importance of convolutional neural networks (CNNs) in image-based tasks (Subsection 2.3.1) before discussing their importance in plant leaf disease detection (Subsection 2.3.2). Finally, literature relating to tree-based image classification (Subsection 2.3.3) and more specifically the applications of gcForest (Subsection 2.3.4) will be outlined.

2.1 Plant leaf disease detection

2.1.1 Mechanics of plant-pathogen transmission

Plants are sessile organisms constantly exposed to evolving pathogens and pests that threaten their growth and reproduction. Common means by which these pathogens are transmitted to plant hosts are via air (airborne transmission), water droplets (waterborne transmission), soil (soil-borne transmission), vectors (vector-borne transmission) or seeds (seed-borne transmission) with vector-borne transmission by insects of the order *Hemiptera* being the most prevalent (Jones 2003; Mauck et al. 2012; Perilla-Henao and Casteel 2016; Huang et al. 2020; van Munster 2020). Human-mediated plant-pathogen transmission from seeds or cuttings also occurs via global trade networks at post-harvesting stages of the agricultural cycle, resulting mostly from poor phytosanitary[1] practices (Sastry and Zitter 2014; Mumford, Macarthur, and Boonham 2015; Constable et al. 2018).

Once a pathogen has been transmitted to a plant host, the manner in which it enters the plant differs based on the type of microbe (Mauck et al. 2012; Fawke, Doumane, and Schornack 2015; Perilla-Henao and Casteel 2016; Willsey, Chatterton, and Cárcamo 2017; Wielkopolan, Jakubowska, and Obrępalska-Stęplowska 2021). During insect vector-borne transmission, pathogens are transported into the plant's phloem tissue by a feeding tube called a stylet which allows the first barrier of plant immunity provided by a waxy cuticle to be bypassed (Perilla-Henao and Casteel 2016; Jiang et al. 2019; Huang et al. 2020). If pathogens are able to spread within the plant via the plant vascular system and reproduce in plant tissue, insects

[1] Phytosanitation involves agricultural quality control protocols of crops entering and leaving farms. More specifically, this involves inspections and cleaning of produce

that then feed on this tissue can become vectors and continue the transmission cycle to other plant hosts (Perilla-Henao and Casteel 2016). It is important to note that plants can be infected by multiple different pathogens simultaneously (called co-infection) which can lead to the selection of highly virulent pathogen variants during intra-host resource competition (López-Villavicencio et al. 2011; Bujarski 2013; Zinga et al. 2013; Tollenaere, Susi, and Laine 2016).

With the mechanics of pathogen transmission in mind, Subsection 2.1.2 will outline the methods employed by plants to prevent entry and disease development.

2.1.2 Plant immunity and mechanisms of pathogen evasion

The first defense mechanism used by the plant to prevent infection is a physical barrier in the form of a waxy cuticle or bark around the exterior of the plant. If a pathogen is able to evade this barrier and enter plant cells, the plant mounts an innate immune response to limit pathogen propagation (Jones and Dangl 2006). The type of innate immune response differs based on the type of microbial infection where bacterial/fungal/oomycetic infections lead to PAMP-triggered immunity (PTI) and effector-triggered immunity (ETI) responses (Jones and Dangl 2006; Zipfel 2014; Savatin et al. 2014; Niehl et al. 2016; Nishad et al. 2020) while viral infections lead to antiviral RNA interference (RNAi) (Yang et al. 2002; de Alba, Elvira-Matelot, and Vaucheret 2013; Zhang et al. 2015; Paudel and Sanfaçon 2018) and salicylic acid-mediated response (SAR) (Yang et al. 2002; de Alba, Elvira-Matelot, and Vaucheret 2013; Nicaise 2014; Zhang et al. 2015; Calil and Fontes 2017; Paudel and Sanfaçon 2018).

Pathogens aim to evade PTI, ETI and SAR through the synthesis of effector proteins which bind to and 'deactivate' proteins involved in the plant's innate immune response (Jones and Dangl 2006; Nishad et al. 2020) while viruses can develop RNAi resistance through the expression of viral suppressors of silencing (VSR) proteins produced via the plant's protein synthesis pathways (Csorba, Kontra, and Burgyán 2015; Paudel and Sanfaçon 2018). A constant evolutionary battle between plant and pathogen occurs where plants adapt by naturally selecting for genetic variants[2] that can produce receptors to detect effector and VSR proteins. This allows them to maintain immunity to foreign pathogens, making them more evolutionary 'fit' than individuals that cannot. In response to this, pathogens can rapidly evolve to evade detection by the plant's immune system, allowing them to grow and propagate in the plant host (Jones and Dangl 2006).

A drawback that results from plants mounting an immune response is the physical damage that occurs at sites of infection. In addition to the immunological strategies described above, certain plant species are able to reduce the collateral damage of mounting an immune response through selection of variants that are able to 'tolerate' infection —this is when pathogenic infection still occurs but the immune response mounted does not have an effect on the plant's evolutionary fitness (Little et al. 2010; Mikaberidze and McDonald 2020).

With Subsections 2.1.1 & 2.1.2 having described the specifics of host-pathogen interactions, the following subsection will outline how the environment affects disease epidemiology.

[2] In genetics, genes may mutate during replication cycles which leads to variation in the gene(s). Individuals in a population that have the mutated version of a gene are called 'variants' as they differ from the typical genetic composition of the population.

2.1.3 Environmental factors affecting transmission and disease progression

In addition to plant-pathogen interactions, the environment forms an additional factor that affects disease epidemiology to form the so-called 'disease triangle' because both biotic and abiotic environmental stressors affect the extent of pathogen transmission and disease progression (Velásquez, Castroverde, and He 2018). Examples of biotic stresses include the abundance and behaviour of vectors (Eigenbrode, Bosque-Pérez, and Davis 2018), the genetic susceptibility of crops (Savary et al. 2006; Joshi et al. 2011; Avelino et al. 2015; Ekroth, Rafaluk-Mohr, and King 2019), and wounds created by the feeding of other vectors. Examples of abiotic stresses include soil quality and pH (Oldfield, Bradford, and Wood 2019), climatic factors such as temperature and rainfall (Burdon and Zhan 2020), and atmospheric CO_2 levels.

Biotic and abiotic factors work in concert to either improve or reduce evolutionary fitness of the plant while transmission pathways tend to increase. Evidence to support this claim includes: an increase in pathogen prevalence occurring in regions where temperature increases and changes in rainfall occur (Burdon and Zhan 2020); temperature changes affecting the bacterial ETI response (Cheng et al. 2013); the rapid emergence of novel viruses may be due to increasing globalisation of trade networks (especially exports from Sub-Saharan Africa) and climate change (Rodoni 2009; Jones, Goodin, and Verchot 2020; Trebicki 2020); or that the expansion and colonisation by pathogens and vectors over larger regions is associated with changes in climatic conditions (Bebber 2015).

With the literature relating to how plant disease comes about outlined above, the following subsection will describe the consequences of plant disease development.

2.1.4 Effects of disease development

The extent to which disease develops depends on a multitude of factors across the different components of the disease triangle. At the level of the plant host, disease leads to changes in physiology in the form of necrosis, wilting, hyperplasia, and hypoplasia (amongst others) that result from the plant mounting an innate immune response (Nazarov et al. 2020). These physiological changes produce dead plant tissue which becomes beneficial for feeding by certain pathogens such as bacteria and fungi (Velásquez, Castroverde, and He 2018).

At a wider scale, disease development during agricultural cycles is estimated to cause global yield losses for staple foods of 10.1%–28.1% for wheat, 24.6%–40.9% for rice, 19.5%–41.1% for maize, 8.1%–21.0% for potatoes and 11.0%–32.4% for soybeans (Savary et al. 2019). Viral, bacterial and fungal infections are estimated to have caused a 25% yield reduction worldwide (Oerke and Dehne 2004) with viral infections being estimated to constitute 47% of worldwide plant epidemics, causing the most considerable monetary loss for individuals who depend upon crop yields for income (Anderson et al. 2004; Sastry and Zitter 2014). Large crop losses can also lead to food insecurity, especially in the developing world, due to low supply and subsequent exorbitant food prices that have led to economic depression and social unrest in certain instances (Fedoroff 2015; Bellemare 2015; Hendrix and Haggard 2015).

As outlined in this subsection, the effects of disease development lead to wider effects on the surrounding communities if the plant is a staple food source. Subsection 2.1.5 will outline how disease is typically managed to mitigate the consequences of disease development.

2.1.5 Disease management and crop diagnostic techniques

In an effort to minimise the effects of disease development as outlined above, preventative measures during the agricultural production cycle are required to reduce the spread of disease along global trade networks (Anderson et al. 2004; Brasier 2008; Miller, Beed, and Harmon 2009). Typical agricultural disease control measures include visual inspections during cultivation (Wang et al. 2012), vector population surveillance, plant vaccination (Nicaise 2014), improvements in quality control measures post-harvest (Jones 2021), use of pesticides, the development and planting of disease-resistant plant varieties (Barrett et al. 2009; Brown 2015; Heck 2018), routine laboratory testing (Ward et al. 2004; Fang and Ramasamy 2015; Lau and Botella 2017), and UAV[3]-assisted crop surveillance (Zarei 2017).

However, the majority of these methodologies are not employed due to a lack of access to the infrastructure and funds needed to carry them out, or because they could cause harm or be ineffective. Pesticides have been shown to have negative impacts on human health and can damage the surrounding ecology (Nicolopoulou-Stamati et al. 2016; Borrelli et al. 2018) while pathogens can rapidly evolve to evade resistance conferred on disease-resistant crops (Barrett et al. 2009; Pagán and García-Arenal 2018; Andersen et al. 2018). More traditional diagnostic techniques such as visual inspections during cultivation or harvesting stages are most often used. This is where changes in physiological features of roots, stems or leaves that are typically associated with different disease types are observed by skilled field workers and treatment or removal of the infected crops is performed accordingly (Wang et al. 2012; Nidhis et al. 2019; Mello Prado 2021).

Section 2.1 has outlined the aspects relating to plant disease development and detection. Section 2.2 will discuss the importance of cassava as a staple crop in developing nations and the factors affecting its production potential.

2.2 Cassava (*Manihot esculenta* Crantz)

2.2.1 Agricultural origins and geography

The origins of cassava remained unclear until the discovery of *Manihot esculenta* ssp. *flabellifolia*, or Pohl, in the Amazon basin which is the closest relative to modern-day cultivated cassava (Rogers and Appan 1973; Hillocks, Thresh, and Bellotti 2002; Olsen 2004; Léotard et al. 2009). Cassava is postulated to have been grown as a subsistence crop from about 8000 years ago in southwest Brazil and about 7500 years ago in northwest Colombia (Dickau, Ranere, and Cooke 2007; Gillman and Erenler 2009). It was introduced by the Portuguese to West Africa in the 16th century and began slowly gaining popularity across Sub-Saharan Africa while also being introduced to other nations such as Indonesia, the Philippines and Sri Lanka along newly established trade routes (Jones 1959; Lötschert and Beese 1983). In the present day, it is cultivated in nations as far as Japan or South Pacific islands but is mainly grown in nations of Sub-Saharan Africa, Southeast Asia and South America.

With the information learned in this subsection about where cassava is popularly grown, the following subsection will outline its functional importance in these areas.

[3]Unmanned aerial vehicle

2.2.2 Utilisation

Cassava's popularity as a crop is mainly due to its root tubers which form a staple source of calories in certain developing nations (Cock 1982; Burns et al. 2010; Chisenga et al. 2019). Additional benefits of the crop include its nutrient-dense leaves which can be used as a vegetable and which also have medicinal benefits. Examples of how it is used include: the production of grained-based foods and animal feed (Gleadow et al. 2009; Ceballos et al. 2011; Bechoff et al. 2017; Mtunguja et al. 2019; Alamu et al. 2019); as a raw material in manufacturing (Tonukari et al. 2015; Karlström et al. 2016; Vasconcelos et al. 2016; Waisundara 2018); or in the production of bioethanol fuel (Nguyen, Gheewala, and Garivait 2007; Jansson et al. 2009). In many developing nations, it is seen as a famine-preventative crop due to its ability to withstand and produce high yields in harsh climates when the yield of other staple crops is low (Burns et al. 2010; Mbanjo et al. 2021).

With the above information about how cassava is used, the following subsection will discuss its growing importance in developing nations across the globe and the factors threatening its production.

2.2.3 Growth in developing nations & factors affecting production

Cassava is the second major source of carbohydrates in the world after maize, and the third major source of carbohydrates in the tropics after maize and rice (FAOSTAT 2020). World cassava production has increased from approximately 287.19 million tonnes in 2015 to approximately 302.66 milion tonnes in 2020, signifying an increase in production volume of 5.39%. Africa contributed the most to 2020 production with 193.63 million tonnes (63.97%), followed by Asia with 81.87 million tonnes (27.05%) and the Americas with 26.92 million tonnes (8.89%) (FAOSTAT 2020). Despite the fact that Africa accounts for more than half of cassava production worldwide and that Sub-Saharan Africa is estimated to account for 62% of worldwide cassava production by 2025 (Bennett 2015; FAOSTAT 2020), it produces the lowest yield[4] which indicates that farmland is not being used efficiently. Asia produces the highest average yield of the three areas (218 889 hg/ha), followed by the Americas (133 245 hg/ha), Oceania (117 265 hg/ha) and finally Africa (86 205 hg/ha).

The above statistics indicate that there are significant yield gaps especially in Africa (Fermont et al. 2009; Shackelford et al. 2018). Even though studies have shown that cassava is less affected by climate change than other staple crops and may actually benefit marginally from certain abiotic changes in the environment (Liu et al. 2008; Jarvis et al. 2012), other environmental factors such as soil quality, atmospheric CO2, poor farming practices and pathogen dissemination have shown to negatively affect cassava production (Gleadow et al. 2009; Graziosi et al. 2016; Kintché et al. 2017; Alene et al. 2018; Burdon and Zhan 2020) with disease development being the major limiting factor (Legg et al. 2011; Chikoti et al. 2019).

With the importance of cassava production for developing nations and the effect that disease development has on yield in mind, Subsection 2.2.4 will describe and outline the major diseases affecting yield.

[4] Measured in hectagrams per hectare of land (hg/ha)

2.2.4 Major diseases affecting yield

Of the major biotic stresses placed on cassava production, pathogen dissemination and concomitant disease development remain the major limiting factors of production. Epidemiological studies have shown that rates of transmission of cassava mosaic disease (CMD) and cassava brown streak disease (CBSD) are associated with whitefly *Bemisia tabaci* population levels (Legg and Ogwal 1998; Sseruwagi et al. 2004; Legg et al. 2011; Legg et al. 2015; Fondong 2017; MacFadyen et al. 2018) and that human-mediated dissemination of infected cuttings and seeds is an additional primary spread of these diseases (Legg et al. 2011; Zinga et al. 2013; Legg et al. 2015; Mwatuni et al. 2015; Maruthi et al. 2017; MacFadyen et al. 2018). The major disease that afflicts areas of cultivation in Latin America is cassava frog skin disease (CFSD) while in Africa, major diseases include cassava mosaic disease (CMD) and cassava brown streak disease (CBSD) (Campo, Hyman, and Bellotti 2011; Njoroge et al. 2017; Kalyebi et al. 2018; Nwezeobi et al. 2020).

We will now consider the importance of cassava disease detection for communities that are dependent on it.

2.2.5 Ramifications of disease detection

Early detection of disease can prevent the detrimental effects of large-scale crop losses. The consequences of untimely detection of cassava disease spread can be seen most clearly in the various African CMD (Zhou et al. 1997; Pita et al. 2001; Neuenschwander et al. 2002; Legg and Fauquet 2004; Bigirimana et al. 2004; Gillman and Erenler 2009; Brown 2015) and CBSD (Hillocks et al. 2001; Maruthi et al. 2017; Alicai et al. 2019) epidemics in previous decades or the CFSD epidemic in Latin America (Calvert, Cuervo, and Lozano 2012; Legg et al. 2015), all of which caused severe food insecurity in certain regions. Losses due to CMD have been particularly devastating as was seen during the Ugandan epidemic where 60 million dollars in cassava production value was lost annually (Legg and Fauquet 2004).

Due to the threat that CMD and CBSD continue to have on food security in African developing nations, which account for the majority of cassava production (as shown in Subsection 2.2.3), surveillance and management of disease transmission networks is vital. Recent occurrences of disease development in locations like Angola, Cameroon and Ghana are of concern to many local and neighbouring African cassava-producing countries (Torkpo et al. 2021; Doungous et al. 2022).

In addition to food and monetary losses, evidence from surveillance data suggests that the various CMD epidemics have led to reductions in cassava genetic diversity in the African nations they have occurred in by selecting for more CMD-resistant plant varieties (Wolfe et al. 2016; Ferguson et al. 2019). High plant genetic diversity mitigates the effects of disease on crop yield (Govindaraj, Vetriventhan, and Srinivasan 2015) and therefore continuous reductions in genetic diversity present a growing challenge for African cassava farmers, especially given that resistance to infection can be transient or may only be disease-specific (Legg et al. 2001; Thresh and Cooter 2005). CMD-resistant cassava cultivars can be used to mitigate disease development (Asare et al. 2014; Fondong 2017; Houngue et al. 2019) with some varieties beginning to be used in Uganda (Tumwegamire et al. 2018), while CBSD-resistant varieties that have been attempted only confer partial protection (Tomlinson et al. 2018). However, as outlined in Subsections 2.1.2 and 2.1.5, the viruses that cause these diseases are able to rapidly mutate and evade the resistance conferred on disease-resistant varieties.

To prevent issues associated with cassava disease development from occurring, other surveillance methodologies could be utilised in conjunction with the protocols outlined above to reduce the chances of disease epidemics occurring. With cassava's significance in Africa and the importance of managing diseases outlined above, Section 2.3 will outline literature relating to image-based statistical learning methodologies for plant disease detection and their application to plant and cassava surveillance protocols.

2.3 Image classification

2.3.1 Emergence of convolutional neural networks (CNNs)

The initial proposal of the perceptron model by Rosenblatt (1960) presented an algorithm that could learn patterns present in data. Later studies done on the visual cortical cells of cats revealed the idea of a 'receptive field' that could detect differences in image feature shapes, inspiring the development of the 'neocognitron' in 1980 (Hubel and Wiesel 1962; Fukushima and Miyake 1982). The development of backpropagation, first introduced by Rumelhart, Hinton, and Williams (1986), provided the building blocks for the development of the first multilayer perceptron model (LeCun et al. 1989) and the subsequent development of LeNet-5 (LeCun et al. 1998), both of which were applied to handwritten digit recognition tasks.

AlexNet (Krizhevsky, Sutskever, and Hinton 2012) was a significant development in deep learning, showing that models could be trained on graphics processing units (GPUs), allowing denser networks to be trained which significantly improved the feature extraction capabilities of CNNs, and allowed for models to be trained on larger and more complex datasets (Hinton, Osindero, and Teh 2006; Glorot and Bengio 2010; Zeiler and Fergus 2014). Other models such as ZFNet (Zeiler and Fergus 2014), VGGNet (Simonyan and Zisserman 2014), GoogLeNet (Szegedy et al. 2016; Szegedy et al. 2017), ResNet (He et al. 2015), SqueezeNet (Iandola et al. 2016), DenseNet (Huang et al. 2017) and MobileNet (Howard et al. 2017; Sandler et al. 2018) were later proposed based on the architectural developments provided by AlexNet, with some of these transfer learning models showing improvements in computational resource cost (Szegedy et al. 2017) or representational learning (Huang et al. 2017).

CNNs have be applied to a diverse range of tasks such as speech recognition (Ossama et al. 2014), text mining (Liang et al. 2017) and audio signal processing (Khan and Kwon 2019), but have mostly been applied to image-based tasks such as facial recognition (Li et al. 2018), object detection (Long, Shelhamer, and Darrell 2015; Pathak, Pandey, and Rautaray 2018), image segmentation (Ren et al. 2015; Ronneberger, Fischer, and Brox 2015; Lu et al. 2019; Yeom et al. 2021), or image classification (LeCun et al. 1989; Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2017; Huang et al. 2017; Sandler et al. 2018) where it is regarded as the standard approach to solving computer vision problems.

With the importance of CNNs for image-based tasks outlined in the above subsection, Subsection 2.3.2 will outline more specifically the application of deep learning to plant leaf disease detection.

2.3.2 Deep learning for plant leaf disease detection

As with other image-based tasks, deep learning approaches are most often used for plant leaf disease detection as they allow for the extraction of subtle spatial relationships between

image features so that disease class can be distinguished (Liu and Wang 2021). Deep transfer learning has been applied to plant leaf disease detection of inter alia corn (Ren, Zhang, and Wang 2019), tomatoes, grapes, peach and bell peppers (Mojjada et al. 2020), cucumber (Fujita et al. 2017), apple trees (Wang, Sun, and Wang 2017), cotton (Jenifa, Ramalakshmi, and Ramachandran 2019), pine (Mojjada et al. 2020), and cassava (Ramcharan et al. 2017; Sambasivam and Opiyo 2021).

Table 2.1 summarises previous literature of deep learning applied to cassava leaf datasets other than the one used in this analysis, whilst Table 2.2 summarises deep learning model performance on the same dataset used in this research project:

Author	CNN architecture	Test Accuracy	Test Weighted F1-score
Ramcharan et al. (2017)	InceptionV3	85.83%	-
Sambasivam and Opiyo (2021)	CNN + SMOTE	93%	-
Maryum, Akram, and Salam (2021)	EfficientNetB4	81%	-
Maryum, Akram, and Salam (2021)	EfficientNetB4 (U-net segmentation)	89.09%	-
Ravi, Acharya, and Pham (2022)	Attention-based EfficientNet	-	0.87
Zhuang (2021)	Vision Transformer (ViT) model	90.02%	-

TABLE 2.1: Summary of deep learning performance on other cassava datasets

Author	CNN architecture	Test Accuracy	Test Weighted F1-score
Dhivyaa, Kandasamy, and Rajendran (2022)	CNN + DCRDB + MLFD + Bi-LSTM	96.40%	0.945
Enkvetchakul and Surinta (2021)	NasNetMobile	84.51%	-
Enkvetchakul and Surinta (2021)	MobileNetV2	83.62%	-
Abayomi-Alli et al. (2021)	MobileNetV2	97.70%	0.9676
Mwebaze et al. (2019)	Resnet50 + se_resnext50 + se_resnext101	93.77%	-

TABLE 2.2: Summary of deep learning performance on the cassava dataset used in this analysis

To ensure fair comparison with previous work on cassava leaf disease detection in Table 2.2, it is important to note that Enkvetchakul and Surinta (2021) and Abayomi-Alli et al. (2021) fit their models to subsets of the dataset used in this research project, limiting their comparison with results achieved in this research project since models were trained and tested differently.

This subsection has outlined the importance and application of deep learning to plant leaf disease detection tasks. The following subsection will address literature relating to tree-based classification methods as an alternative to deep learning for image-based tasks.

2.3.3 Tree-based classification methods

The inception of tree-based statistical learning methods can be traced back to the development of Automatic Interaction Detector (AID) in the 1960s (Morgan and Sonquist 1963), which inspired the development of THeta Automatic Interaction Detection (THAID) (Messenger and Mandell 1972) and CHi-squared Automatic Interaction Detector (CHAID)(Kass 1980) that differed from AID in how recursive partitioning occurred and allowed for the algorithm to be extended to classification tasks.

However, these methodologies suffered from issues related to overfitting, multicollinearity and computational complexity which ID3 (Quinlan 1986) and C4.5 (Quinlan 1993) aimed to remedy by using random training subsets during model fitting, while Fast and Accurate Classification Tree (FACT)(Loh and Vanichsetakul 1988) and Quick, Unbiased and Efficient

Statistical Tree (QUEST)(Loh and Shih 1997) aimed to solve these issues by performing partitioning based on variable importance.

However, FACT suffered from variable selection bias which Classification Rule with Unbiased Interaction Selection and Estimation (CRUISE)(Kim and Loh 2001), Conditional Inference Trees (CTREE) (Hothorn, Hornik, and Zeileis 2006), and Generalised, Unbiased, Interaction Detection and Estimation (GUIDE)(Loh 2009) aimed to solve by ranking variables using different ranking criteria than that used in Loh and Vanichsetakul (1988). Classification and regression trees (CARTs) and the subsequently developed tree-based ensemble learning approaches such as bagging (Breiman 1996; Breiman 2001) and boosting (Freund and Schapire 1997; Freund and Schapire 1999; Chen and Guestrin 2016) became the most successful and widely used of the tree-based methodologies.

In terms of image classification, tree-based ensemble methods have been applied to objection detection (Khan, Hanbury, and Stoettinger 2010; Baumann et al. 2013), remote sensing scene classification (Brodley and Friedl 1997; jun Du, Tan, and jun Su 2009; Jhonnerie et al. 2015; Holloway et al. 2019; Yang et al. 2020), X-ray image classification (Ko, Kim, and Nam 2011), neuroimaging classification (Sarica, Cerasa, and Quattrone 2017), animal fiber classification (Zhu, Duan, and Wu 2021), and plant leaf disease classification (Hall et al. 2015; Govardhan and Veena 2019; Wójtowicz et al. 2021).

CNN feature extraction has been coupled with tree-based ensemble methods in certain applications, allowing researchers to take advantage of the excellent feature extraction capabilities of CNNs and the fast and accurate classifications provided by tree-based classifiers (Hall et al. 2015; Thongsuwan et al. 2021; Zhu, Duan, and Wu 2021; Bui et al. 2021; Jiao, Hao, and Qin 2021; Huertas-Tato et al. 2022).

The following subsection will outline literature relating to a relatively new tree-based ensembling method called gcForest.

2.3.4 Multi-grained cascade forest (gcForest)

The multi-grained cascade forests (gcForests) algorithm was first introduced in the seminal paper by Zhou and Feng (2017) and presents a potential non-deep learning tree-based alternative to CNNs. This algorithm has a layer-by-layer processing structure where multiple tree-based ensemble learners produce outputs that form the inputs for the next layer of ensemble learners in the cascade (see Subsection 4.2.2 for the methodological description). In the work done by Zhou and Feng (2017), gcForest was benchmarked against a range of different tasks which included image-based classification, where gcForest showed competitive performance against deep learning.

For non-image based applications, gcForest has been applied to damage estimation of household appliances (Xia et al. 2017), industrial fault diagnosis (Hu et al. 2018; Liu et al. 2019a; Zhang et al. 2022), cancer subtype classification (Dong et al. 2019), transformer fault detection (Liu et al. 2021b), detecting defects in wheat flour production (Zheng et al. 2022), voltage stability measurement (Wang et al. 2021), and automobile maintenance (Chen et al. 2020).

In terms of image-based tasks, gcForest has been applied to facial recognition (Liu et al. 2019c; Guehairia et al. 2020; Shen, Liu, and Wu 2020), feature extraction (Bi, Xue, and Zhang 2020), cloud/snow recognition (Xia et al. 2018), land usage estimation (Weng et al. 2020), eye

fundus assessment and glaucoma classification (Liu et al. 2021a), clothing fabric classification (Han et al. 2018), UAV tracking (Liu and Yang 2019), hyperspectral image classification (Li et al. 2017; Yin et al. 2018; Liu et al. 2019b), maize leaf disease classification (Arora, Agrawal, and Sharma 2020), and remote sensing image classification (Boualleg, Farah, and Farah 2019; Ma et al. 2019; Nie et al. 2021).

Certain applications have coupled CNN feature extraction with a gcForest classifier for remote sensing scene classification (Boualleg, Farah, and Farah 2019) and bearing fault diagnosis (Zhang et al. 2022; Xie et al. 2022), all of which showed competitive or better performance compared to standard deep learning approaches. Given that the CNN-gcForest model, developed by Boualleg, Farah, and Farah (2019), outperformed deep transfer learning and that remote sensing image classification requires granular features in an image to be classified, transferring this approach to distinguishing subtle disease differences in cassava disease classification tasks may work well.

Improvements to the gcForest algorithm have been made in subsequent publications to improve the drawbacks relating to memory and time constraints (Pang et al. 2018; Dong et al. 2019; Ma et al. 2022; Zhang et al. 2022). One such methodology, called multi-grained scanning with confidence screening (gcForestCS) showed similar or better performance compared to gcForest while decreasing both memory usage and training time (Pang et al. 2018). Therefore, gcForestCS is used rather than the original gcForest implementation in conjunction with CNN feature extraction in this research project.

2.4 Conclusion

This chapter has provided an in-depth review of the relevant literature relating to plant leaf disease detection in general, and cassava disease detection in particular, as well as image classification statistical learning methodologies that have been used to automate both of these tasks. In Chapter 3, we will discuss the different aspects of the data used in this research project.

Chapter 3

Data

This chapter will explore the dataset used in this research project and detail the modifications made to the training set in the data pipeline. Section 3.1 will provide a description of how the data were collected and labelled, after which details relating to each class label (Section 3.2), the class distribution (Section 3.3), and the limitations (Section 3.4) associated with the dataset are outlined. Finally, Section 3.5 will provide details relating to data preparation (Subsection 3.5.1) and how some of the limitations associated with the dataset were accounted for before model training (Subsections 3.5.2 and 3.5.3).

3.1 Description

The dataset used in this research project consists of images of cassava leaves taken with smartphone cameras that were collected from various Ugandan farmers by the Artificial Intelligence (AI) lab at Makerere University in Kampala, Uganda, and labelled by experts at the National Crops Resources Research Institute (NaCRRI)(Mwebaze and Owomugisha 2017). Although data cleaning was performed on the dataset, objects other than cassava leaves are present in the images so as to mimic image characteristics that would be present in-field (more information on this in Section 3.4). The images also vary in dimensionality and quality.

The dataset was used in the iCassava 2019 challenge, which aimed to build a light-weight model that could be trained on minimal cassava leaf image data and deployed on conventional smartphones for use by cassava farmers. The dataset is available for download via this link (Abadi et al. 2016).

3.2 Class labels

3.2.1 Cassava bacterial blight (CBB)

Cassava bacterial blight (CBB), caused by the bacterium *Xanthomonas phaseoli* pv. *manihotis*, is a major bacterial disease affecting cassava, causing yield reductions of up to 75% in some areas (López and Bernal 2012). Transmission of CBB on a single farm is typically either from the impact of raindrops on infected plants or poor phytosanitation practices, while transmission between farms is typically via transport of infected cuttings (McCallum, Anjanappa, and Gruissem 2017).

CBB disease development is characterised by the development of brown spotting on leaf tissue (see Figure 3.1) which can lead to necrosis and wilting of plant tissue in stems and leaves in advanced disease progression (McCallum, Anjanappa, and Gruissem 2017; Zárate-Chaves et al. 2021).

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.1: Example images of cassava plants with CBB.

3.2.2 Cassava brown streak disease (CBSD)

Cassava brown streak disease (CBSD), which was first identified in eastern Africa during the 1930s, is caused by two viral species: *Cassava brown streak virus* (CBSV) and *Ugandan cassava brown streak virus* (UCBSV), both of which are members of the family *Polyviridae* and genus *Ipomovirus* (Storey 1936; Winter et al. 2010; Alicai et al. 2016). Transmission of these viral species over short distances have been shown to be correlated with numbers of the whitefly vector *Bemisia tabaci* in experiments conducted in Tanzania (Maruthi et al. 2005), while it is transported over long distances via infected harvested plant material (Patil et al. 2015).

The symptoms associated with this disease include discolouration of the leaves that can lead to brown streaks on the stems in severe disease progression and rot of the root tubers (Figure 3.2), making them inedible and unmarketable for farmers (McCallum, Anjanappa, and Gruissem 2017; Ano et al. 2021).

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.2: Example images of cassava plants with CBSD

3.2.3 Cassava green mite (CGM)

Cassava green mite (CGM) is caused by *Mononychellus tanajoa* which was unintentionally transported to Africa from South America in 1971 and has since spread through central and eastern Africa (Thresh, Fargette, and Otim-Nape 2016). The presentation of this disease includes yellow/white mosaic-patterned spots on leaves, and changes in leaf morphology (Figure 3.3)(Thresh, Fargette, and Otim-Nape 2016; Mwebaze and Owomugisha 2017; McCallum, Anjanappa, and Gruissem 2017).

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.3: Example images of cassava plants with CGM

3.2.4 Cassava mosaic disease (CMD)

CMD was first reported in Tanzania in 1894 and is caused by infection with cassava mosaic begomoviruses (CMBs) which include *African cassava mosaic virus* (ACMV), *East African cassava mosaic virus* (EACMV), *South African cassava mosaic virus* (SACMV) and *Indian cassava mosaic virus* (ICMV) (Hillocks, Thresh, and Hillocks 2000; Thresh, Fargette, and Otim-Nape 2016). It is the major contributor to cassava production losses, especially in Sub-Saharan Africa (Brown 2015). The presentation of this disease includes changes in leaf morphology and white/yellow mosaic leaf discolouration (Figure 3.4).

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.4: Example images of cassava plants with CMD

3.2.5 Healthy

The final class label represents images of healthy cassava leaves (Figure 3.5) where there is an absence of any of the above symptoms indicative of disease. The main signs of healthy cassava development include green leaves with no discolouration, absence of discoloured spotting on leaves, absence of streaks developing on stems/leaves, and correct leaf maturation.

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.5: Example images of healthy cassava plants

3.3 Class distribution

Figure 3.6 shows the class distribution of the full dataset (9430 images) where the CMD and CBSD class labels are disproportionately represented in the dataset. Class imbalance can lead to a model becoming biased toward classifying an unknown observation as a majority class label and therefore addressing this imbalance can improve model generalisability.

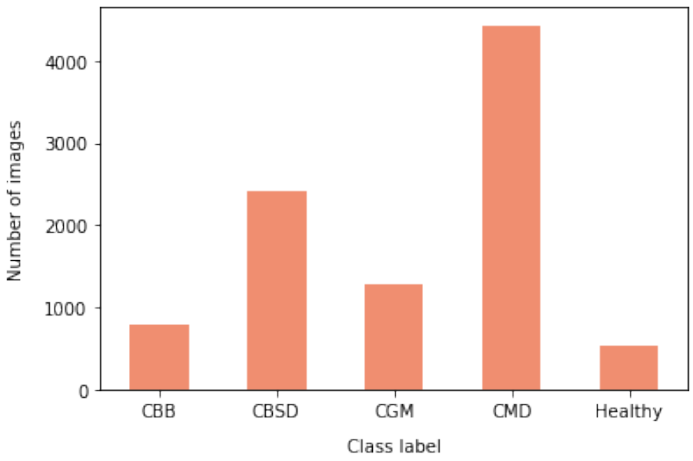


FIGURE 3.6: Illustration of class distribution of full dataset

To best account for class imbalance, a reproducible stratified split into training (5656 images), validation (1885 images) and test (1889 images) sets was performed, maintaining class proportions in all splits (Figure 3.7). Weighted F1-score (Subsection 4.3.1), which is typically used for imbalanced datasets, was used to measure performance of models trained on imbalanced data. Data augmentation of minority classes and downsampling of majority classes was also applied to the training set to produce perfect class balance, allowing the effect of changing class balance to be assessed.

3.4 Limitations

The dataset used in this research project is not without its limitations. The most obvious limitation is that each image is labelled with a single disease label, although a plant can be infected with multiple diseases (called co-infection) which can lead to a mixed disease presentation that can be difficult for a model to learn. Other limitations include: the severity of

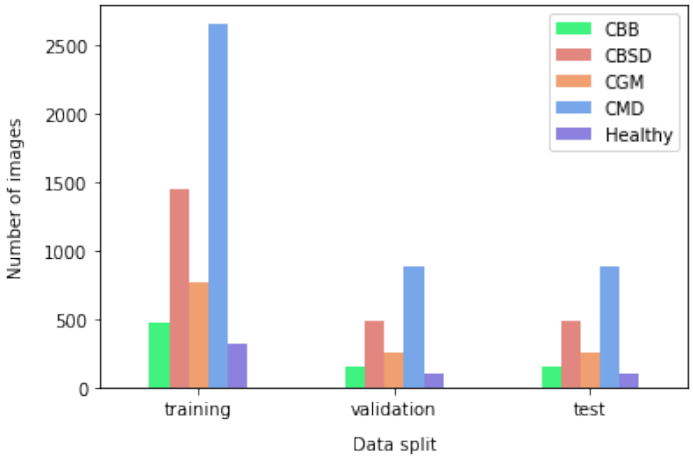


FIGURE 3.7: Illustration of class distribution across the various data splits

disease is not included in class predictions; images were taken with varying backgrounds and at different times of day; there are a number of images that contained anomalous features (hands, feet, boots, etc., see Figure 3.8) and/or had poor focus on the leaves (Figure 3.11); and the dataset is only representative of cassava disease in Uganda.

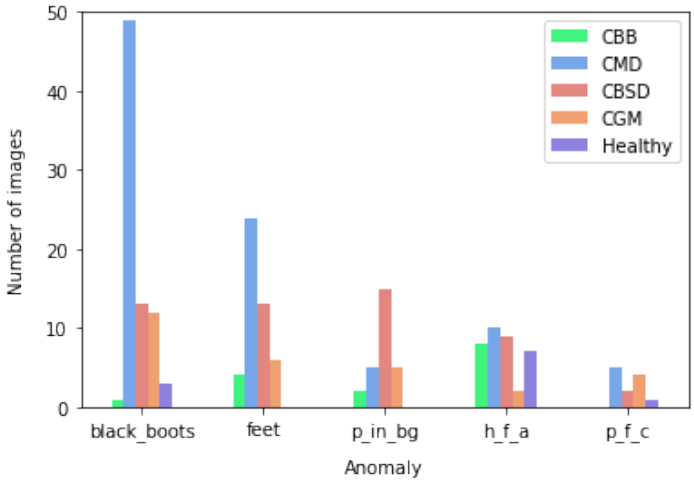
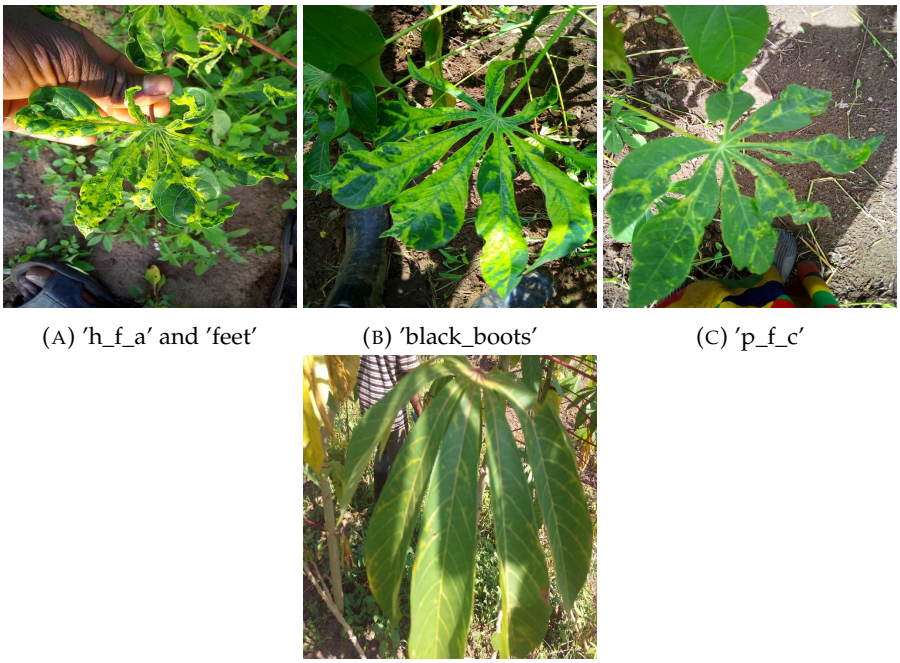


FIGURE 3.8: Illustration of 5 most occurring anomalies in training set

There were 21 anomalies identified in the training set; the frequency of the top 5 most occurring ones are displayed in Figure 3.8. These 5 most occuring anomalies were black gumboots ('black_boots'); human feet ('feet'); farmworkers in the background ('p_in_bg'); hands, fingers or arms ('h_f_a'); and patterned fabric clothing ('p_f_c'). Examples of these anomalies are shown in Figure 3.9. Given that the number of images with anomalies present per class is low relative to the number of training images in each class, training images with these anomalies were not removed but will be accounted for during interpretation in Chapter 5.

Images with poor focus/exposure on the leaves (Figure 3.10) lower the quality of the training data. To account for this, training images with poor focus will be removed during curation and models retrained to assess the effect this has on model generalisability. The number of training images with poor focus/exposure split by class is shown in Figure 3.11.



(A) 'h_f_a' and 'feet' (B) 'black_boots' (C) 'p_f_c'

(D) 'p_in_bg'

FIGURE 3.9: Example training images containing anomalies

(A) Example 1 (B) Example 2 (C) Example 3

FIGURE 3.10: Example training images with poor focus/exposure on leaves

3.5 Data modification

3.5.1 Image pre-processing

Before model training and testing, images were resized to have a shape of (224, 224, 3) since the input layer for the DenseNet201 and MobileNetV2 deep transfer learning models require inputs to adhere to this dimensionality. Pixel values were also normalised to the range [0,1] by dividing by 255 (the maximum value a pixel can have) to reduce the bias when the spatial relationships between image features are being learnt and to allow for quicker convergence during deep learning model training.

3.5.2 Training set curation

Images with poor focus on leaves (Figure 3.11) were removed during curation to assess whether the quality of the training data affects model performance. This resulted in 287 images being removed, leaving a remainder of 5369 training images with similar class balance to the original training set.

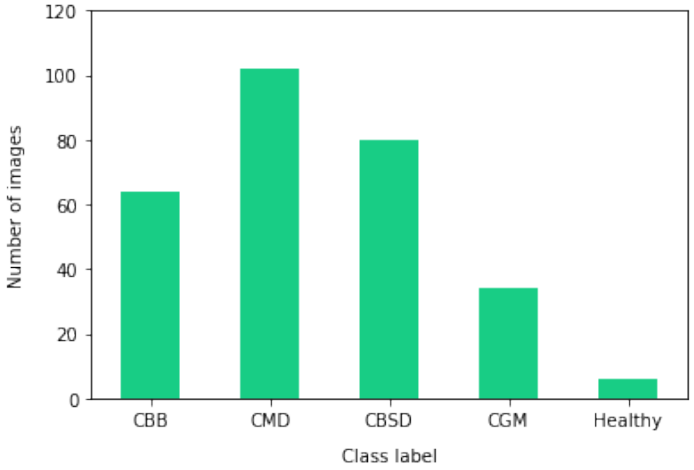


FIGURE 3.11: Illustration of number of training images with poor focus per class

3.5.3 Data augmentation & downsampling

Data augmentation is applied to datasets to upsample training data. This typically allows for more effective training of deep learning models by reducing the effects of class imbalance in classification tasks. There are different augmentation types that can be applied to image data, with popular types being flips, rotations, blurring, zooming, and shifting. Augmentation can either be applied to training inputs during the training process, referred to as on-line data augmentation, or can be applied before training occurs (off-line data augmentation).

In this research project, vertical and/or horizontal flips were applied to upsample training images of the 3 minority classes ('CGM', 'CBB', 'Healthy') which were then downsampled to 1200 training images per class. It is important to note that during downsampling, none of the original images were removed. Offline data augmentation was used to ensure that the same balanced training set was used across all model combinations. Majority classes ('CMD' and 'CBSD') were downsampled to 1200 images per class, such that the number of images in the balanced training set equated to 6000 images (1200 images per class).

The next chapter will outline the methodology used to perform image classification on the dataset discussed in this chapter.

Chapter 4

Methodology

As discussed in Chapter 1, this research project will compare the application of traditional deep learning approaches to CNN-gcForestCS for cassava leaf disease detection by testing different model combinations (Figure 4.1 and Tables 4.1 & 4.2). Section 4.1 will provide an overview of the different model combinations tested while Section 4.2 will outline the methodologies behind each approach.

4.1 Data Pipeline

Figure 4.1 illustrates the data pipeline followed when forming the different model combinations performed in this analysis. There are 38 model combinations in total – 30 base model combinations and 8 model combinations based on the best CNN-gcForestCS model. The latter includes 2 models investigating the effect of training a regular deep learning model on the curated training set. Solid lines connecting components indicate the base model combinations – also summarised in Table 4.1 – while dashed lines indicate that the component was only applied on the best CNN-gcForestCS model combination selected from the base model combinations (Table 4.2), again with the exception of the model combinations where training set curation was performed before deep learning.

To account for class imbalance in the training set and to determine the effect this has on model performance, augmentation and downsampling of classes was performed to produce a total of 6000 training images (for more details on the augmentations applied, refer to Subsection 3.5.3). At the next level in the pipeline, CNN feature extraction was optionally applied to the resized input images ($224 \times 224 \times 3$) with either pre-trained weights from training on the ImageNet[1] dataset, or weights that had been fine-tuned on the training set during the training of deep learning model combinations. Different feature map sizes (28×28, 14×14 and 7×7) were extracted from 3 candidate layers in the convolutional network as inputs with these dimensions showed promising deep forest model performance in previous work (Pang et al. 2018; Boualleg, Farah, and Farah 2019). Finally, either a gcForestCS classifier or a fully-connected network (FCN) were trained for final prediction.

Further investigation of the best CNN-gcForestCS model combinations for each CNN backbone was done where the effect of adding a pooling step after multi-grained scanning (MGS) and training models based on a curated version of the training set was assessed (for more information on training set curation, see Subsection 3.5.2).

[1]The ImageNet dataset contains the largest number of images for a variety of different applications – 14 197 122 annotated images (Krizhevsky, Sutskever, and Hinton 2012).

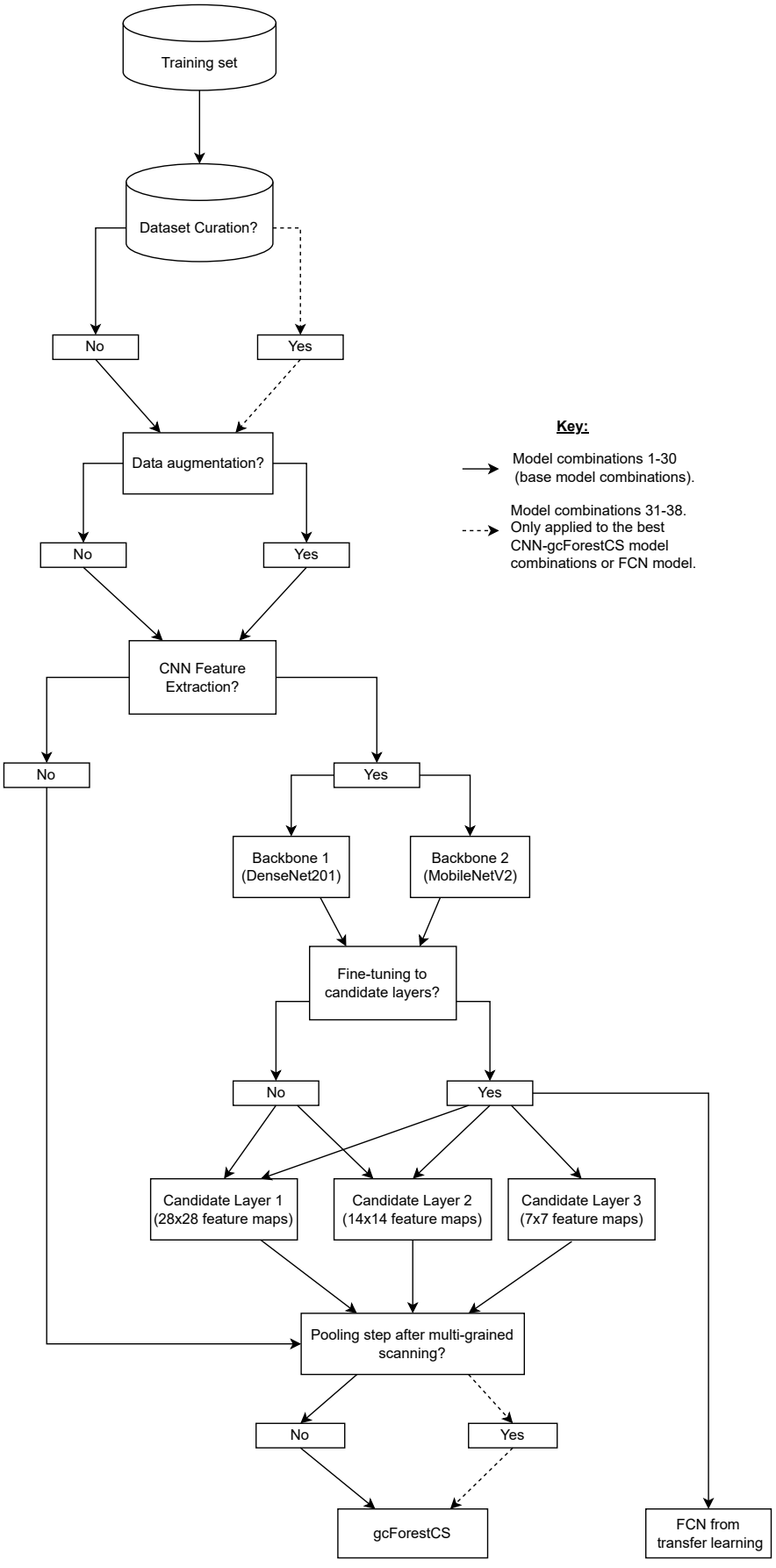


FIGURE 4.1: Illustration of different model combination components in data pipeline

Model combination	Training set curation?	Data augmentation?	MGS Pooling?	CNN Backbone	Fine-tuning?	Feature extraction layer	Classifier
1	No	None	None	None	None	None	gcForestCS
2	No	Yes	None	None	None	None	gcForestCS
3	No	None	None	DenseNet201	None	Candidate Layer 1	gcForestCS
4	No	None	None	DenseNet201	None	Candidate Layer 2	gcForestCS
5	No	None	None	DenseNet201	None	Candidate Layer 3	gcForestCS
6	No	Yes	None	DenseNet201	None	Candidate Layer 1	gcForestCS
7	No	Yes	None	DenseNet201	None	Candidate Layer 2	gcForestCS
8	No	Yes	None	DenseNet201	None	Candidate Layer 3	gcForestCS
9	No	None	None	MobileNetV2	None	Candidate Layer 1	gcForestCS
10	No	None	None	MobileNetV2	None	Candidate Layer 2	gcForestCS
11	No	None	None	MobileNetV2	None	Candidate Layer 3	gcForestCS
12	No	Yes	None	MobileNetV2	None	Candidate Layer 1	gcForestCS
13	No	Yes	None	MobileNetV2	None	Candidate Layer 2	gcForestCS
14	No	Yes	None	MobileNetV2	None	Candidate Layer 3	gcForestCS
15	No	None	None	DenseNet201	Yes	None	FCN
16	No	Yes	None	DenseNet201	Yes	None	FCN
17	No	None	None	MobileNetV2	Yes	None	FCN
18	No	Yes	None	MobileNetV2	Yes	None	FCN
19	No	None	None	DenseNet201	Yes	Candidate Layer 1	gcForestCS
20	No	None	None	DenseNet201	Yes	Candidate Layer 2	gcForestCS
21	No	None	None	DenseNet201	Yes	Candidate Layer 3	gcForestCS
22	No	Yes	None	DenseNet201	Yes	Candidate Layer 1	gcForestCS
23	No	Yes	None	DenseNet201	Yes	Candidate Layer 2	gcForestCS
24	No	Yes	None	DenseNet201	Yes	Candidate Layer 3	gcForestCS
25	No	None	None	MobileNetV2	Yes	Candidate Layer 1	gcForestCS
26	No	None	None	MobileNetV2	Yes	Candidate Layer 2	gcForestCS
27	No	None	None	MobileNetV2	Yes	Candidate Layer 3	gcForestCS
28	No	Yes	None	MobileNetV2	Yes	Candidate Layer 1	gcForestCS
29	No	Yes	None	MobileNetV2	Yes	Candidate Layer 2	gcForestCS
30	No	Yes	None	MobileNetV2	Yes	Candidate Layer 3	gcForestCS

TABLE 4.1: Tabular depiction of base model combinations

Model combination	Training set curation?	Data augmentation?	MGS Pooling?	CNN Backbone?	Fine-tuning?	Feature extraction layer	Classifier
31	No	No	Yes	DenseNet201	Yes	Candidate Layer 3	gcForestCS
32	No	No	Yes	MobileNetV2	Yes	Candidate Layer 3	gcForestCS
33	Yes	No	No	DenseNet201	Yes	-	FCN
34	Yes	No	No	MobileNetV2	Yes	-	FCN
35	Yes	No	No	DenseNet201	Yes	Candidate Layer 3	gcForestCS
36	Yes	No	Yes	DenseNet201	Yes	Candidate Layer 3	gcForestCS
37	Yes	No	No	MobileNetV2	Yes	Candidate Layer 3	gcForestCS
38	Yes	No	Yes	MobileNetV2	Yes	Candidate Layer 3	gcForestCS

TABLE 4.2: Tabular depiction of model combinations based on best CNN-gcForestCS models

In light of the details of the different model combinations tested in this analysis outlined above, the following section will discuss the methodologies underpinning the different components of the data pipeline.

4.2 Model architecture

This section will begin by discussing convolutional neural networks (CNNs) before outlining the gcForestCS methodology. Given that deep learning is the more established methodology, Subsection 4.2.1 will only briefly describe the key components of the deep learning methodology as applied to computer vision tasks, while Subsection 4.2.2 will provide a more in-depth focus on the mechanics of the gcForestCS implementation.

4.2.1 Convolutional neural networks (CNNs)

4.2.1.1 Basic structure

Deep learning models are composed of layers of nodes connected by weights that form a multi-layered network (LeCun et al. 1989; Krizhevsky, Sutskever, and Hinton 2012). Incoming data enters the network via an input layer where it is progressively processed by intermediary 'hidden' layers and finally used to produce predictions via an output layer (Rumelhart, Hinton, and Williams 1986; LeCun, Bengio, and Hinton 2015). Weight vectors

and bias terms are adjusted during training to better fit the data distribution of the training data and therefore make up the trainable parameters of neural networks (LeCun et al. 1989; LeCun, Bengio, and Hinton 2015).

Convolutional neural networks (CNNs) are made up of two main modules: convolutional layers and fully-connected layers (LeCun et al. 1989; LeCun et al. 1998). Convolutional layers apply convolution to input image patches to allow higher-order spatial relationships between features to be distinguished via the formation of feature maps (LeCun et al. 1989) (for a brief explanation of the convolution process, see Subsubsection 4.2.1.2). The feature maps produced from the convolution module provide a re-representation of the original input images that can allow for more effective training of the fully-connected layers (Krizhevsky, Sutskever, and Hinton 2012; Zeiler and Fergus 2014; Srivastava et al. 2014). Pooling layers are typically applied after convolutional blocks to reduce the dimensionality of the feature maps (Simonyan and Zisserman 2014; Szegedy et al. 2017). Feature maps produced from the convolution module of CNNs are then propagated to the fully-connected network of CNNs which allow for classification. Fully-connected layers in CNNs can be prone to overfitting, hence the use of dropout layers where weights between nodes are randomly removed during training to reduce chances of co-adaptation during gradient descent and to improve model generalisation (Srivastava et al. 2014).

Using basic knowledge gained about the structure of CNNs above, the next subsubsection will briefly outline how feature maps are produced in the convolution module.

4.2.1.2 Convolution

The success of CNNs in image-based tasks can be owed partly to the advanced feature extraction capabilities of the convolutional layers (Hinton, Osindero, and Teh 2006; Glorot and Bengio 2010). Convolution operations are performed via sequential application of element-wise matrix multiplication of a kernel (or filter) of fixed size to sections of an input image matrix and can be seen as a 'sliding window' composed of weights that are convolved with sections of an input image they superimpose (Krizhevsky, Sutskever, and Hinton 2012; Zeiler and Fergus 2014; Gu et al. 2018). During the convolution step, the kernel moves along the image horizontally and/or vertically by a predefined number of pixels called the stride (Krizhevsky, Sutskever, and Hinton 2012; Yamashita et al. 2018; Alzubaidi et al. 2021). The dot-product of the kernel weights and the pixel values of the input image section the kernel is being applied to form the next element of the resulting matrix called a feature map (Yamashita et al. 2018; Alzubaidi et al. 2021). A convolution step is complete once a kernel has traversed an entire input matrix and transformed it into a feature map. After convolution steps, a network typically applies an activation function to the feature map followed by a pooling layer (Simonyan and Zisserman 2014; Szegedy et al. 2017; Alzubaidi et al. 2021).

Multiple rounds of convolution are typically applied to training images across several layers in a process called 'layer-by-layer processing' where spatial relationships between features are refined to help differentiate classes during classification (Yamashita et al. 2018; Gu et al. 2018). The feature maps produced from the chosen output layer of the convolutional module (in this work, these are termed **candidate layers**) form the inputs for either a fully-connected network (deep learning) or a separate classifier (Ossama et al. 2014; Gu et al. 2018).

4.2.1.3 Backpropagation

During training for a supervised image-based classification task, training images are propagated through a network to produce classifications in a process called forward propagation (LeCun et al. 1989). During this process, training images are often grouped into mini-batches of fixed size which aim to minimise memory consumption whilst improving training times (LeCun et al. 1989; Gu et al. 2018). After each mini-batch is processed, predicted training image labels produced by the network are compared with the true image labels via a loss function (see Subsubsection 4.2.1.5), of which the resulting value is used to sequentially optimise the weights and biases of the network in a process termed mini-batch gradient descent (Hinton, Osindero, and Teh 2006; Gu et al. 2018; Alzubaidi et al. 2021). The process where the weights are updated from the output layer to the input layer is referred to as backpropagation (Rumelhart, Hinton, and Williams 1986; LeCun et al. 1989; Hecht-Nielsen 1992). The optimisation algorithm used to update weights and biases during backpropagation, called an optimiser, is explicitly defined before training with the SGD (stochastic gradient descent) (Bottou 2012) and Adam (Kingma and Ba 2014) optimisers being most commonly used. The learning rate controls the degree to which weights and biases are optimised during a backpropagation step and can either be static or adaptive based on the optimiser used (Gu et al. 2018).

Once all training image batches have been processed and used to update model weights (each iteration is termed an epoch), an explicit validation set – or validation fold during cross-validation – can be used to monitor model generalisability to reduce chances of overfitting (Srivastava et al. 2014). Training generally occurs across n epochs or until the validation loss function does not improve for a certain number of epochs (referred to as early stopping), indicating that a global or local minima of the loss function has been reached during gradient descent (Yao, Rosasco, and Caponnetto 2007; Prechelt 2012).

4.2.1.4 Activation functions

The majority of practical image classification tasks exhibit complex relationships between features that are not linearly separable (Krizhevsky, Sutskever, and Hinton 2012; LeCun, Bengio, and Hinton 2015). The CNN methodology accounts for this by applying non-linear transformations in the form of activation functions between convolution steps, promoting the learning of non-linear relationships between features (Ding, Qian, and Zhou 2018; Gu et al. 2018; Yamashita et al. 2018). Activation functions range in the type of transformations they apply, with the rectified linear unit (ReLU) activation function being most commonly used (Nair and Hinton 2010; LeCun, Bengio, and Hinton 2015). The softmax activation function is applied in the final layer of the fully-connected network to produce a probability distribution that allows for classification of an observation based on the maximum probability value produced (Yamashita et al. 2018)(for more information on the mechanics of the different types of activation functions available, refer to Ding, Qian, and Zhou (2018)).

4.2.1.5 Loss functions

Loss functions are used to determine the degree to which true and predicted training labels produced by a neural network differ during the training process. The resulting loss produced by the loss function is used to perform weight updates during backpropagation so that the model progressively captures variation in the training data, thereby reducing the loss. The validation loss is also typically assessed after each epoch to prevent model overfitting from occurring during training but is not used to perform weight updates (Yao,

Rosasco, and Caponnetto 2007; Prechelt 2012; Srivastava et al. 2014). For multi-class classification tasks, categorical cross entropy tends to be used to measure the difference in the probability distributions of the predicted and true labels (for an in-depth explanation of the theory underpinning categorical cross-entropy, refer to Zhang and Sabuncu (2018)).

4.2.1.6 Transfer learning

Before model training, weights and biases can either be randomly initialised or repurposed from a model pre-trained with data representative of the current application (Chen et al. 2020). Following the former path requires all weights and bias of a model to be trained from scratch for accurate model performance. Even though improvements in computer hardware and algorithm design have led to more efficient model training (LeCun, Bengio, and Hinton 2015), training from scratch can still be time- and computationally-intensive, especially when dense and/or deep neural networks are employed (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2017). Given that the depth of neural networks and training set size affect the extent to which complex non-linear relationships between features can be learned (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2017; Kim et al. 2018; Li, Nie, and Chao 2020), techniques that allow for efficient and effective training of these types of models are crucial.

Transfer learning provides a solution whereby pre-trained kernel weights and biases in the convolutional layers of some CNN model are repurposed for a similar task to the one it was originally trained on (Chen et al. 2020). The convolution module of deep transfer learning models can either be used as a feature extractor for a separate classifier, or a fully-connected network specific to the new task can be appended, after which the CNN model can be trained in a process called fine-tuning (Yamashita et al. 2018; Chen et al. 2020). With fine-tuning, early layers in the pre-trained convolutional module are 'frozen' (i.e. these weights are not optimised further during training) while later convolutional layers and an appended fully-connected network are 'unfrozen' (i.e. these weights are optimised for the new task during training) to allow for more specific variation in the training data of the new task to be captured during training (Yamashita et al. 2018; Chen et al. 2020). Models used in transfer learning each have their own combination and types of convolutional, pooling, and fully connected layers together with specific activation functions and optimisers, referred to as the model architecture. When these pre-trained models are fine-tuned to a new task, they are referred to as the backbone in the transfer learning process.

This research uses as backbone the two best-performing deep transfer learning models on the cassava leaf image dataset in question that allow for an input image shape of $224 \times 224 \times 3$ (Enkvetchakul and Surinta 2021; Huertas-Tato et al. 2022), namely DenseNet201 (Figure 4.2) and MobileNetV2 (Figure 4.3). Best practice for transfer learning is to use models trained on data representative of the problem at hand. However, there is a lack of pre-existing weights from reputable repositories relating to plant disease detection for the DenseNet201 and MobileNetV2 architectures, therefore the weights obtained from these models being trained on the ImageNet dataset were used. This allowed for basic low-level image features common across image classification tasks (such as edges and blobs) to be extracted during early convolution steps (Shin et al. 2016; Chen et al. 2020).

The above subsection has provided a primer on the technical details of how CNNs are able to model complex problems. The following subsection will focus on the mechanics of the different components of the gcForestCS methodology.

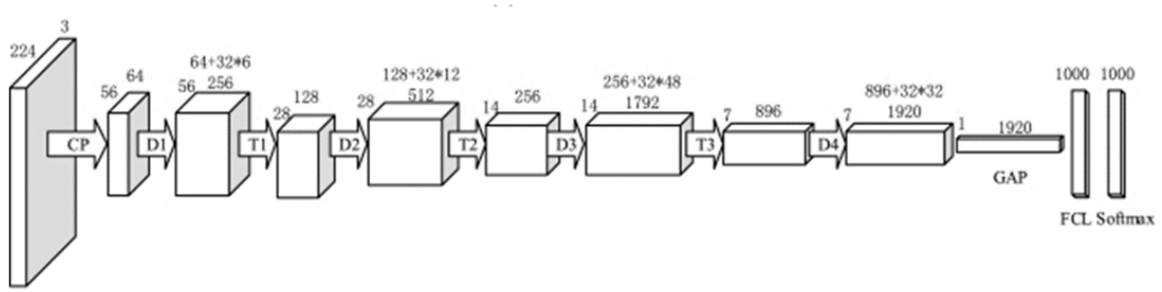


FIGURE 4.2: Illustration of DenseNet201 architecture as initially proposed by Huang et al. (2017)

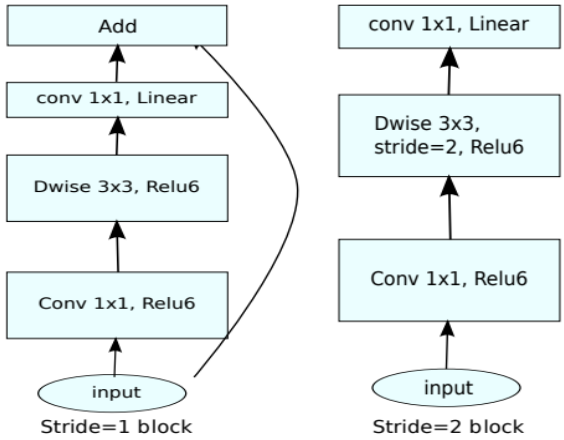


FIGURE 4.3: Illustration of MobileNetV2 architecture as proposed by Sandler et al. (2018)

4.2.2 gcForestCS

gcForest, or 'deep forest', was initially proposed by Zhou and Feng (2017) as a potential alternative to deep learning. This methodology employs an 'ensemble of ensembles' learning approach where multiple ensemble learners perform feature transformation of input data across multiple layers to allow for pertinent spatial relationships between features to be learned during image classification tasks.

The deep forest methodology is composed of two main modules: multi-grained scanning and cascade forests. The multi-grained scanning module performs feature re-representation (similar to the convolutional module in CNNs) while cascade forests perform classification via layer-by-layer processing (similar to fully-connected layers in CNNs).

This subsection will begin by briefly outlining the methodology behind decision trees and tree-based ensemble learning before providing a more in-depth explanation of the different modules of gcForest. Finally, the mechanics of confidence screening in gcForestCS will be discussed.

4.2.2.1 Tree-based ensemble learning

Tree-based methods form a popular branch of machine learning algorithms that can be used for both regression- and classification-based tasks. The most basic of the tree-based methodologies is a decision tree, where levels composed of nodes that are connected by branches

produce a model structure synonymous of a tree structure. All data initially start in a 'root node' and are progressively separated into child nodes by a process called recursive binary splitting (Breiman et al. 1984). Partitions are made based on splitting criteria, such as Gini Index or Information Entropy, which aim to minimise node impurity (i.e. the proportion of incorrect classifications) in child nodes produced by a split (Breiman et al. 1984; Rokach and Maimon 2005). Partitioning continues until either class homogeneity is achieved in a node or the threshold for hyperparameters controlling tree depth are reached, producing a set of leaf nodes which can be used for class prediction during classification tasks (Strobl, Malley, and Tutz 2009). In the majority of real-world applications, leaf nodes are not homogenous and a probability vector is produced which represents the proportions of each class in the node. During classification on testing data, an unseen observation traverses through the tree structure until it ends up in a leaf node where the class with the majority vote in the node is assigned to the observation (Breiman et al. 1984).

There are multiple drawbacks associated with using a single decision tree during model training: lack of robustness to noisy data, high chances of overfitting if hyperparameters such as tree depth are not controlled, and high memory and time usage constraints given that all data needs to be loaded during training (Rokach and Maimon 2005; Hothorn, Hornik, and Zeileis 2006; Song and Lu 2015). To combat these issues, an ensemble of multiple uncorrelated decision trees (or 'weak learners') can be trained with the idea being that the aggregation of information in the training set learned by each weak learner will produce a final model that can accurately reflect variation in the data (Breiman 1996; Freund and Schapire 1997; Freund and Schapire 1999; Breiman 2001; Lee et al. 2020). Results across the multiple weak learners are aggregated, with the final prediction for each observation being the majority class vote across all learners.

There is strong evidence to support the claim that ensemble learning improves upon the performance of single tree classifiers (Bauer et al. 1999; Dietterich 2000; Miao et al. 2011). The two main tree-based ensembling approaches are boosting (Freund and Schapire 1997; Freund and Schapire 1999; Chen and Guestrin 2016) and bagging (Breiman 1996), a latter of which works by training trees using random bootstrap samples of the training set. Random forests (Breiman 2001) follow the bagging principle and use random subsets of features to produce the best split during each partitioning step. Extremely-randomised forests (Geurts, Ernst, and Wehenkel 2006) build on the idea of random forests with the main differences being that partitions are done randomly based on a random subset of features, and each tree in the forest structure is grown based on the full training set.

The benefits associated with tree-based ensemble learning as compared to parametric methods are that fewer assumptions about the data need to be made, less hyperparameter tuning is required, and parallelisation of tree growth can be leveraged to produce high throughput during training when training observations have relatively few features. However, a major drawback of tree-based ensembling learning is the memory and time constraints when a model is fitted to high-dimensional datasets.

With the knowledge gained about tree-based ensemble learning above, the next subsubsection will build upon this understanding by covering the methodology underlying the multi-grained scanning module of gcForestCS.

4.2.2.2 Multi-grained scanning

Experiments by Zhou and Feng (2017) and Pang et al. (2018) have shown that the addition of multi-grained scanning before training cascade forests (see Subsubsection 4.2.2.3) improves their generalisation. Based on the concept of representation learning performed during convolution steps in deep learning, multi-grained scanning begins by splitting an input image with dimensions $(w \times h \times d)$ into 'instances' via the scanning of a sliding window of size (S_x, S_y) with strides Str_x along the x-axis, and Str_y along the y-axis. After an instance has been created, it is assigned the class label of the image it originated from. Equation 4.1 can be used to determine the number of instances produced from scanning:

$$N_{inst} = \left(\frac{w - S_x}{Str_x} + 1\right)\left(\frac{h - S_y}{Str_y} + 1\right) = (N_x)\,(N_y)\,. \tag{4.1}$$

After all instances for an image have been created, NF forests are trained based on these instances where it is recommended to perform k-fold cross validation to reduce overfitting. After training, NF probability vectors are produced for each instance representing the probability of the instance belonging to each class. For an input image, N_{inst} probability vectors in the form of a (N_x, N_y, C) probability matrix are produced from each of the NF forests, where C represents the number of classes in the probability vector. The probability matrices produced from each forest are reshaped into (C, N_x, N_y) probability matrices, flattened and concatenated to each other to produce a final transformed feature vector that is a re-representation of the original image and can be used to train the cascade forest classification module.

In the case where M sliding window sizes are used, M transformed feature vectors will be produced for an input image which will sequentially be used during training of 'meta-layers' in the cascade forest structure, as explained in the following subsubsection.

Figure 4.4 gives an illustrated example of this process where a sliding window of size 10×10 with stride 1 along both the x- and y-axes is applied to a raw 20×20 input image. Using Equation 4.1, we can determine that 121 instances will be produced after which these instances will be used to train both a random forest classifier (Forest A) and a completely-random forest classifier (Forest B). After training, 121 probability vectors with length 3 for a three-class classification task will be produced from each forest for each input image (yielding a column vector with length 363 for each forest) which are then concatenated to form a transformed feature vector of length 726.

In the case where a pooling step is applied (similar to pooling after convolution steps in CNNs mentioned in Subsubsection 4.2.1.2), a pooling sliding window of size (P_x, P_y) with stride P_{str} will scan the (N_x, N_y, C) probability matrix produced from each forest and return either an averaged probability vector (average pooling) or the probability vector with the maximum value (max pooling) of the image patch that is superimposed. This produces a new pooled probability matrix of size $\left(\left(\frac{N_x - P_x}{P_{str}} + 1\right), \left(\frac{N_y - P_y}{P_{str}} + 1\right), C\right)$ which follows the same process of reshaping, flattening and concatenation to form a transformed feature vector as described above.

The concept of multi-grained scanning has been outlined above. The following subsubsection will detail the cascade forest module of gcForest which is trained based on the outputs of multi-grained scanning.

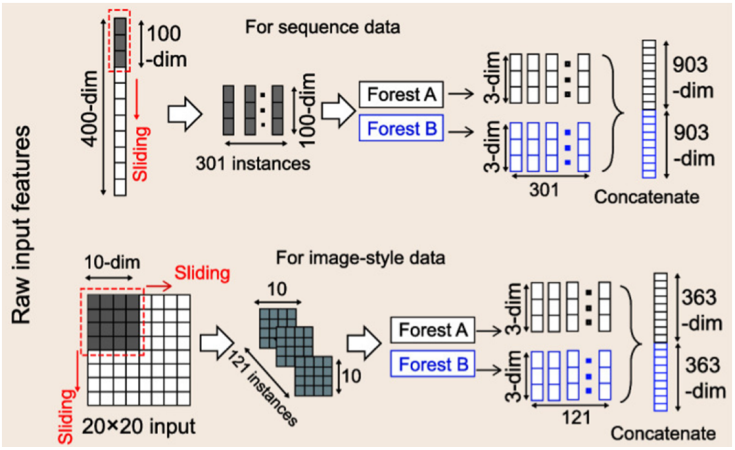


FIGURE 4.4: Illustration of multi-grained scanning as proposed by Zhou and Feng (2017)

4.2.2.3 Cascade forests

The layer-by-layer processing aspect of the gcForest methodology is realised in this module. In the initial layer of the cascade forest module, the transformed feature vectors produced for each input training image from multi-grained scanning forms the training inputs for the cascade forests. Outputs from the preceding level of the cascade forest structure are used to train NF forests in the current level, afterwhich NF probability vectors are produced for each transformed feature vector. These probability vectors are concatenated to each other and to the initial transformed feature vector produced from multi-grained scanning to form the input for the next level (Figure 4.5). Layers are progressively added until the validation accuracy during k-fold cross validation does not improve for a given number of early stopping rounds, or until a maximum number of layers have been added. The final layer produces a set of probability vectors from each of the NF forest classifiers which are then averaged and used to perform the final prediction.

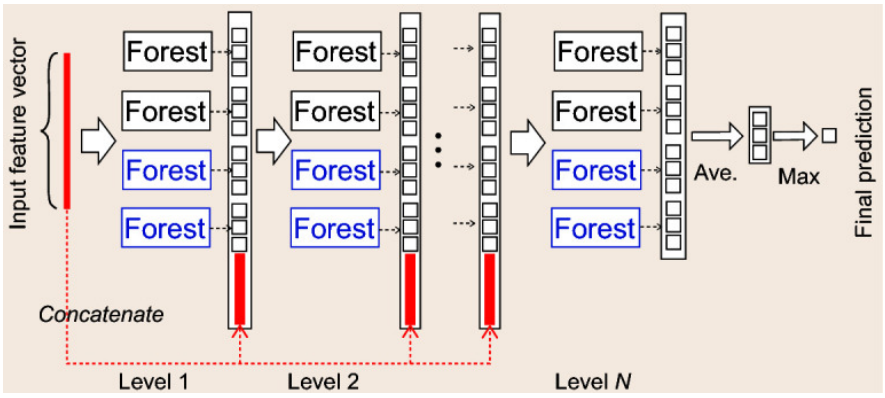


FIGURE 4.5: Illustration of cascade forest module as proposed by Zhou and Feng (2017)

In the case where M sliding window sizes are used during multi-grained scanning, a 'meta-layer' with M sub-levels is produced in the cascade forest structure where a similar process of concatenation of initial feature vectors as described above occurs. The major difference

is that the transformed feature vectors produced from each sliding window size in multi-grained scanning are sequentially concatenated to the probability vectors produced from the preceding sub-level of a meta-layer (Figure 4.6). The output from each sub-level and the next transformed feature vector from multi-grained scanning are then used as the input feature vector for the next sub-level of the meta-layer until the forests in all sub-levels have been trained.

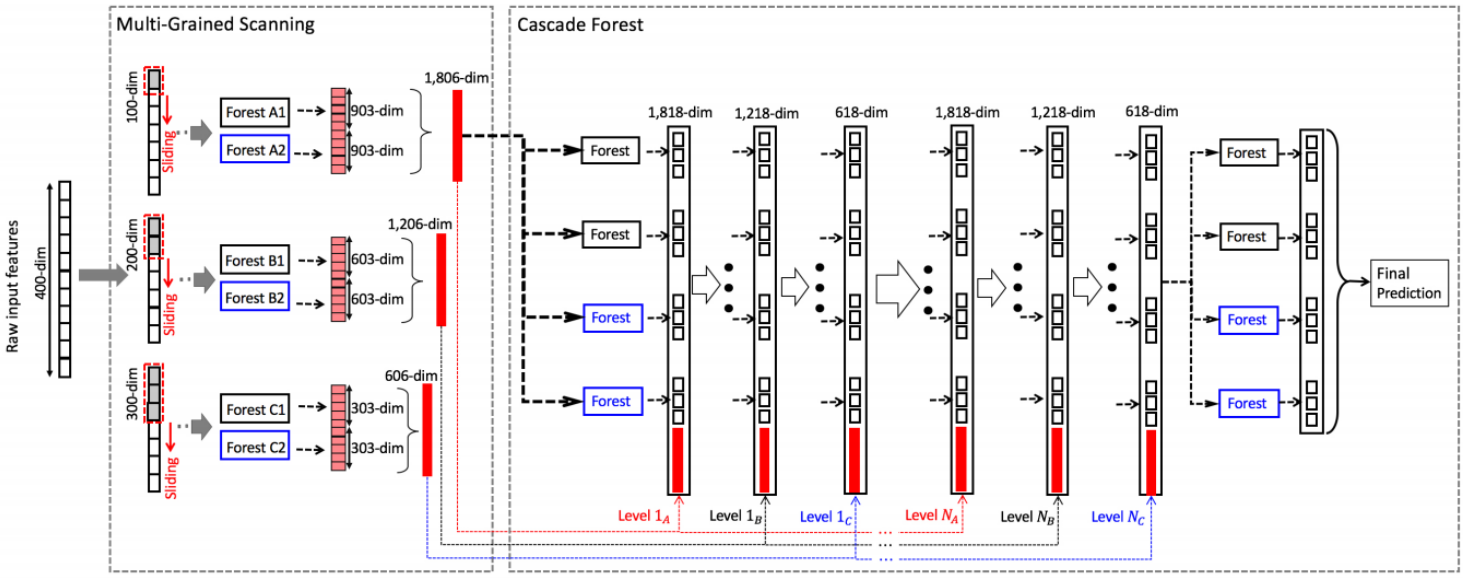


FIGURE 4.6: Illustration of overall gcForest implementation as proposed by Zhou and Feng (2017)

Provided with the underlying workings of the multi-grained scanning and cascade forest modules of gcForest, the next subsubsection will cover the addition of confidence screening to gcForest to complete the explanation of the gcForestCS algorithm.

4.2.2.4 Confidence screening

A major criticism of the original gcForest implementation proposed by Zhou and Feng (2017) is that all instances produced in multi-grained scanning are processed by all layers in the cascade forest structure regardless of whether they aid in class differentiation, leading to high memory consumption and training time. GcForestCS aims to resolve this through the addition of 'confidence screening', which entails random sampling of instances during multi-grained scanning and varying model complexity in the cascade forest structure (Pang et al. 2018).

The confidence screening mechanism makes use of a logic gate before probability vectors are concatenated into transformed feature vectors at the end of a level (Figure 4.7). The purpose of these logic gates is to determine whether an observation requires more learning in order to make an accurate class prediction and therefore needs to be propagated to the next level of the cascade forest structure. These logic gates assess whether the confidence value for an observation, which is the maximum value of the averaged probability vector, is greater than the calculated threshold value η_t. An observation i with confidence $c_i > \eta_t$ is said to have high confidence and is classified directly from the current level in the cascade forest

structure without being propagated to the following levels. By only propagating observations that require more learning through to the next level of the cascade forest structure, the memory consumption and training times have been shown to be significantly reduced while maintaining model performance (Pang et al. 2018).

$$\eta_t = \min \{c_k | L(x_1, ..., x_k) < a\epsilon_t, k \in [1, m]\} . \tag{4.2}$$

The threshold η_t at level t that determines whether an observation has high or low confidence is calculated by sorting all m training inputs from highest to lowest confidence, after-which Equation 4.2 is applied. $L(x_1, ..., x_k)$ represents the error rate for the k inputs with the highest prediction confidences while $a\epsilon_t$ represents a fractional value of the cross-validation error rate where $a \epsilon$ (0,1).

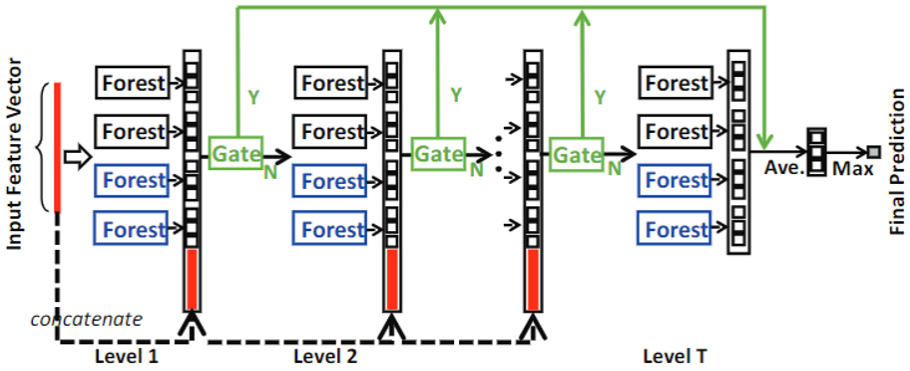


FIGURE 4.7: Illustration of confidence screening mechanism as proposed by Pang et al. (2018)

In addition to the confidence screening mechanism, gcForestCS also makes use of random instance sub-sampling for each training input at the beginning of multi-grained scanning, as well as varying model complexity as the number of levels in the cascade forest structure increase. Figure 4.8 depicts the process of instance sub-sampling which reduces the number of instances processed for each image, and the resulting size of the transformed feature vectors produced by multi-grained scanning and the cascade forests. Increasing model complexity in the cascade forest module to classify complex instances that consistently produce sub-threshold confidence values is employed because an observation that traverses through many levels of the cascade forest structure indicates that more model complexity (via more complex forests or more trees) is required to accurately capture variation present in these observations.

This section has provided an in-depth discussion of the CNN-gcForestCS methodology, with more focus on explaining the mechanics of gcForestCS. The following section will briefly describe how model performance will be assessed during validation and testing.

4.3 Performance metrics

For the purposes of this research, weighted F1-score and overall accuracy were taken into account when selecting the best hyperparameter combinations and assessing model performance. This subsection will briefly outline how each metric is calculated and what it measures.

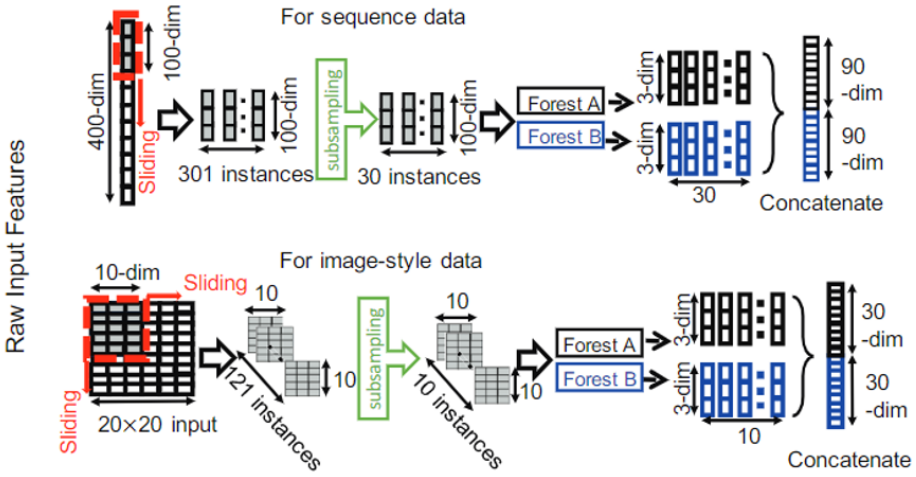


FIGURE 4.8: Illustration of instance subsampling in gcForestCS as proposed by Pang et al. (2018)

4.3.1 Weighted F1-score

F1-score (Equation 4.5) is a metric that produces the harmonic mean between the precision (Equation 4.3) and recall (Equation 4.4) when the weight parameter in the F-score is set to $\beta = 1$. F1-score forms an important metric for measuring model generalisability when trained on an imbalanced dataset because it takes into account both the precision and recall achieved by a model for each class. The score ranges between 0 and 1 where the higher the F1-score, the better the model generalises to unseen data. When calculating the weighted F1-score, the F1-score produced for each class is weighted based on the number of observations of that class in the unseen data being assessed. F1-score is calculated as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{4.3}$$

$$Recall = \frac{TP}{TP + FN}, \tag{4.4}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4.5}$$

where TP = true positive, FP = false positive, and FN = false negative predictions.

4.3.2 Overall accuracy

The overall accuracy produces a measure of how many predictions produced by a model on unseen data were correct as a fraction of the total number of unseen observations. This metric was chosen as the main performance metric for models trained on the balanced training set and is calculated using Equation 4.6 below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

where TN = true negative predictions.

The next section will elaborate on the specific hardware and software used during training and the hyperparameter search spaces used during model combination gridsearches.

4.4 Model training

4.4.1 Hardware & Software

During prototyping, a Google Colab notebook with 2 x Intel(R) Xeon(R) CPUs @ 2.20GHz (12GB RAM) and an Nvidia Tesla T4 GPU @ 1.59GHz (16GB RAM) was utilised. Model scripting was performed in Visual Studio Code using Python 3.7 and was backed up on a GitHub repository.

Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: hpc.uct.ac.za. Dell C6420 CPU servers with 40 cores and 384GB RAM each were made available for performing hyperparameter gridsearches as well as data augmentation and curation steps. For training the deep learning models, TensorFlow 2.8 was loaded onto an Nvidia A100 GPU with 20GB RAM and 14 CPU cores. It is pertinent to note that when training CNN-gcForestCS model combinations, CPU servers were used for both CNN feature extraction and for training the gcForestCS classifers so that the hybrid model could run on a single machine without truncation across GPU and CPU servers.

4.4.2 Hyperparameter selection

Tables 4.3–4.6 depict the hyperparameter search spaces used during model combination gridsearches in this analysis. During multi-grained scanning (Table 4.3), three sliding window sizes of 3×3, 5×5, and 7×7 were used since these sizes were associated with high performing gcForest models in previous gcForest image classification tasks (Boualleg, Farah, and Farah 2019; Liu and Yang 2019). For each sliding window, a stride value of 2 was used along both the x- and y-axes, and 5-fold cross-validation was used to reduce the chances of overfitting during training. The number of forests was initially varied, however this led to excessive memory usage. Therefore, the default number and type of forests for multi-grained scanning (2 forests – 1 random forest and 1 extremely-randomised forest) were used. The number of trees in each forest was varied according to applications of gcForest to image classification tasks in previous work, which showed that 50–100 trees produced optimal results (Yin et al. 2018; Liu and Yang 2019; Liu et al. 2019b; Boualleg, Farah, and Farah 2019). For model combinations 1 and 2, only 50 trees were used due to memory constraints.

Model combinations	Min samples per leaf	Max tree depth	Splitting criterion	Nforests	Ntrees	Window sizes	Stride (x,y)
1	10	10	Gini	2 (1 RF, 1 ETS)	[50]	3x3, 5x5, 7x7	(2,2)
2	10	10	Gini	2 (1 RF, 1 ETS)	[50]	3x3, 5x5, 7x7	(2,2)
3-10, 15-25, 27-30	10	10	Gini	2 (1 RF, 1 ETS)	[50, 100]	3x3, 5x5, 7x7	(2,2)

TABLE 4.3: Hyperparameter search space for multi-grained scanning

For the cascade forest gridsearches (Table 4.4), 5-fold cross-validation was performed to reduce chances of overfitting and for similar reasons as in multi-grained scanning, the default number of forests was used (8 forests in total – 4 random forest, 4 extremely-randomised forests). The splitting criterion, maximum tree depth, minimum samples per leaf, the maximum number of layers, and early stopping rounds were all kept constant. The number of

trees in each forest was varied with different search spaces applied to model combinations 1 and 2 due to time and memory constraints.

Model combinations	Min samples per leaf	Max tree depth	Splitting criterion	Nforests	Ntrees	Max layers	Early stopping rounds
1	10	10	Gini	8 (4 RF, 4 ETS)	[50]	20	3
2	10	10	Gini	8 (4 RF, 4 ETS)	[50, 100]	20	3
3–14, 19–30, 35, 37	10	10	Gini	8 (4 RF, 4 ETS)	[50, 100]	20	3

TABLE 4.4: Hyperparameter search space for cascade forests

For gcForestCS and CNN-gcForestCS model combinations where a pooling step was performed after multi-grained scanning, a sliding window of size 2×2 with stride 2 was used and average pooling was performed (Table 4.5).

Model combinations	Pooling window size	Stride (x-axis, y-axis)	Pooling method
31, 32, 36, 38	2x2	(1,1)	average

TABLE 4.5: Hyperparameter search space for model combinations performing a pooling step after multi-grained scanning

For the deep learning hyperparameter gridsearches (Table 4.6), three learning rates of 0.0001, 0.001 (default), and 0.01 were included in the hyperparameter search space while the optimiser search space included the adam and sgd optimisers. A dropout layer was added to the fully connected network where the dropout rate was set as either 0.25, 0.5 or 0.75. The number of epochs and early stopping rounds were held constant at 100 and 75 respectively. Max pooling was used after the dropout layer as this pooling method was used in the best performing deep learning models in previous work done on cassava leaf disease classification (Sambasivam and Opiyo 2021; Enkvetchakul and Surinta 2021; Dhivyaa, Kandasamy, and Rajendran 2022).

Model combinations	Dropout rate	Optimiser	Learning rate	Epochs	Early stopping rounds
15–18, 33, 34	[0.25, 0.5, 0.75]	[adam, sgd]	[0.0001, 0.001, 0.01]	100	75

TABLE 4.6: Hyperparameter search space for deep learning gridsearches

For model combinations trained on an imbalanced training set, validation weighted F1-score was used to determine optimal hyperparameter selection while overall validation accuracy was used for model combinations trained on a balanced training set. However, further investigation found that for imbalanced datasets the same hyperparameter settings yielded the best results whether ranking based on weighted F1-score or overall accuracy for most model combinations.

4.5 Conclusion

This chapter started by summarising all the model combinations implemented in this research. This was followed by an in-depth discussion of the statistical learning methods underpinning these models, along with the details relating to model performance assessment and the specifics relating to model training. The next chapter will present the results and discuss the findings in relation to the research objectives.

Chapter 5

Results and Discussion

This chapter will present and discuss the results relating to each of the research objectives and subsequent research questions as set out in Section 1.3. Section 5.1 will present the results, after which the results will be discussed in relation to each research question (Section 5.2).

5.1 Results

MC	Class balancing?	CNN backbone	Fine-tuned?	FE layer	Classifier	OA (%)	WF1	Training time[1]	Peak RAM (GB)
1	No	None	No	None	DF	54.75	0.46	17:11:27	125.97
2	Yes	None	No	None	DF	52.47	0.473	13:17:52	134.6
3	No	DN201	No	CL1	DF	60	0.511	03:13:42	93.94
4	No	DN201	No	CL2	DF	63.61	0.59	00:57:02	45.32
5	No	DN201	No	CL3	DF	72.31	0.71	00:17:26	13.43
6	Yes	DN201	No	CL1	DF	59.95	0.509	03:16:52	99.65
7	Yes	DN201	No	CL2	DF	61.96	0.598	01:06:17	48.07
8	Yes	DN201	No	CL3	DF	71.35	0.711	00:20:04	14.24
9	No	MNV2	No	CL1	DF	57.93	0.484	02:37:15	70.48
10	No	MNV2	No	CL2	DF	58.09	0.502	00:27:09	29.14
11	No	MNV2	No	CL3	DF	71.14	0.693	00:11:30	8.95
12	Yes	MNV2	No	CL1	DF	56.82	0.483	02:19:43	74.77
13	Yes	MNV2	No	CL2	DF	56.76	0.513	00:26:11	30.91
14	Yes	MNV2	No	CL3	DF	69.07	0.686	00:15:04	9.5
15	No	DN201	Yes	-	FCN	87.43	0.871	01:21:07	3.81
16	Yes	DN201	Yes	-	FCN	85.89	0.859	01:25:05	4.32
17	No	MNV2	Yes	-	FCN	82.87	0.826	00:35:43	1.8
18	Yes	MNV2	Yes	-	FCN	82.55	0.822	00:41:12	1.88
19	No	DN201	Yes	CL1	DF	66.58	0.608	04:58:43	93.94
20	No	DN201	Yes	CL2	DF	81.59	0.806	02:20:17	45.32
21	No	DN201	Yes	CL3	DF	86.68	0.865	02:17:26	13.43
22	Yes	DN201	Yes	CL1	DF	63.82	0.582	04:09:13	99.65
23	Yes	DN201	Yes	CL2	DF	80.95	0.806	02:14:10	48.07
24	Yes	DN201	Yes	CL3	DF	85.41	0.854	01:42:08	14.24
25	No	MNV2	Yes	CL1	DF	59.26	0.521	03:04:10	70.48
26	No	MNV2	Yes	CL2	DF	64.88	0.613	01:21:09	29.14
27	No	MNV2	Yes	CL3	DF	83.66	0.834	00:50:06	8.96
28	Yes	MNV2	Yes	CL1	DF	57.77	0.497	03:28:51	74.77
29	Yes	MNV2	Yes	CL2	DF	60.64	0.543	01:19:40	30.91
30	Yes	MNV2	Yes	CL3	DF	82.23	0.824	00:56:38	9.5

TABLE 5.1: Test set results for base model combinations 1–30

[1] measured in HH:MM:SS

Tables 5.1 and 5.2 depict the overall test results for each of the 38 model combinations tested in this research project. The optimal hyperparameter settings for each model combination are not included in these tables but are included for reference in Appendix A, along with breakdowns of the training execution times and peak memory usage (Appendix B), and test prediction execution times and peak RAM usage (Appendix C).

As explained in the previous chapter, the backbones (where applicable) are either DenseNet201 (DN201) or MobileNetV2 (MNV2); the feature extraction candidate layers (CL1–3) refer to feature maps of size 28×28, 14×14, and 7×7 respectively; the classifier is either a fully connected network (FCN) or deep forest (DF); and reported metrics are overall accuracy (OA) and weighted F1-score (WF1).

MC	Curation?	MGS pooling?	CNN backbone	Fine-tuned?	FE layer	Classifier	OA (%)	WF1	Training time	Peak RAM (GB)
31	No	Yes	DN201	Yes	CL3	DF	86.79	0.866	01:56:48	13.43
32	No	Yes	DN201	Yes	CL3	DF	83.4	0.831	00:50:15	8.95
33	Yes	No	DN201	Yes	-	FCN	85.57	0.851	01:09:05	3.84
34	Yes	No	MNV2	Yes	-	FCN	82.76	0.825	00:29:57	1.04
35	Yes	No	DN201	Yes	CL3	DF	84.51	0.843	01:33:12	12.75
36	Yes	Yes	DN201	Yes	CL3	DF	84.56	0.844	01:25:30	12.75
37	Yes	No	MNV2	Yes	CL3	DF	83.5	0.833	00:41:56	8.5
38	Yes	Yes	MNV2	Yes	CL3	DF	83.24	0.83	00:41:12	8.5

TABLE 5.2: Test set results for model combinations 31–38 based on best CNN-gcForestCS base model combinations

Given the large number of model combinations and the complexity associated with comparing model combinations individually, model combinations will be grouped and assessed based on the research questions outlined in Chapter 1.

5.1.1 Effect of class balance

To assess the effect that training set class imbalance has on CNN-gcForestCS, model combinations were grouped based on whether they fit certain model descriptions, as well as whether they were trained on the original imbalanced dataset, or the balanced version of the training set (for information on how the balanced training set was produced, see Subsection 3.5.3). Model performance was assessed based on the weighted F1-score achieved for the test set, as this is typically a metric used for assessing classification models trained on imbalanced data (for more on how weighted F1-score is calculated, see Subsection 4.3.1). The average test weighted F1-score achieved for the CNN-gcForest model groupings are presented in Table 5.3, while the test weighted F1-score for the gcForestCS and deep transfer learning model combinations is presented in Table 5.4. Figure 5.1 provides a graphical illustration of the effect of class balance across all model descriptions.

Table 5.3 and Figure 5.1 show that there was a negligible change in CNN-gcForestCS model performance when class imbalance was accounted for. For the DenseNet201-gcForestCS models where the CNN feature extractor was not fine-tuned on the training set, there was a negligible improvement in average weighted F1-score when training the model on balanced data (WF1 = +0.002) – however, this came at the cost of slightly longer training time and slightly higher peak memory usage. Conversely, when the DenseNet201-gcForestCS feature extractor was fine-tuned, there was a slight reduction in average weighted F1-score (-0.013), together with a reduction in training time and increase in peak RAM usage, when models

Model description	Mean weighted F1-score		Mean training time		Mean peak RAM usage	
	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced
DN201-gcForestCS (not fine-tuned)	0.604	0.606	01:29:23	01:34:24	50.9	53.99
DN201-gcForestCS (fine-tuned)	0.76	0.747	03:12:08	02:41:50	50.9	53.99
MNV2-gcForestCS (not fine-tuned)	0.56	0.561	01:05:17	01:00:19	36.19	38.39
MNV2-gcForestCS (fine-tuned)	0.656	0.621	01:45:08	01:55:02	36.19	38.39

TABLE 5.3: Comparison of averaged CNN-gcForestCS model performance on the test set across model combinations 1–30 stratified based on whether models were trained on either imbalanced or balanced data

Model description	Mean weighted F1-score		Training time		Peak RAM usage	
	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced
raw resized images	0.46	0.473	17:11:27	13:17:52	125.97	134.6
DenseNet201	0.871	0.859	01:21:07	01:25:05	3.81	4.32
MobileNetV2	0.826	0.822	00:35:43	00:41:12	1.8	1.88

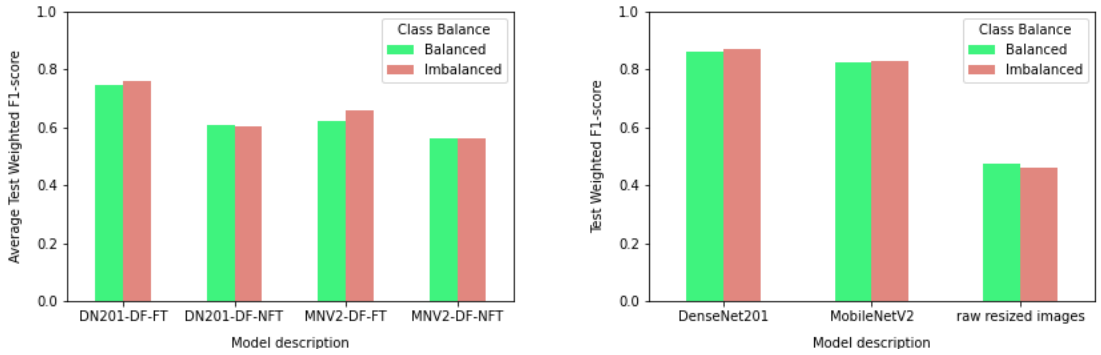
TABLE 5.4: Comparison of test set model performance of deep learning and gcForestCS model combinations across model combinations 1–30 stratified based on whether models were trained on imbalanced or balanced data

were trained on balanced data.

For MobileNetV2-gcForestCS models where the CNN feature extractor was not fine-tuned, there was a negligible improvement in average weighted F1-score (+0.001) and lower training time when balanced training data was used – however, this was at the expense of slightly higher peak memory usage compared to training on the imbalanced training set. When the MobileNetV2 feature extractor was fine-tuned, there was a slight decrease in the average weighted F1-score (-0.035), an increase in training time, and an increase in peak memory usage when balanced training data was used.

As shown in Table 5.4 and Figure 5.1, when the gcForestCS classifier was trained on original resized images ($224 \times 224 \times 3$) in the balanced version of the training set, there was a slight improvement in the weighted F1-score (+0.013), a lower training time and a slightly higher peak RAM usage as compared to training on imbalanced data. The DenseNet201 (-0.012) and MobileNetV2 (-0.004) deep learning models saw a minor reduction in weighted F1-score, and an increase in training time and peak RAM usage when balanced training data was used.

Across all model combination groupings in Tables 5.3 and 5.4, there was a negligible change in model performance when class imbalance was accounted for. Interestingly, this was even the case for the deep learning model combinations which typically generalise better to unseen data when trained on more data provided from data augmentation steps. Overall, the deep learning models perform the best of the model combinations trained on either balanced or imbalanced training data, followed by DenseNet201-gcForestCS (fine-tuned), MobileNetV2-gcForestCS (fine-tuned), DenseNet201-gcForestCS (not fine-tuned), MobileNetV2-gcForestCS (not fine-tuned) and finally training gcForestCS on the raw resized images. There was an increase in peak RAM usage when the balanced training set was used, while the training time either increased or decreased depending on the model combination.



(A) CNN-gcForestCS models

(B) gcForestCS and deep learning models

FIGURE 5.1: Illustration of the effect of training set class balance on test classification performance of different model combination groupings

5.1.2 Effect of CNN feature extraction

The effect on model performance by applying CNN feature extraction before training a gcForestCS classifier was assessed across 3 different convolutional candidate layers from two different transfer learning CNN backbones, DenseNet201 (presented in Table 5.5) and MobileNetV2 (Table 5.6).

Input dimensions	Mean accuracy (%)	Mean weighted F1-score	Mean training time	Mean peak RAM usage (GB)
224x224x3	53.61	0.467	15:14:32	130.29
28x28x256	62.59	0.553	03:54:37	96.8
14x14x896	72.03	0.7	01:39:26	46.7
7x7x1920	78.94	0.785	01:09:16	13.84

TABLE 5.5: Comparison of CNN-gcForestCS model performance on the test set across different DenseNet201 feature map input sizes for model combinations 1–30

Input dimensions	Mean accuracy (%)	Mean weighted F1-score	Mean training time	Mean peak RAM usage (GB)
224x224x3	53.61	0.467	15:14:32	130.29
28x28x192	57.95	0.496	02:52:29	72.63
14x14x576	60.09	0.543	01:11:02	30.03
7x7x1280	76.53	0.759	00:33:19	9.23

TABLE 5.6: Comparison of CNN-gcForestCS model performance on the test set across different MobileNetV2 feature map input sizes for model combinations 1–30

When either the DenseNet201 or MobileNetV2 transfer learning architectures were used for feature extraction in CNN-gcForestCS model combinations, there was an improvement in average overall accuracy and average weighted F1-score, and a reduction in training time and peak memory usage as compared to not performing CNN feature extraction before gcForestCS model training. For DenseNet201-gcForestCS, the largest improvement in average gcForestCS model performance came from reducing the feature map size from $28 \times 28 \times 256$

to 14×14×896 (OA = +9.44%, WF1 = +0.147, training time = -02:15:11, peak RAM usage = -50.1GB) while for MobileNetV2, the largest improvement in performance came from reducing the feature map size from 14×14×576 to 7×7×1280 (OA = +16.44%, WF1 = +0.216, training time = -00:37:43, peak RAM usage = -20.8).

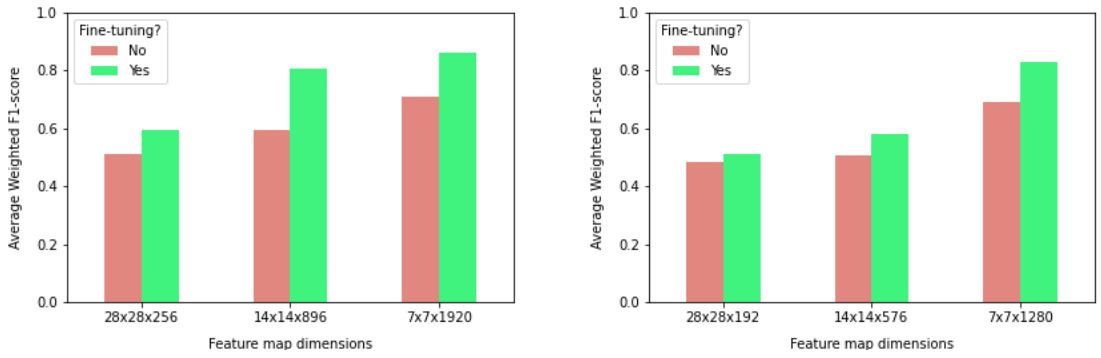
Overall, there was an improvement in model performance, training time and peak memory usage when smaller feature maps were used. DenseNet201-gcForestCS showed larger improvements in model performance as feature size reduced when compared to MobileNetV2-gcForestCS. Tables 5.5 and 5.6 also indicate that when comparing CNN-gcForestCS performance across the CNN backbones used when feature map size is held constant, DenseNet201-gcForestCS performs slightly better than MobileNetV2-gcForestCS at the cost of longer training times and higher peak RAM usage.

5.1.3 Effect of CNN feature extractor fine-tuning

To assess the effect of fine-tuning CNN feature extractor weights on CNN-gcForestCS model performance, CNN-gcForestCS model combinations were grouped based on whether the CNN feature extractor had been fine-tuned on the training set, the results of which are displayed in Table 5.7 (for more information of the process of fine-tuning, see Subsubsection 4.2.1.6). These model combination groupings were then additionally stratified based on feature map size, the results of which are shown in Figure 5.2.

CNN-gcForestCS backbone	**Mean accuracy (%)**		**Mean WF1**		**Mean training time**		**Mean peak RAM usage**	
	Not FT	FT	Not FT	FT	Not FT	FT	Not FT	FT
DenseNet201	64.86	77.51	0.605	0.754	01:31:53	02:56:59	52.44	52.44
MobileNetV2	61.64	68.07	0.56	0.639	01:02:48	01:50:05	37.29	37.29

TABLE 5.7: Comparison of CNN-gcForestCS model performance on the test set for each CNN backbone stratified based on whether the CNN feature extractor was fine-tuned (FT) or not (Not FT)



(A) DenseNet201-gcForestCS (B) MobileNetV2-gcForestCS

FIGURE 5.2: Comparison of the effect of CNN feature extractor fine-tuning on CNN-gcForestCS model performance on the test set across different feature map sizes for model combinations 1–30

Table 5.7 shows that CNN feature extractor fine-tuning leads to a notable improvement in average overall accuracy and average weighted F1-score at the cost of longer training

times. DenseNet201-gcForestCS models show larger improvement in average model performance (OA = +12.65%, WF1 = +0.149) than MobileNetV2-gcForestCS (OA = +6.43%, WF1 = +0.079) – however, DenseNet201-gcForestCS training time and peak RAM usage is higher than MobileNetV2-gcForestCS.

Figure 5.2 shows that across feature map sizes, fine-tuning led to an improvement in average CNN-gcForestCS model performance for both CNN backbones used for feature extraction. For DenseNet201, the largest improvement was seen when 14×14 feature maps were used (WF1 = +0.212) while for MobileNetV2, it was when 7×7 feature maps were used (WF1 = +0.139). These results also corroborate the findings in the previous subsection which showed how CNN-gcForestCS model performance improves as feature map size reduced.

5.1.4 Effect of adding a pooling step after multi-grained scanning

To assess the effect of adding a pooling step after multi-grained scanning via a 2×2 sliding window with stride 1 that performs average pooling (for more information on how pooling works in multi-grained scanning, see Subsubsection 4.2.2.2), equivalent CNN-gcForestCS model combinations that only differed based on whether a pooling step was applied after multi-grained scanning were compared (Table 5.8).

CNN backbone	Feature map dimensions	Mean accuracy (%)		Mean WF1		Mean training time		Mean peak RAM usage (GB)	
		No pooling	Pooling	No pooling	Pooling	No pooling	Pooling	No pooling	Pooling
DenseNet201	7x7x1920	85.6	85.68	0.854	0.855	01:55:19	01:41:08	13.09	13.09
MobileNetV2	7x7x1280	83.58	83.32	0.834	0.831	00:46:00	00:45:43	8.73	8.73

TABLE 5.8: Comparison of average CNN-gcForestCS model performance on the test set for equivalent model combinations that differ based on whether a pooling step was performed after multi-grained scanning from model combinations 31–38

The results show that there is a negligible change in average CNN-gcForestCS model performance when either DenseNet201 or MobileNetV2 were used for feature extraction. There was a minor improvement in average model performance (OA = +0.08%, WF1 = +0.001) when a pooling step was used for DenseNet201-gcForestCS, while reducing training time (-00:14:08). For MobileNetV2-gcForestCS, there was a minor reduction in average model performance (OA = -0.26%, WF1 = -0.003), whilst slightly reducing training time (-00:00:17).

5.1.5 Effect of training set curation

To assess whether improving the quality of the training data via training set curation improves model test performance, equivalent deep learning (Table 5.9) and CNN-gcForestCS (Table 5.10) model combinations that differed based on whether they were trained on curated training data were compared.

Transfer learning model	Overall Accuracy (%)		Weighted F1-score		Training time		Peak RAM usage	
	No curation	Curation	No curation	Curation	No curation	Curation	No curation	Curation
DenseNet201	87.43	85.57	0.871	0.851	01:21:07	01:09:05	3.81	3.84
MobileNetV2	82.87	82.76	0.826	0.825	00:35:43	00:29:57	1.8	1.04

TABLE 5.9: Comparison of test performance of equivalent deep learning model combinations trained with or without curated training data

CNN backbone	Feature map dimensions	Mean accuracy (%)		Mean WF1		Mean training time		Mean peak RAM usage	
		No curation	Curation	No curation	Curation	No curation	Curation	No curation	Curation
DenseNet201	7x7x1920	86.74	84.54	0.866	0.844	01:56:50	01:29:20	13.43	12.75
MobileNetV2	7x7x1280	83.53	83.37	0.833	0.832	00:50:10	00:41:38	8.96	8.5

TABLE 5.10: Comparison of average test performance of fine-tuned CNN-gcForestCS models trained with or without curated training data

Table 5.9 shows that training set curation leads to a reduction in deep learning model performance of DenseNet201 (OA = -1.86%, WF1 = -0.02) and MobileNetV2 (OA = -0.11%, WF1 = -0.001), with an associated reduction in training time (-00:12:02). Interestingly, there is a slight increase in peak RAM usage when training the DenseNet201 transfer learning model (+0.03GB), while there was a reduction in peak memory usage when training MobileNetV2 (-0.76GB). A similar trend was observed for the CNN-gcForestCS model combinations (Table 5.10), where curation yielded a slight reduction in model performance for both DenseNet201-gcForestCS (OA = -2.2%, WF1 = -0.022) and MobileNetV2-gcForestCS (OA = -0.16%, WF1 = -0.001), with a reduction in training time and peak RAM usage (training time = -00:27:30, peak RAM = -0.68GB). Overall, the results show that there is little change in average model performance for MobileNetV2-gcForestCS model combinations while for DenseNet201-gcForestCS, model performance slightly decreases when models are trained on curated training data.

5.1.6 CNN-gcForestCS as an alternative to deep learning

To assess whether CNN-gcForestCS is a viable alternative to CNNs for cassava leaf disease detection, the best deep learning and best CNN-gcForestCS model combination for each CNN backbone were compared (Table 5.11).

Model type	Overall Accuracy (%)	Weighted F1-score	Training time	Peak Memory Usage (GB)
DenseNet201	87.43	0.871	01:21:07	3.81
DenseNet201-gcForestCS	86.79	0.866	01:56:48	13.43
MobileNetV2	82.87	0.826	00:35:43	1.8
MobileNetV2-gcForestCS	83.66	0.834	00:50:06	8.96

TABLE 5.11: Comparison of model test performance for best CNN model and best CNN-gcForestCS for each CNN backbone used

The results in Table 5.11 show that the performance of the best DenseNet201-gcForestCS model was slightly lower than that of DenseNet201 (OA = -0.64%, WF1 = -0.005), while there was an increase in training time and an especially large increase in peak RAM usage (training time = +00:35:41, peak RAM = +9.62GB). MobileNetV2-gcForestCS slightly outperformed MobileNetV2 (OA = +0.79%, WF1 = +0.008) at the expense of longer training time and a much higher peak RAM usage (training time = +00:14:23, peak RAM = +7.16GB). Overall, DenseNet201 and DenseNet201-gcForestCS outperform MobileNetV2 and MobileNetV2-gcForestCS respectively, each at the cost of longer training time and higher peak RAM usage.

5.2 Discussion

With the results being outlined in the previous section, this section will discuss these results in relation to each research question. The sub-research objectives will be discussed first, after which the main research question – whether CNN-gcForestCS is a viable alternative to deep learning for this problem – will be discussed.

1. **What is the effect of class balance on the performance of CNN-gcForestCS?**

   CNN-gcForestCS model combination groups trained on balanced data showed a minimal difference in average model performance compared to being trained on imbalanced training data, which is surprising given that accounting for class imbalance would have been expected to improve the generalisability of tree-based ensembling methods as indicated by Batista, Prati, and Monard (2004) and Yuan and Zhao (2019). Possible explanations for this may include that there was an overrepresentation of majority classes in the test set, or that the downsampling of majority classes in the protocol followed during training set class balancing may have led to a reduction in variation that could be modelled during training.

   For the two deep transfer learning models, there was a decrease in test performance as a result of class balancing. Based on Japkowicz and Stephen (2002) and Liu and Wang (2021), this is unexpected given that addressing class imbalance typically improves deep learning model generalisability in general and for plant leaf disease classification tasks. This finding helps to support the case that downsampling of the majority classes is likely the reason for the reduction in model generalisability seen. It should be noted that only upsampling the under-represented classes was not considered in this research due to time and resource constraints.

   Upon stratifying the CNN-gcForestCS model combination groupings based on whether their feature extractor had been fine-tuned, there is a reduction in the average performance of the fine-tuned DenseNet201-gcForestCS model combination when balanced training data was used. This may be due to premature confidence screening in the cascade forest module, indicating that certain observations that may have required further processing were predicted too early by the gcForestCS classifier. This is supported by the reduction in training times for DenseNet201-gcForestCS, which did not occur for the MobileNetV2-gcForestCS model combination grouping. This may indicate that the MobileNetV2 transfer learning model is more robust to the effects of class balancing compared to DenseNet201 which was also reflected in the difference in performance of the DenseNet201-gcForestCS model combinations based on class balance.

2. **What is the effect of CNN feature extraction, and feature map size, on the performance of gcForestCS?**

   With regards to applying CNN feature extraction before training a gcForestCS classifier, the improvement in model performance over using the raw resized images is likely due to the more specific spatial information captured between important image features by feature maps, as well as the reduction in width and height of inputs provided by the feature maps. Images with low dimensionality produced high image classification performance in both the original gcForest implementation paper by Zhou and Feng (2017) and the gcForestCS implementation by Pang et al. (2018), and therefore this result was to be expected.

Furthermore, smaller feature map sizes are extracted from deeper within the transfer learning networks used, allowing for more refining of spatial feature relationships across multiple convolution steps. This notion is further supported by the findings that DenseNet201-gcForestCS generalises better than MobileNetV2-gcForestCS, as DenseNet201 is a deeper transfer learning architecture than MobileNetV2. Deeper learning architectures may lead to more pertinent features being extracted during convolution – however, as outlined in Krizhevsky, Sutskever, and Hinton (2012), Russakovsky et al. (2015), and Shen, Wu, and Suk (2017), this is dependent on factors regarding dataset size which has also been an issue with regards to plant leaf disease detection applications specifically (Liu and Wang 2021). However, it is important to note that the number of channels for each feature map size was greater in DenseNet201 than in MobileNetV2. This may have allowed for DenseNet201-gcForestCS models to be trained on more information than MobileNetV2-gcForestCS for each feature map size as well, in addition to a deeper CNN feature extractor being used.

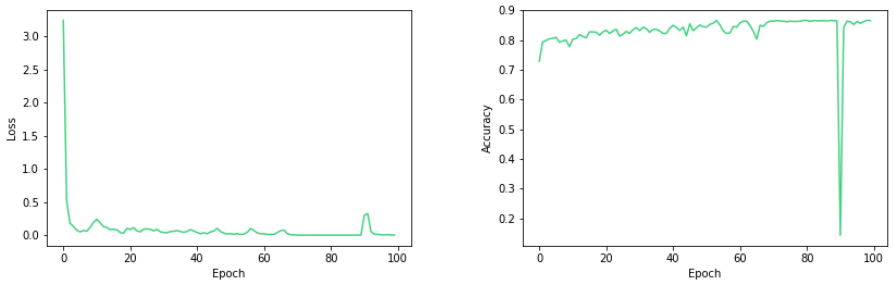
The reduction in training times and peak RAM usage when feature map height and width reduce is to be expected because the process of instance generation at the beginning of multi-grained scanning is the most memory-intensive part of the gcForest algorithm (Zhou and Feng 2017). When image height and width is large, many more instances are created before multi-grained scanning when compared to smaller input sizes provided by feature maps.

3. **What is the effect of CNN feature extractor fine-tuning on the performance of CNN-gcForestCS?**

   The effect of fine-tuning convolutional module weights of the transfer learning models on the training set showed an improvement in CNN-gcForestCS model performance over using the default ImageNet weights. This was to be expected given that transfer learning models with ImageNet weights tend to generalise well for majority of image-based processing tasks but to allow for more domain-specific variation to be captured by the models, fine-tuning of later convolutional layers needs to occur (Shin et al. 2016).

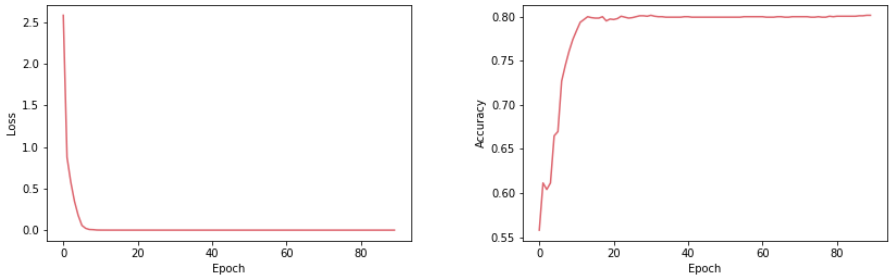
   The better test performance of DenseNet201-gcForestCS compared to MobileNetV2-gcForestCS may be due to the deeper learning that occurs in DenseNet201, as well as the fact that a larger proportion of the convolutional module in DenseNet201 (92.79%) was fine-tuned than in MobileNetV2 (79.87%) which may have allowed the DenseNet201 architecture to generalise better than MobileNetV2. The deeper learning that occurs in DenseNet201 is also reflected in the longer training time and higher peak RAM usage than MobileNetV2 which is due to the larger number of trainable model parameters and layers which need to be stored in memory and updated during training (Huang et al. 2017).

   Taking a closer look at the fine-tuning process for both DenseNet201 (Figure 5.3) and MobileNetV2 (Figure 5.4) shows that the fine-tuning process for both CNN backbones led to successful loss function convergence during training, and an improvement in deep learning model generalisability to cassava leaf disease detection over using default ImageNet weights. There was a sharp drop in performance in the training of DenseNet201 around the 90th epoch, but given that a callback to restore the epoch that produced the best weights was specified, this had no effect on the final model's test performance.



(A) Loss (B) Validation accuracy

FIGURE 5.3: Deep learning training graphs for best DenseNet201 model combination



(A) Loss (B) Validation accuracy

FIGURE 5.4: Deep learning training graphs for best MobileNetV2 model combination

As alluded to earlier, CNN-gcForestCS model performance improves as feature map size reduces. For each feature map size, fine-tuning led to an improvement in test performance, with the largest improvement for DenseNet201-gcForestCS coming from using 14×14 feature maps over 28×28 feature maps. This improvement is likely due to the increase in the number of input matrix elements that occurs when going from 28×28×56 to 14×14×896, allowing the model to be trained on more information for each training image, as well as 14×14×896 being a candidate layer found deeper within the network, allowing for greater refinement of spatial relationships by the CNN feature extractor. As for MobileNetV2-gcForestCS, where the largest improvement in performance came from using 7×7 over 14×14 feature maps, it was also likely the case that the improvement in CNN-gcForestCS came from extracting feature maps from later convolutional layers.

4. **What is the effect of adding a pooling step after multi-grained scanning on performance of CNN-gcForestCS?**

   The addition of a pooling step after multi-grained scanning (2×2 sliding window, stride 1, average pooling), which is based on the idea of pooling layers used in CNN convolutional modules, showed a negligible change in CNN-gcForestCS model performance. This is possibly due to the small-scale dimensionality reduction that occurred when applying pooling to the best CNN-gcForestCS model combinations that extracted 7×7 feature maps. It is therefore potentially the case that pooling may have more of an effect on test performance when larger feature map sizes are used – however, this is conjecture given the lack of findings in this research project to support

this claim, and therefore future work should look into investigating the application of multi-grained scanning pooling when larger feature map sizes are used during CNN feature extraction.

The results also indicate that there was a slight improvement in model performance when pooling was applied during DenseNet201-gcForestCS training, while this was not the case for MobileNetV2-gcForestCS. This may indicate that the benefits of pooling only become noticeable when there are more elements present in the feature maps, as there were more channels in the DenseNet201-gcForestCS 7×7 feature maps than in the MobileNetV2-gcForestCS 7×7 feature maps. The reduction in training time, while peak RAM usage remained constant, was to be expected as the majority of memory usage comes from the formation of instances which occurs before pooling is applied (Zhou and Feng 2017). Reducing the size of outputs from multi-grained scanning via pooling seemed to allow for quicker training of forests in each level of the cascade forest module, yielding a reduction in training time.

5. **What is the effect of training set curation on the performance of CNN-gcForestCS?**

   The reduction in the test performance of the deep learning and CNN-gcForestCS model combinations when training set curation occurred may indicate that the inclusion of images with poor focus/exposure on leaves does not affect training data quality as much as anticipated, and including them seems to allow models to improve their generalisation to unseen data. This is likely indicative of the challenge associated with striking a balance between training data quality and quantity, which affects deep learning generalisability as outlined by Najafabadi et al. (2015) and Sarker (2021).

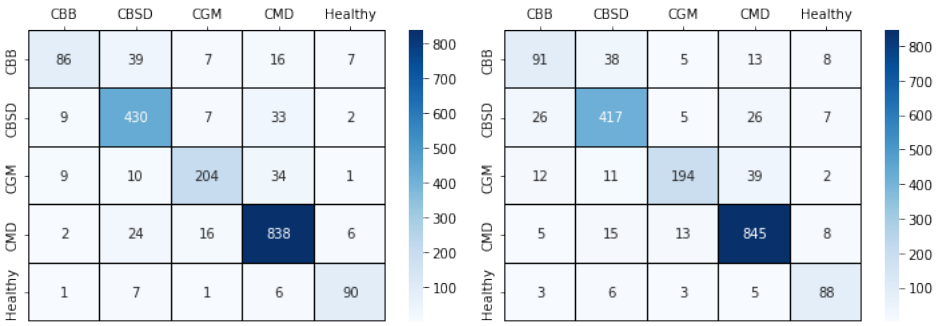
   The reduction in test performance of DenseNet201-gcForestCS and not MobileNetV2-gcForestCS may indicate, as alluded to earlier, that MobileNetV2 is less sensitive to changes in the training set due to its shallower learning architecture as compared to DenseNet201, which then affects the training of the associated gcForestCS classifiers after CNN feature extraction.

6. **Is CNN-gcForestCS a viable alternative to deep learning?**

   The results indicate that CNN-gcForestCS achieves similar performance to deep learning. DenseNet201-gcForestCS performs marginally worse than DenseNet201, whereas MobileNetV2-gcForestCS performs marginally better than MobileNetV2. These results may indicate that CNN-gcForestCS outperforms deep learning when shallower deep transfer learning models are employed (MobileNetV2) – this is corroborated by previous work which compared CNN-gcForest to deep transfer learning for an image classification task (Boualleg, Farah, and Farah 2019) – while deeper learning architectures (DenseNet201) seem to outperform CNN-gcForestCS. However, based on the findings in this dissertation and given that coupling CNN feature extraction and deep forest is a relatively new methodology with few past applications, there is little evidence to substantiate this claim.

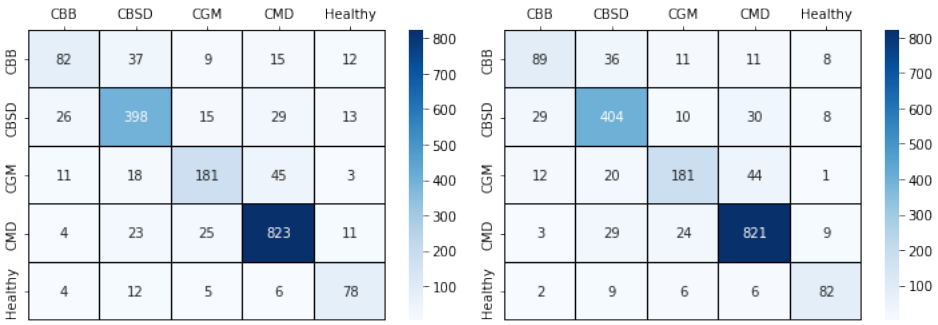
   When comparing the training times and peak RAM usage of the best CNN-gcForestCS models to deep learning, the latter requires shorter training time and lower memory usage than CNN-gcForestCS. This is due to CNNs being able to train on GPU hardware with mini-batch processing during training which gcForestCS is unable to do. The higher training time of CNN-gcForestCS is reflected by the best CNN-gcForestCS models requiring fine-tuning of the CNN feature extractors before CNN-gcForestCS

training, which constitutes almost all of the training time of the deep learning models. The higher peak RAM usage of CNN-gcForestCS is due to the full training set needing to be loaded into memory before training a gcForestCS classifier. Fitting CNN-gcForestCS models to the cassava leaf dataset used in this dissertation required ∼16GB of RAM, indicating that this methodology is not suitable for lower-resource environments. Deep learning within this research project and in previous benchmarking papers (as outlined in Chapter 2) allowed for more efficient usage of computational resources than CNN-gcForestCS while producing similar or better performance.



(A) DenseNet201 (B) DenseNet201-gcForestCS

FIGURE 5.5: Comparison of best model combination test confusion matrices produced from deep learning and CNN-gcForestCS for DenseNet201 backbone



(A) MobileNetV2 (B) MobileNetV2-gcForestCS

FIGURE 5.6: Comparison of best model combination test confusion matrices produced from deep learning and CNN-gcForestCS for MobileNetV2 backbone

Figures 5.5 and 5.6 provide a more in-depth comparison of test performance between the best CNN-gcForestCS model and its deep learning equivalent for each CNN backbone. For DenseNet201, there are slightly more correct classifications across all classes except for the CMD class when compared to the best DenseNet201-gcForestCS model, whilst for MobileNetV2 there was a slight improvement in the number of correct classifications across the CBB, CBSD and Healthy classes for MobileNetV2-gcForestCS as compared to its deep learning equivalent. This possibly indicates that MobileNetV2-gcForestCS is better able to classify minority classes in unseen data than MobileNetV2. Overall, it remains inconclusive as to whether CNN-gcForestCS is a viable alternative to deep learning given the similar performance to deep learning and its high computational cost during training.

Although many confounding factors were accounted for, this analysis was not devoid of limitations. When choosing candidate convolutional layers, the best matching feature map sizes from each transfer learning network were chosen for each CNN-gcForestCS model combination. However, the DenseNet201 feature maps had more channels than MobileNetV2 feature maps, allowing DenseNet201-gcForestCS models to be trained on more information than MobileNetV2-gcForestCS models. This may have accounted for the improved DenseNet201-gcForestCS model performance compared to MobileNetV2-gcForestCS. As outlined in the limitations associated with the dataset labels in Chapter 3, some plants can have co-infection which may lead to multiple disease symptoms being presented, yet only one label is given to an image based on the most prevalent disease type present. Anomalies in the training images were also included which may have led to models learning to distinguish classes based on whether an anomaly associated with a class label was present instead of the difference in leaf symptoms associated with disease types. This indicates that future work on training set curation may need to account for these anomalies but it is speculative as to whether these anomalies will have any effect on model generalisability.

The above chapter provided a detailed analysis of the results achieved in this research project in relation to each of the research questions. The final chapter, Chapter 6, will assess the achievement of these research objectives, provide considerations for future work, and summarise the work contained within this dissertation.

Chapter 6

Conclusions

The conclusions drawn for each research question are outlined below based on the findings in Chapter 5. Considerations for future research to account for the limitations found in this analysis are provided, after which final remarks will be presented.

6.1 Answers to research questions

1. **What is the effect of class balance on the performance of CNN-gcForestCS?**

   Addressing class imbalance in the training set led to a negligible change in gcForestCS, CNN and CNN-gcForestCS model performance. This indicates that the effect of class balance on CNN-gcForestCS performance, which was expected to affect model performance, was minimal.

2. **What is the effect of CNN feature extraction, and feature map size, on the performance of gcForestCS?**

   Applying CNN feature extraction significantly improves model performance of gcForestCS across both CNN transfer learning backbones used when compared to training a gcForestCS classifier on raw resized images. Smaller feature map sizes lead to better model performance while reducing training times and peak RAM usage. Overall, DenseNet201-gcForestCS outperformed MobileNetV2-gcForestCS across all feature map sizes.

3. **What is the effect of CNN feature extractor fine-tuning on the performance of CNN-gcForestCS?**

   Fine-tuning the latter part of the convolutional modules of DenseNet201 and MobileNetV2 led to an improvement in CNN-gcForestCS model performance. This was also reflected when models were grouped based on feature map size where there was an improvement in model performance as smaller feature map sizes were used.

4. **What is the effect of adding a pooling step after multi-grained scanning on performance of CNN-gcForestCS?**

   The addition of a pooling step after multi-grained scanning showed a negligible change in CNN-gcForestCS model performance. However, the small feature map sizes of the CNN-gcForestCS models where pooling was applied indicate that the benefits of dimensionality reduction would have been minimal, and therefore the benefits of pooling might only be realised if applied to larger feature map sizes.

5. **What is the effect of training set curation on the performance of CNN-gcForestCS?**

   It was expected that CNN-gcForestCS model performance would improve as a result of training set curation. However, worse performance was observed, with the difference between DenseNet201 and DenseNet201-gcForestCS being greater than MobileNetV2 and MobileNetV2-gcForestCS. This indicates that poor focus/exposure had less effect on training data quality, and therefore model generalisability, than was initially thought.

6. **Is CNN-gcForestCS a viable alternative to deep learning?**

   The best DenseNet201 model combination outperforms the best DenseNet201-gcForestCS model combination, while the best MobileNetV2-gcForestCS model combination outperforms the best MobileNetV2 model combination. However, the slight improvement in MobileNetV2-gcForestCS model performance over MobileNetV2 comes at the cost of longer training time and peak memory usage. CNN-gcForestCS may present a competitive alternative performance to deep learning for cassava leaf disease detection, but based on the results produced in this research project, it is unclear whether CNN-gcForestCS is an improvement over CNNs and justifies the additional computational cost during training.

6.2 Considerations for future research

To account for the limitations associated with this research, future work on similar applications could look into investigating the following:

- The use of U-Net image segmentation pre-processing steps before modelling, which would allow for anomalies and background noise to be removed.
- The use of different early stopping criteria for the cascade forest classifier when determining the addition of layers, i.e. using validation F-score instead of validation accuracy as loss function.
- Following a different protocol during class balancing than that used in this analysis where no downsampling of majority classes needs to occur.
- Using different CNN backbones for feature extraction that also performed well for cassava leaf disease detection (such as EfficientNetB4 or ResNet).
- Applying pooling steps after multi-grained scanning to feature maps larger than 7×7.

6.3 Summary

The central aim of the above dissertation was to challenge the popular usage of deep learning for image classification tasks in plant disease detection. An investigation of each research question was thoroughly carried out, with results showing that class balance, the addition of a pooling step after multi-grained scanning, and training models on curated training data lead to minimal changes in model generalisability, whereas including CNN feature extraction, and particularly fine-tuned feature extractors, before training a gcForestCS classifier led to improvements in gcForestCS model performance on the test set.

Deep learning still remains a highly effective and efficient statistical learning methodology for cassava disease detection. The initial claim of gcForestCS needing lower resource requirements than deep learning during training was unfounded in this work; deep learning required less memory and training time while producing similar performance to CNN-gcForestCS. CNN-gcForestCS might present a viable alternative to traditional deep learning – however, it is inconclusive as to whether CNN-gcForestCS methodology offers an improvement over standard deep transfer learning approaches, and therefore further investigation will be required to better determine whether this is the case.

Appendix A

Optimal Hyperparameter Settings

Model Combination	Number of trees (MGS)	Number of trees (CF)	Pooling after MGS?
1	50	50	False
2	50	100	False
3	100	50	False
4	100	100	False
5	100	50	False
6	100	100	False
7	100	100	False
8	100	100	False
9	100	100	False
10	50	100	False
11	100	50	False
12	100	100	False
13	50	100	False
14	100	100	False
19	100	100	False
20	100	50	False
21	50	100	False
22	100	100	False
23	100	50	False
24	50	100	False
25	100	100	False
26	100	100	False
27	50	50	False
28	100	100	False
29	100	100	False
30	50	100	False
31	100	50	True
32	50	50	False
35	100	50	False
36	50	100	True
37	100	50	False
38	50	50	True

TABLE A.1: Optimal hyperparameter settings for gcForestCS and CNN-gcForestCS model combinations

Model combination	**Dropout rate**	**Optimiser**	**Learning rate**
15	0.75	Adam	0.0001
16	0.75	Adam	0.0001
17	0.25	SGD	0.01
18	0.25	Adam	0.0001
33	0.75	Adam	0.0001
34	0.25	SGD	0.01

TABLE A.2: Optimal hyperparameter settings for deep learning model combinations

Appendix B

Training Times & Peak RAM usage

Model Combination	Offline Data Augmentation (s)	CNN Fine-tuning (s)	CNN Feature Extraction (s)	gcForestCS (s)	Total (HH:MM:SS)
1	-	-	-	61887.20	17:11:27
2	14.16	-	-	47857.87	13:17:52
3	-	-	50.25	11571.71	03:13:42
4	-	-	93.02	3329.05	00:57:02
5	-	-	112.74	933.20	00:17:26
6	14.16	-	52.57	11745.00	03:16:52
7	14.16	-	106.87	3856.31	01:06:17
8	14.16	-	109.34	1080.71	00:20:04
9	-	-	17.41	9417.30	02:37:15
10	-	-	22.74	1606.06	00:27:09
11	-	-	22.72	666.91	00:11:30
12	14.16	-	18.12	8350.47	02:19:43
13	14.16	-	25.58	1530.83	00:26:11
14	14.16	-	25.51	864.02	00:15:04
15	-	4867.02	-	-	01:21:07
16	14.16	5091.25	-	-	01:25:05
17	-	2142.71	-	-	00:35:43
18	14.16	2458.18	-	-	00:41:12
19	-	4867.02	47.59	13008.08	04:58:43
20	-	4867.02	88.77	3460.87	02:20:17
21	-	4867.02	107.63	3271.81	02:17:26
22	14.16	5091.25	46.38	9801.31	04:09:13
23	14.16	5091.25	95.21	2848.91	02:14:10
24	14.16	5091.25	109.79	912.34	01:42:08
25	-	2142.71	18.36	8889.32	03:04:10
26	-	2142.71	22.48	2703.93	01:21:09
27	-	2142.71	22.83	840.22	00:50:06
28	14.16	2458.18	18.99	10039.37	03:28:51
29	14.16	2458.18	24.70	2282.93	01:19:40
30	14.16	2458.18	24.40	900.82	00:56:38
31	-	4867.02	101.51	2039.76	01:56:48
32	-	2142.71	22.43	849.78	00:50:15
33	-	4144.76	-	-	01:09:05
34	-	1796.93	-	-	00:29:57
35	-	4144.76	96.94	1350.49	01:33:12
36	-	4144.76	100.96	883.94	01:25:30
37	-	1796.93	22.02	697.16	00:41:56
38	-	1796.93	21.67	653.19	00:41:12

TABLE B.1: Training Time of components in model combinations 1-38

Model Combination	Offline Data Augmentation (MB)	CNN Fine-tuning (MB)	CNN Feature Extraction (MB)	gcForestCS (MB)	Max Memory Usage (GB)
1	-	-	-	125973.19	125.97
2	10838.3	-	-	134596.38	134.6
3	-	-	4543.15	93936.79	93.94
4	-	-	3977.81	45316.19	45.32
5	-	-	2134.75	13427.75	13.43
6	10838.3	-	4819.32	99647.35	99.65
7	10838.3	-	4219.45	48071.03	48.07
8	10838.3	-	2264.21	14244.05	14.24
9	-	-	3407.46	70480.49	70.48
10	-	-	2556.56	29138.48	29.14
11	-	-	1421.69	8953.48	8.95
12	10838.3	-	3614.60	74765.04	74.77
13	10838.3	-	2711.91	30909.89	30.91
14	10838.3	-	1507.99	9497.75	10.84
15	-	3809.68	-	-	3.81
16	10838.3	4324.74	-	-	10.84
17	-	1879.03	-	-	1.88
18	10838.3	1879.82	-	-	10.84
19	-	3809.68	4543.27	93936.78	93.94
20	-	3809.68	3978.37	45316.17	45.32
21	-	3809.68	2135.59	13427.10	13.43
22	10838.3	4324.74	4819.45	99647.38	99.65
23	10838.3	4324.74	4220.02	48071.04	48.07
24	10838.3	4324.74	2265.05	14243.42	14.24
25	-	1879.03	3407.53	70480.48	70.48
26	-	1879.03	2556.70	29139.47	29.14
27	-	1879.03	1421.88	8957.23	8.96
28	10838.3	1879.82	3614.66	74765.09	74.77
29	10838.3	1879.82	2712.05	30910.83	30.91
30	10838.3	1879.82	1508.18	9497.14	10.84
31	-	3809.68	2135.60	13427.73	13.43
32	-	1879.03	1421.88	8953.52	8.95
33	-	3837.31	-	-	3.84
34	-	1882.71	-	-	1.88
35	-	3837.31	2027.59	12746.79	12.75
36	-	3837.31	2027.59	12751.04	12.75
37	-	1882.71	1349.87	8503.58	8.5
38	-	1882.71	1349.87	8503.59	8.5

TABLE B.2: Training peak RAM usage of components in model combinations 1-38

Appendix C

Test Prediction Times & Peak RAM Usage

Model Combination	CNN feature extraction (s)	FCN classification (s)	gcForestCS classification (s)	Total (s)	Total (HH:MM:SS)
1	-	-	835.28	835.28	00:13:55
2	-	-	829.83	829.83	00:13:50
3	15.93	-	102.75	118.68	00:01:59
4	29.02	-	102.08	131.11	00:02:11
5	33.99	-	146.26	180.25	00:03:00
6	15.68	-	124.64	140.32	00:02:20
7	30.53	-	104.65	135.18	00:02:15
8	30.01	-	132.07	162.08	00:02:42
9	5.06	-	125.65	130.71	00:02:11
10	7.09	-	51.05	58.14	00:00:58
11	7.05	-	92.60	99.65	00:01:40
12	5.36	-	92.67	98.03	00:01:38
13	6.92	-	41.79	48.71	00:00:49
14	7.28	-	92.21	99.49	00:01:39
15	-	19.23	-	19.23	00:00:19
16	-	19.33	-	19.33	00:00:19
17	-	5.36	-	5.36	00:00:05
18	-	5.08	-	5.08	00:00:05
19	15.08	-	190.43	205.50	00:03:26
20	25.99	-	120.98	146.97	00:02:27
21	32.67	-	287.06	319.72	00:05:20
22	14.17	-	137.20	151.37	00:02:31
23	25.99	-	103.82	129.81	00:02:10
24	31.57	-	148.11	179.68	00:03:00
25	5.81	-	107.89	113.69	00:01:54
26	7.08	-	57.28	64.37	00:01:04
27	7.49	-	96.96	104.45	00:01:44
28	5.69	-	92.76	98.45	00:01:38
29	6.88	-	54.48	61.36	00:01:01
30	6.59	-	102.86	109.45	00:01:49
31	31.57	-	184.83	216.40	00:03:36
32	6.65	-	91.68	98.32	00:01:38
33	-	19.51	-	19.51	00:00:20
34	-	7.56	-	7.56	00:00:08
35	29.14	-	164.17	193.31	00:03:13
36	31.17	-	156.62	187.79	00:03:08
37	6.57	-	98.85	105.42	00:01:45
38	6.42	-	68.88	75.30	00:01:15

TABLE C.1: Breakdown of test prediction execution times for each component in model combinations 1-38

Model Combination	CNN Feature Extraction (MB)	FCN classification (MB)	gcForestCS classification (MB)	Max memory usage (GB)
1	-	-	35634.86	35.63
2	-	-	35634.84	35.63
3	1513.33	-	24495.80	24.5
4	1324.17	-	11933.33	11.93
5	709.39	-	3608.74	3.61
6	1513.34	-	24495.79	24.5
7	1324.19	-	11933.92	11.93
8	709.40	-	3608.58	3.61
9	1135.01	-	18395.15	18.4
10	851.26	-	7677.46	7.68
11	472.94	-	2407.00	2.41
12	1135.03	-	18395.37	18.4
13	851.28	-	7677.46	7.68
14	472.95	-	2407.06	2.41
15	-	5111.31	-	5.11
16	-	5124.86	-	5.12
17	-	0.00	-	0
18	-	0.00	-	0
19	1513.33	-	24495.75	24.5
20	1324.17	-	11933.74	11.93
21	709.39	-	3608.80	3.61
22	1513.34	-	24492.97	24.49
23	1324.19	-	11933.75	11.93
24	709.40	-	3608.60	3.61
25	1135.00	-	18391.53	18.39
26	851.26	-	7677.45	7.68
27	472.93	-	2407.17	2.41
28	1135.01	-	18395.30	18.4
29	851.28	-	7677.75	7.68
30	472.95	-	2406.96	2.41
31	709.39	-	3608.75	3.61
32	472.93	-	2406.97	2.41
33	-	5124.86	-	5.12
34	-	2134.92	-	2.13
35	709.39	-	3608.58	3.61
36	709.39	-	3608.82	3.61
37	472.93	-	2406.81	2.41
38	472.93	-	2406.95	2.41

TABLE C.2: Breakdown of test prediction peak RAM usage for each component in model combinations 1-38

Appendix D

Code

The GitHub repository containing the scripts and notebooks developed within this dissertation can be accessed via this link.

Bibliography

Abadi, M. et al. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Abayomi-Alli, O. et al. (2021). "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning". In: *Expert Systems* 38 (7), e12746. ISSN: 1468-0394. DOI: 10.1111/EXSY.12746. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12746.

Alamu, E. et al. (2019). "Evaluation of cassava processing and utilization at household level in Zambia". In: *Food Security* 11 (1), pp. 141–150. ISSN: 18764525. DOI: 10.1007/S12571-018-0875-3. URL: https://link.springer.com/article/10.1007/s12571-018-0875-3.

Alene, A. et al. (2018). "Identifying crop research priorities based on potential economic and poverty reduction impacts: The case of cassava in Africa, Asia, and Latin America". In: *PLoS ONE* 13 (8), e0201803. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0201803. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0201803.

Alicai, T. et al. (2016). "Cassava brown streak virus has a rapidly evolving genome: implications for virus speciation, variability, diagnosis and host resistance". In: *Scientific Reports* 6 (1), pp. 1–14. ISSN: 2045-2322. DOI: 10.1038/srep36164. URL: https://www.nature.com/articles/srep36164.

Alicai, T. et al. (2019). "Expansion of the cassava brown streak pandemic in Uganda revealed by annual field survey data for 2004 to 2017". In: *Scientific Data* 6 (1), pp. 1–8. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0334-9. URL: https://www.nature.com/articles/s41597-019-0334-9.

Alzubaidi, L. et al. (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8 (1), pp. 1–74. ISSN: 2196-1115. DOI: 10.1186/S40537-021-00444-8. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8.

Andersen, E. et al. (2018). "Disease Resistance Mechanisms in Plants". In: *Genes (Basel)* 9 (7), p. 339. ISSN: 20734425. DOI: 10.3390/GENES9070339. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6071103/.

Anderson, P. et al. (2004). "Emerging infectious diseases of plants: Pathogen pollution, climate change and agrotechnology drivers". In: *Trends in Ecology and Evolution* 19 (10), pp. 535–544. ISSN: 01695347. DOI: 10.1016/J.TREE.2004.07.021.

Ano, C. et al. (2021). "Cassava Brown Streak Disease Response and Association With Agronomic Traits in Elite Nigerian Cassava Cultivars". In: *Frontiers in Plant Science* 12. ISSN: 1664462X. DOI: 10.3389/FPLS.2021.720532. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8646096/.

Arora, J., U. Agrawal, and P. Sharma (2020). "Classification of Maize leaf diseases from healthy leaves using Deep Forest". In: *Journal of Artificial Intelligence and Systems* 2, pp. 14–26. ISSN: 2642-2859. DOI: 10.33969/AIS.2020.21002. URL: https://iecscience.org/jpapers/42.

Asare, P. et al. (2014). "Phenotypic and molecular screening of cassava (Manihot esculentum Crantz) genotypes for resistance to cassava mosaic disease". In: *Journal of General and*

Molecular Virology 6 (2), pp. 6–18. ISSN: 2141-6648. DOI: 10.5897/JGMV2014.0056. URL: https://academicjournals.org/journal/JGMV/article-abstract/92A90B945293.

Avelino, J. et al. (2015). "The coffee rust crises in Colombia and Central America (2008–2013): impacts, plausible causes and proposed solutions". In: *Food Security* 7 (2), pp. 303–321. ISSN: 18764525. DOI: 10.1007/S12571-015-0446-9. URL: https://link.springer.com/article/10.1007/s12571-015-0446-9.

Ayanlade, A. and M. Radeny (2020). "COVID-19 and food security in Sub-Saharan Africa: implications of lockdown during agricultural planting seasons". In: *npj Science of Food* 4 (13), pp. 1–6. ISSN: 2396-8370. DOI: 10.1038/s41538-020-00073-0. URL: https://www.nature.com/articles/s41538-020-00073-0.

Barrett, L. et al. (2009). "Diversity and Evolution of Effector Loci in Natural Populations of the Plant Pathogen Melampsora lini". In: *Molecular Biology and Evolution* 26 (11), pp. 2499–2513. ISSN: 0737-4038. DOI: 10.1093/MOLBEV/MSP166. URL: https://academic.oup.com/mbe/article/26/11/2499/1013842.

Batista, G., R. Prati, and M. Monard (2004). "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD Explorations Newsletter* 6 (1), pp. 20–29. ISSN: 1931-0145. DOI: 10.1145/1007730.1007735. URL: https://dl.acm.org/doi/10.1145/1007730.1007735.

Bauer, E. et al. (1999). "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants". In: *Machine Learning* 36 (1), pp. 105–139. ISSN: 1573-0565. DOI: 10.1023/A:1007515423169. URL: https://link.springer.com/article/10.1023/A:1007515423169.

Baumann, F. et al. (2013). "Cascaded Random Forest for Fast Object Detection". In: *Scandinavian Conference on Image Analysis (SCIA): Image Analysis* 7944, pp. 131–142. ISSN: 03029743. DOI: 10.1007/978-3-642-38886-6_13. URL: https://link.springer.com/chapter/10.1007/978-3-642-38886-6_13.

Bebber, D. (2015). "Range-expanding pests and pathogens in a warming world". In: *Annual review of phytopathology* 53, pp. 335–356. ISSN: 1545-2107. DOI: 10.1146/ANNUREV-PHYTO-080614-120207. URL: https://pubmed.ncbi.nlm.nih.gov/26047565/.

Bechoff, A. et al. (2017). "Cassava traits and end-user preference: Relating traits to consumer liking, sensory perception, and genetics". In: *Critical Reviews in Food Science and Nutrition* 58 (4), pp. 547–567. ISSN: 15497852. DOI: 10.1080/10408398.2016.1202888. URL: https://www.tandfonline.com/doi/abs/10.1080/10408398.2016.1202888.

Bellemare, M. (2015). "Rising Food Prices, Food Price Volatility, and Social Unrest". In: *American Journal of Agricultural Economics* 97 (1), pp. 1–21. ISSN: 1467-8276. DOI: 10.1093/AJAE/AAU038. URL: https://onlinelibrary.wiley.com/doi/10.1093/ajae/aau038.

Bennett, B. (2015). *Guest editorial: Smallholder cassava production and the cassava processing sector in Africa*. 1-2. Vol. 5. Practical Action Publishing, pp. 1–3. DOI: 10.3362/2046-1887.2015.001.

Bi, Y., B. Xue, and M. Zhang (2020). "Evolving Deep Forest with Automatic Feature Extraction for Image Classification Using Genetic Programming". In: *International Conference on Parallel Problem Solving from Nature (PPSN XVI)* 12269, pp. 3–18. ISSN: 16113349. DOI: 10.1007/978-3-030-58112-1_1. URL: https://link.springer.com/chapter/10.1007/978-3-030-58112-1_1.

Bigirimana, S. et al. (2004). "First evidence for the spread of East African cassava mosaic virus - Uganda (EACMV-UG) and the pandemic of severe cassava mosaic disease to Burundi". In: *Plant Pathology* 53 (2), p. 231. ISSN: 00320862. DOI: 10.1111/J.0032-0862.2004.00971.X.

Borrelli, V. et al. (2018). "The Enhancement of Plant Disease Resistance Using CRISPR/Cas9 Technology". In: *Frontiers in Plant Science* 9, p. 1245. ISSN: 1664462X. DOI: 10.3389/FPLS.2018.01245.

Bottou, L. (2012). *Stochastic gradient descent tricks*. Ed. by G. Montavon, G. Orr, and K. Müller. Vol. 7700. Springer, pp. 421–436. ISBN: 9783642352881. DOI: 10.1007/978-3-642-35289-8_25. URL: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_25.

Boualleg, Y., M. Farah, and I. Farah (2019). "Remote Sensing Scene Classification Using Convolutional Features and Deep Forest Classifier". In: *IEEE Geoscience and Remote Sensing Letters* 16.12, pp. 1944–1948. ISSN: 15580571. DOI: 10.1109/LGRS.2019.2911855.

Brasier, C. (2008). "The biosecurity threat to the UK and global environment from international trade in plants". In: *Plant Pathology* 57 (5), pp. 792–808. ISSN: 1365-3059. DOI: 10.1111/J.1365-3059.2008.01886.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1365-3059.2008.01886.x.

Breiman, L. (1996). "Bagging predictors". In: *Machine Learning* 24 (2), pp. 123–140. ISSN: 1573-0565. DOI: 10.1007/BF00058655. URL: https://link.springer.com/article/10.1007/BF00058655.

— (2001). "Random Forests". In: *Machine Learning* 45, pp. 5–32.

Breiman, L. et al. (1984). *Classification and regression trees*. 1st ed. Routledge, pp. 1–358. ISBN: 9781351460491. DOI: 10.1201/9781315139470. URL: https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman-jerome-friedman-richard-olshen-charles-stone.

Brodley, C. and M. Friedl (1997). "Decision tree classification of land cover from remotely sensed data". In: *Remote Sensing of Environment* 61 (3), pp. 399–409. ISSN: 0034-4257. DOI: 10.1016/S0034-4257(97)00049-7.

Brown, J. (2015). "Durable Resistance of Crops to Disease: A Darwinian Perspective". In: *Annual Review of Phytopathology* 53, pp. 513–539. ISSN: 15452107. DOI: 10.1146/ANNUREV-PHYTO-102313-045914. URL: https://www.annualreviews.org/doi/abs/10.1146/annurev-phyto-102313-045914.

Bui, Q. et al. (2021). "Gradient Boosting Machine and Object-Based CNN for Land Cover Classification". In: *Remote Sensing* 13 (14), p. 2709. ISSN: 2072-4292. DOI: 10.3390/RS13142709. URL: https://www.mdpi.com/2072-4292/13/14/2709.

Bujarski, J. (2013). "Genetic recombination in plant-infecting messenger-sense RNA viruses: Overview and research perspectives". In: *Frontiers in Plant Science* 4 (MAR), p. 68. ISSN: 1664462X. DOI: 10.3389/FPLS.2013.00068.

Burdon, J. and J. Zhan (2020). "Climate change and disease in plant communities". In: *PLOS Biology* 18 (11), e3000949. ISSN: 1545-7885. DOI: 10.1371/JOURNAL.PBIO.3000949. URL: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000949.

Burns, A. et al. (2010). "Cassava: The Drought, War and Famine Crop in a Changing World". In: *Sustainability* 2 (11), pp. 3572–3607. ISSN: 2071-1050. DOI: 10.3390/SU2113572. URL: https://www.mdpi.com/2071-1050/2/11/3572.

Butler, C. (2016). "Sounding the Alarm: Health in the Anthropocene". In: *International Journal of Environmental Research and Public Health* 13 (7), p. 665. ISSN: 1660-4601. DOI: 10.3390/IJERPH13070665. URL: https://www.mdpi.com/1660-4601/13/7/665.

Calil, I. and E. Fontes (2017). "Plant immunity against viruses: antiviral immune receptors in focus". In: *Annals of Botany* 119 (5), pp. 711–723. ISSN: 10958290. DOI: 10.1093/AOB/MCW200. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5604577/.

Calvert, L., M. Cuervo, and I. Lozano (2012). *Cassava viral diseases of South America*. Ed. by B. O. Patiño and H. Ceballos. Centro Internacional de Agricultura Tropical (CIAT); Latin American et al., pp. 309–318. URL: https://cgspace.cgiar.org/handle/10568/81857?show=full.

Campo, B., G. Hyman, and A. Bellotti (2011). "Threats to cassava production: Known and potential geographic distribution of four key biotic constraints". In: *Food Security* 3 (329). ISSN: 18764517. DOI: 10.1007/S12571-011-0141-4. URL: https://link.springer.com/

article/10.1007/s12571-011-0141-4#:~:text=The%20goal%20of%20this%20study,and%20cassava%20brown%20streak%20disease..

Cassidy, E. et al. (2013). "Redefining agricultural yields: from tonnes to people nourished per hectare". In: *Environmental Research Letters* 8 (3), p. 034015. ISSN: 1748-9326. DOI: 10.1088/1748-9326/8/3/034015. URL: https://iopscience.iop.org/article/10.1088/1748-9326/8/3/034015.

Ceballos, H. et al. (2011). *Crop Adaptation to Climate Change*. Ed. by S. S. Yadav et al. 1st ed. John Wiley & Sons, Inc. ISBN: 9780813820163.

Chavarriaga-Aguirre, P. et al. (2016). "The potential of using biotechnology to improve cassava: a review". In: *In Vitro Cellular and Developmental Biology - Plant* 52 (5), pp. 461–478. ISSN: 10545476. DOI: 10.1007/S11627-016-9776-3. URL: https://link.springer.com/article/10.1007/s11627-016-9776-3.

Chen, J. et al. (2020). "Using deep transfer learning for image-based plant disease identification". In: *Computers and Electronics in Agriculture* 173, p. 105393. ISSN: 0168-1699. DOI: 10.1016/J.COMPAG.2020.105393.

Chen, T. and C. Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. DOI: 10.1145/2939672. URL: http://dx.doi.org/10.1145/2939672.2939785.

Cheng, C. et al. (2013). "Plant immune response to pathogens differs with changing temperatures". In: *Nature Communications 2013 4:1* 4 (1), pp. 1–9. ISSN: 2041-1723. DOI: 10.1038/ncomms3530. URL: https://www.nature.com/articles/ncomms3530.

Chikoti, P. et al. (2019). "Cassava mosaic disease: a review of a threat to cassava production in Zambia". In: *Journal of Plant Pathology* 101 (3), pp. 467–477. ISSN: 22397264. DOI: 10.1007/S42161-019-00255-0. URL: https://link.springer.com/article/10.1007/s42161-019-00255-0.

Chisenga, S. et al. (2019). "Progress in research and applications of cassava flour and starch: a review". In: *Journal of Food Science and Technology* 56 (6), p. 2799. ISSN: 09758402. DOI: 10.1007/S13197-019-03814-6. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6542882/.

Cock, J. (1982). "Cassava: A Basic Energy Source in the Tropics". In: *Science* 218 (4574), pp. 755–762. ISSN: 00368075. DOI: 10.1126/SCIENCE.7134971. URL: https://www.science.org/doi/10.1126/science.7134971.

Constable, F. et al. (2018). "Detection in Australia of Cucumber green mottle mosaic virus in seed lots of cucurbit crops". In: *Australasian Plant Disease Notes* 13 (1), pp. 1–3. ISSN: 1833928X. DOI: 10.1007/S13314-018-0302-9. URL: https://link.springer.com/article/10.1007/s13314-018-0302-9.

Csorba, T., L. Kontra, and J. Burgyán (2015). "viral silencing suppressors: Tools forged to fine-tune host-pathogen coexistence". In: *Virology* 479-480, pp. 85–103. ISSN: 0042-6822. DOI: 10.1016/J.VIROL.2015.02.028.

de Alba, A., E. Elvira-Matelot, and H. Vaucheret (2013). "Gene silencing in plants: A diversity of pathways". In: *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1829 (12), pp. 1300–1308. ISSN: 1874-9399. DOI: 10.1016/J.BBAGRM.2013.10.005.

Deininger, K. and D. Byerlee (2012). "The Rise of Large Farms in Land Abundant Countries: Do They Have a Future?" In: *World Development* 40 (4), pp. 701–714. ISSN: 0305-750X. DOI: 10.1016/J.WORLDDEV.2011.04.030.

Dhivyaa, C., N. Kandasamy, and S. Rajendran (2022). "Integration of dilated convolution with residual dense block network and multi-level feature detection network for cassava plant leaf disease identification". In: *Concurrency and Computation: Practice and Experience*, e6879. ISSN: 1532-0634. DOI: 10.1002/CPE.6879. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6879.

Dickau, R., A. Ranere, and R. Cooke (2007). "Starch grain evidence for the preceramic dispersals of maize and root crops into tropical dry and humid forest of Panama". In: *Proceedings of the National Academy of Sciences of the United States of America* 104 (9), pp. 3651–3656. ISSN: 00278424. DOI: 10.1073/PNAS.0611605104. URL: www.pnas.org/cgi/content/full/.

Dietterich, T. (2000). "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization". In: *Machine Learning* 40 (2), pp. 139–157. ISSN: 1573-0565. DOI: 10.1023/A:1007607513941. URL: https://link.springer.com/article/10.1023/A:1007607513941.

Ding, B., H. Qian, and J. Zhou (2018). "Activation functions and their characteristics in deep neural networks". In: *Proceedings of the 30th Chinese Control and Decision Conference*, pp. 1836–1841. DOI: 10.1109/CCDC.2018.8407425.

Dong, Y. et al. (2019). "MLW-gcForest: A multi-weighted gcForest model for cancer subtype classification by methylation data". In: *Applied Sciences (Switzerland)* 9.17. ISSN: 20763417. DOI: 10.3390/APP9173589.

Doungous, O. et al. (2022). "Cassava mosaic disease and its whitefly vector in Cameroon: Incidence, severity and whitefly numbers from field surveys". In: *Crop Protection* 158, p. 106017. ISSN: 0261-2194. DOI: 10.1016/J.CROPRO.2022.106017. URL: https://linkinghub.elsevier.com/retrieve/pii/S0261219422001132.

Eigenbrode, S., N. Bosque-Pérez, and T. Davis (2018). "Insect-Borne Plant Pathogens and Their Vectors: Ecology, Evolution, and Complex Interactions". In: *Annual Review of Entomology* 63, pp. 169–191. ISSN: 00664170. DOI: 10.1146/ANNUREV-ENTO-020117-043119.

Ekroth, A., C. Rafaluk-Mohr, and K. King (2019). "Diversity and disease: evidence for the monoculture effect beyond agricultural systems". In: *bioRxiv*, p. 668228. DOI: 10.1101/668228. URL: https://www.biorxiv.org/content/10.1101/668228v1.

El-Sharkawy, M. (2006). "International research on cassava photosynthesis, productivity, eco-physiology, and responses to environmental stresses in the tropics". In: *Photosynthetica* 44 (4), pp. 481–512. ISSN: 1573-9058. DOI: 10.1007/S11099-006-0063-0. URL: https://link.springer.com/article/10.1007/s11099-006-0063-0.

Enkvetchakul, P. and O. Surinta (2021). "Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition". In: *Applied Science and Engineering Progress* 15 (3). ISSN: 26729156. DOI: 10.14416/J.ASEP.2021.01.003.

Evans, S. (2019). "The "Age of Agricultural Ignorance": Trends and Concerns for Agriculture Knee-Deep into the Twenty-First Century". In: *Agricultural History* 93 (1), pp. 4–34. DOI: 10.3098/ah.2019.093.1.004.

Fang, Y. and R. Ramasamy (2015). "Current and Prospective Methods for Plant Disease Detection". In: *Biosensors* 5 (3), p. 537. ISSN: 20796374. DOI: 10.3390/BIOS5030537. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4600171/.

FAO et al. (2020). *Food Security and Nutrition in the World: The State of Transforming Food Systems for Affordable Healthy Diets*. FAO. DOI: 10.4060/ca9692en. URL: https://doi.org/10.4060/ca9692en.

— (2021). *The State of Food Security And Nutrition in the World: The State of Transforming Food Systems for Food Security, Improved Nutrition and Affordable Healthy Diets for All*. FAO. DOI: 10.4060/cb4474en. URL: https://doi.org/10.4060/cb4474en.

FAOSTAT (July 2020). *FAOSTAT*. URL: https://www.fao.org/faostat/en/#home.

Fawke, S., M. Doumane, and S. Schornack (2015). "Oomycete Interactions with Plants: Infection Strategies and Resistance Principles". In: *Microbiology and Molecular Biology Reviews* 79 (3), p. 263. ISSN: 1092-2172. DOI: 10.1128/MMBR.00010-15. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4468149/.

Fedoroff, N. (2015). "Food in a future of 10 billion". In: *Agriculture and Food Security* 4 (1), pp. 1–10. ISSN: 20487010. DOI: 10.1186/S40066-015-0031-7. URL: https://agricultureandfoodsecuri biomedcentral.com/articles/10.1186/s40066-015-0031-7.

Ferguson, M. et al. (2019). "A Global Overview of Cassava Genetic Diversity". In: *PLoS ONE* 14 (11), e0224763. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0224763. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0224763.

Fermont, A. et al. (2009). "Closing the cassava yield gap: An analysis from smallholder farms in East Africa". In: *Field Crops Research* 112 (1), pp. 24–36. ISSN: 0378-4290. DOI: 10.1016/J.FCR.2009.01.009.

Fondong, V. (2017). "The search for resistance to cassava mosaic geminiviruses: How much we have accomplished, and what lies ahead". In: *Frontiers in Plant Science* 8, p. 408. ISSN: 1664462X. DOI: 10.3389/FPLS.2017.00408.

Freund, Y. and R. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55 (1), pp. 119–139. ISSN: 00220000. DOI: 10.1006/JCSS.1997.1504.

— (1999). "A Short Introduction to Boosting". In: *Journal of Japanese Society for Artificial Intelligence* 14 (5), pp. 771–780. URL: https://www.semanticscholar.org/paper/A-Short-Introduction-to-Boosting-Freund-Schapire/c834bddd5e75a64ca9bb80c195cf84345c38bb9b.

Fujita, E. et al. (2017). "Basic Investigation on a Robust and Practical Plant Diagnostic System". In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 989–992. DOI: 10.1109/ICMLA.2016.0178.

Fukushima, K. and S. Miyake (1982). "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position". In: *Pattern Recognition* 15 (6), pp. 455–469. ISSN: 00313203. DOI: 10.1016/0031-3203(82)90024-3.

Geurts, P., D. Ernst, and L. Wehenkel (2006). "Extremely randomized trees". In: *Machine Learning* 63 (1), pp. 3–42. ISSN: 1573-0565. DOI: 10.1007/S10994-006-6226-1. URL: https://link.springer.com/article/10.1007/s10994-006-6226-1.

Giller, K. (2020). "The Food Security Conundrum of sub-Saharan Africa". In: *Global Food Security* 26, p. 100431. ISSN: 2211-9124. DOI: 10.1016/J.GFS.2020.100431.

Gillman, M. and H. Erenler (2009). "The genetic diversity and cultural importance of cassava and its contribution to tropical forest sustainability". In: *Journal of Integrative Environmental Sciencs* 6 (3), pp. 189–200. ISSN: 1943-815X. DOI: 10.1080/19438150903090509. URL: https://www.tandfonline.com/doi/abs/10.1080/19438150903090509.

Gleadow, R. et al. (2009). "Growth and nutritive value of cassava (Manihot esculenta Cranz.) are reduced when grown in elevated CO2". In: *Plant Biology* 11 (SUPPL.1), pp. 76–82. ISSN: 1438-8677. DOI: 10.1111/J.1438-8677.2009.00238.X. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1438-8677.2009.00238.x.

Glorot, X. and Y. Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of Machine Learning Research* 9, pp. 249–256. URL: https://www.semanticscholar.org/paper/Understanding-the-difficulty-of-training-deep-Glorot-Bengio/b71ac1e9fb49420d13e084ac67254a0bbd40f83.

Godfray, H. et al. (2018). "Meat consumption, health, and the environment". In: *Science* 361 (6399). ISSN: 10959203. DOI: 10.1126/SCIENCE.AAM5324. URL: https://www.science.org/doi/abs/10.1126/science.aam5324.

Godlee, F. (1991). "Health implications of climatic change." In: *British Medical Journal* 303 (6812), p. 1256. ISSN: 09598146. DOI: 10.1136/BMJ.303.6812.1254. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1671557/.

Govardhan, M. and M. Veena (2019). "Diagnosis of Tomato Plant Diseases using Random Forest". In: *Global Conference for Advancement in Technology (GCAT)*. DOI: 10.1109/GCAT47503.2019.8978431.

Govindaraj, M., M. Vetriventhan, and M. Srinivasan (2015). "Importance of Genetic Diversity Assessment in Crop Plants and Its Recent Advances: An Overview of Its Analytical Perspectives". In: *Genetics Research International* 2015, p. 431487. ISSN: 20903162. DOI: 10.1155/2015/431487. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4383386/.

Grassini, P., K. Eskridge, and K. Cassman (2013). "Distinguishing between yield advances and yield plateaus in historical crop production trends". In: *Nature Communications* 4 (1), pp. 1–11. ISSN: 2041-1723. DOI: 10.1038/ncomms3918. URL: https://www.nature.com/articles/ncomms3918.

Graziosi, I. et al. (2016). "Emerging pests and diseases of South-east Asian cassava: a comprehensive evaluation of geographic priorities, management options and research needs". In: *Pest Management Science* 72 (6), pp. 1071–1089. ISSN: 1526-4998. DOI: 10.1002/PS.4250. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ps.4250.

Gu, D., K. Andreev, and M. Dupre (2021). "Major Trends in Population Growth Around the World". In: *China CDC Weekly* 3 (28), p. 604. ISSN: 2096-7071. DOI: 10.46234/CCDCW2021.160. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8393076/.

Gu, J. et al. (2018). "Recent advances in convolutional neural networks". In: *Pattern Recognition* 77, pp. 354–377. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2017.10.013.

Guehairia, O. et al. (2020). "Feature fusion via Deep Random Forest for facial age estimation". In: *Neural Networks* 130, pp. 238–252. DOI: 10.1016/j.neunet.2020.07.006. URL: https://doi.org/10.1016/j.neunet.2020.07.006.

Hall, D. et al. (2015). "Evaluation of features for leaf classification in challenging conditions". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 797–804. DOI: 10.1109/WACV.2015.111.

Han, L. et al. (2018). "A Clothes Classification Method Based on the gcForest". In: *2018 3rd IEEE International Conference on Image, Vision and Computing*, pp. 829–832. DOI: 10.1109/ICIVC.2018.8492801.

He, K. et al. (2015). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December, pp. 770–778. ISSN: 10636919. DOI: 10.48550/arxiv.1512.03385. URL: https://arxiv.org/abs/1512.03385v1.

Hecht-Nielsen, R. (1992). *Theory of the Backpropagation Neural Network*. Academic Press, pp. 65–93. DOI: 10.1016/B978-0-12-741252-8.50010-8.

Heck, M. (2018). "Insect Transmission of Plant Pathogens: a Systems Biology Perspective". In: *mSystems* 3 (2). ISSN: 23795077. DOI: 10.1128/MSYSTEMS.00168-17. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5881024/.

Hendrix, C. and S. Haggard (2015). "Global food prices, regime type, and urban unrest in the developing world". In: *Journal of Peace Research* 52 (2), pp. 143–157. ISSN: 14603578. DOI: 10.1177/0022343314561599. URL: https://journals.sagepub.com/doi/full/10.1177/0022343314561599?casa_token=niDWVha9TSsAAAAA%3A28Lb5klREcJGmmkTC_KFA4fkg6QNMhbKAezrrYC46n5IdNOG2gLYQruDw7yyRQM1okY1t-fgKzBn9w.

Hillocks, R., J. Thresh, and A. Bellotti (2002). *Cassava Biology, Production and Utilization*, pp. 1–9. ISBN: 0 85199 524 1.

Hillocks, R. et al. (2001). "Effects of Brown Streak Virus Disease on Yield and Quality of Cassava in Tanzania". In: *Journal of Phytopathology* 149 (7-8), pp. 389–394. ISSN: 1439-0434. DOI: 10.1111/J.1439-0434.2001.TB03868.X. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1439-0434.2001.tb03868.x.

Hillocks, R., J. Thresh, and R. Hillocks (2000). "Cassava mosaic and cassava brown streak virus diseases in Africa: A comparative guide to symptoms and aetiologies". In: *Roots* 7 (1), pp. 3–8. URL: https://www.researchgate.net/publication/238096673.

Hinton, G., S. Osindero, and Y.-W. Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural Computation* 18 (7), pp. 1527–1554. URL: https://doi.org/10.1162/neco.2006.18.7.1527.

Holloway, J. et al. (2019). "A Decision Tree Approach for Spatially Interpolating Missing Land Cover Data and Classifying Satellite Images". In: *Remote Sensing* 11 (15), p. 1796. ISSN: 2072-4292. DOI: 10.3390/RS11151796. URL: https://www.mdpi.com/2072-4292/11/15/1796.

Hothorn, T., K. Hornik, and A. Zeileis (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework". In: *Journal of Computational and Graphical Statistics* 15 (3), pp. 651–674.

Houngue, J. et al. (2019). "Evaluation of resistance to cassava mosaic disease in selected African cassava cultivars using combined molecular and greenhouse grafting tools". In: *Physiological and Molecular Plant Pathology* 105, pp. 47–53. ISSN: 0885-5765. DOI: 10.1016/J.PMPP.2018.07.003.

Howard, A. et al. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *arXiv*. DOI: 10.48550/arxiv.1704.04861. URL: https://arxiv.org/abs/1704.04861v1.

Hu, G. et al. (2018). "A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis". In: *Computers in Industry* 100, pp. 287–296. ISSN: 0166-3615. DOI: 10.1016/J.COMPIND.2018.04.002.

Huang, G. et al. (2017). "Densely Connected Convolutional Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269. URL: https://doi.org/10.1109/CVPR.2017.243.

Huang, W. et al. (2020). "Bacterial Vector-Borne Plant Diseases: Unanswered Questions and Future Directions". In: *Molecular Plant* 13 (10), pp. 1379–1393. ISSN: 1674-2052. DOI: 10.1016/J.MOLP.2020.08.010.

Hubel, D. and T. Wiesel (1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160 (1), pp. 106–154. ISSN: 1469-7793. DOI: 10.1113/JPHYSIOL.1962.SP006837. URL: https://onlinelibrary.wiley.com/doi/full/10.1113/jphysiol.1962.sp006837.

Huertas-Tato, J. et al. (2022). "Fusing CNNs and statistical indicators to improve image classification". In: *Information Fusion* 79, pp. 174–187. ISSN: 1566-2535. DOI: 10.1016/J.INFFUS.2021.09.012.

Hunter, M. et al. (2017). "Agriculture in 2050: Recalibrating Targets for Sustainable Intensification". In: *BioScience* 67 (4), pp. 386–391. ISSN: 0006-3568. DOI: 10.1093/BIOSCI/BIX010. URL: https://academic.oup.com/bioscience/article/67/4/386/3016049.

Iandola, F. et al. (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size". In: *arXiv*. DOI: 10.48550/arxiv.1602.07360. URL: https://arxiv.org/abs/1602.07360v4.

Iizumi, T. et al. (2017). "Responses of crop yield growth to global temperature and socioeconomic changes". In: *Scientific Reports 2017 7:1* 7 (1), pp. 1–10. ISSN: 2045-2322. DOI: 10.1038/s41598-017-08214-4. URL: https://www.nature.com/articles/s41598-017-08214-4.

Ijirshar, V. (2015). "The empirical analysis of agricultural exports and economic growth in Nigeria". In: *Journal of Development and Agricultural Economics* 7 (3), pp. 113–122. ISSN: 2006-9774. DOI: 10.5897/JDAE2014.0615. URL: https://academicjournals.org/journal/JDAE/article-abstract/C40062250615.

Irz, X. et al. (2001). "Agricultural Productivity Growth and Poverty Alleviation". In: *Development Policy Review* 19 (4), pp. 449–466.

Jansson, C. et al. (2009). "Cassava, a potential biofuel crop in China". In: *Applied Energy* 86 (SUPPL. 1). ISSN: 03062619. DOI: 10.1016/J.APENERGY.2009.05.011.

Japkowicz, N. and S. Stephen (2002). "The class imbalance problem: A systematic study". In: *Intelligent Data Analysis* 6 (5), pp. 429–449. DOI: 10.5555/1293951.1293954. URL: https://dl.acm.org/doi/10.5555/1293951.1293954.

Jarvis, A. et al. (2012). "Is Cassava the Answer to African Climate Change Adaptation?" In: *Tropical Plant Biology* 5 (1), pp. 9–29. ISSN: 19359756. DOI: 10.1007/S12042-012-9096-7. URL: https://link.springer.com/article/10.1007/s12042-012-9096-7#:~:text=Generally%2C%20cassava%20reacted%20very%20well,West%20Africa%20and%20the%20Sahel)..

Jayne, T., J. Chamberlin, and D. Headey (2014). "Land pressures, the evolution of farming systems, and development strategies in Africa: A synthesis". In: *Food Policy* 48, pp. 1–17. ISSN: 0306-9192. DOI: 10.1016/J.FOODPOL.2014.05.014.

Jenifa, A., R. Ramalakshmi, and V. Ramachandran (2019). "Cotton Leaf Disease Classification using Deep Convolution Neural Network for Sustainable Cotton Production". In: *2019 International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development, INCCES 2019*. DOI: 10.1109/INCCES47820.2019.9167715.

Jhonnerie, R. et al. (2015). "Random Forest Classification for Mangrove Land Cover Mapping Using Landsat 5 TM and Alos Palsar Imageries". In: *Procedia Environmental Sciences* 24, pp. 215–221. ISSN: 1878-0296. DOI: 10.1016/J.PROENV.2015.03.028.

Jiang, Y. et al. (2019). "Challenging battles of plants with phloem-feeding insects and prokaryotic pathogens". In: *Proceedings of the National Academy of Sciences of the United States of America* 116 (47), pp. 23390–23397. ISSN: 10916490. DOI: 10.1073/PNAS.1915396116. URL: www.pnas.org/cgi/doi/10.1073/pnas.1915396116.

Jiao, W., X. Hao, and C. Qin (2021). "The Image Classification Method with CNN-XGBoost Model Based on Adaptive Particle Swarm Optimization". In: *Information* 12 (4), p. 156. ISSN: 2078-2489. DOI: 10.3390/INFO12040156. URL: https://www.mdpi.com/2078-2489/12/4/156.

Johnson, D. (1997). "Agriculture and the Wealth of Nations". In: *The American Economic Review* 87 (2), pp. 1–12. URL: https://www.jstor.org/stable/2950874.

Jones, D. (2003). "Plant Viruses Transmitted by Whiteflies". In: *European Journal of Plant Pathology 2003 109:3* 109 (3), pp. 195–219. ISSN: 1573-8469. DOI: 10.1023/A:1022846630513. URL: https://link.springer.com/article/10.1023/A:1022846630513.

Jones, J. and J. Dangl (2006). "The plant immune system". In: *Nature* 444 (7117), pp. 323–329. ISSN: 1476-4687. DOI: 10.1038/nature05286. URL: https://www.nature.com/articles/nature05286.

Jones, R. (2021). "Global Plant Virus Disease Pandemics and Epidemics". In: *Plants* 10 (2), pp. 1–41. ISSN: 22237747. DOI: 10.3390/PLANTS10020233. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7911862/.

Jones, R., M. Goodin, and J. Verchot (2020). "Disease Pandemics and Major Epidemics Arising from New Encounters between Indigenous Viruses and Introduced Crops". In: *Viruses* 12 (12), p. 1388. ISSN: 1999-4915. DOI: 10.3390/V12121388. URL: https://www.mdpi.com/1999-4915/12/12/1388.

Jones, W. (1959). *Manioc in Africa*. Vol. 1. Stanford University Press, p. 315. ISBN: 0804700028, 9780804700023. URL: https://www.cabdirect.org/cabdirect/abstract/19611403361.

Joshi, A. et al. (2011). "Delivering rust resistant wheat to farmers: A step towards increased food security". In: *Euphytica* 179 (1), pp. 187–196. ISSN: 00142336. DOI: 10.1007/S10681-010-0314-9. URL: https://link.springer.com/article/10.1007/s10681-010-0314-9.

jun Du, P., K. Tan, and H. jun Su (2009). "Feature extraction for target identification and image classification of OMIS hyperspectral image". In: *Mining Science and Technology (China)* 19 (6), pp. 835–841. ISSN: 1674-5264. DOI: 10.1016/S1674-5264(09)60152-6.

Kalyebi, A. et al. (2018). "African cassava whitefly, Bemisia tabaci, cassava colonization preferences and control implications". In: *PLoS ONE* 13 (10), e0204862. ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0204862. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6177144/.

Karlström, A. et al. (2016). "Biological implications in cassava for the production of amylose-free starch: Impact on root yield and related traits". In: *Frontiers in Plant Science* 7 (MAY2016), p. 604. ISSN: 1664462X. DOI: 10.3389/FPLS.2016.00604/BIBTEX.

Kass, G. (1980). "An Exploratory Technique for Investigating Large Quantities of Categorical Data". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 29 (2), pp. 119–127. URL: https://www.jstor.org/stable/2986296.

Kawabata, M. et al. (2020). "Food security and nutrition challenges in Tajikistan: Opportunities for a systems approach". In: *Food Policy* 96. ISSN: 03069192. DOI: 10.1016/J.FOODPOL.2020.101872. URL: https://doi.org/10.1016/j.foodpol.2020.101872.

Khan, M. and S. Kwon (2019). "A CNN-Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition". In: *Sensors* 20 (1), p. 183. ISSN: 1424-8220. DOI: 10.3390/S20010183. URL: https://www.mdpi.com/1424-8220/20/1/183.

Khan, R., A. Hanbury, and J. Stoettinger (2010). "Skin detection: A random forest approach". In: *International Conference on Image Processing (ICIP)*, pp. 4613–4616. ISSN: 15224880. DOI: 10.1109/ICIP.2010.5651638.

Kim, G. et al. (2018). "Comparison of Shallow and Deep Learning Methods on Classifying the Regional Pattern of Diffuse Lung Disease". In: *Journal of Digital Imaging* 31 (4), p. 415. ISSN: 1618727X. DOI: 10.1007/S10278-017-0028-9. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6113148/.

Kim, H. and W.-Y. Loh (2001). "Classification Trees with Unbiased Multiway Splits". In: *Journal of the American Statistical Association* 96 (454), pp. 589–604.

Kingma, D. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. DOI: 10.48550/arxiv.1412.6980. URL: https://arxiv.org/abs/1412.6980v9.

Kintché, K. et al. (2017). "Cassava yield loss in farmer fields was mainly caused by low soil fertility and suboptimal management practices in two provinces of the Democratic Republic of Congo". In: *European Journal of Agronomy* 89, pp. 107–123. ISSN: 1161-0301. DOI: 10.1016/J.EJA.2017.06.011.

Ko, B., S. Kim, and J. Nam (2011). "X-ray Image Classification Using Random Forests with Local Wavelet-Based CS-Local Binary Patterns". In: *Journal of Digital Imaging* 24 (6), p. 1141. ISSN: 08971889. DOI: 10.1007/S10278-011-9380-3. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3222545/.

Krizhevsky, A., I. Sutskever, and G. Hinton (2012). "ImageNet classification with deep convolutional neural networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 1097–1105. ISSN: 15577317. DOI: 10.1145/3065386. URL: https://dl.acm.org/doi/abs/10.1145/3065386.

Lau, H. and J. Botella (2017). "Advanced DNA-based point-of-care diagnostic methods for plant diseases detection". In: *Frontiers in Plant Science* 8, p. 2016. ISSN: 1664462X. DOI: 10.3389/FPLS.2017.02016.

LeCun, Y. et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1 (4), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/NECO.1989.1.4.541.

LeCun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86 (11), pp. 2278–2323. ISSN: 00189219. DOI: 10.1109/5.726791.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". In: *Nature* 521 (7553), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539.

Lee, J. et al. (2020). "Reliable solar irradiance prediction using ensemble learning-based models: A comparative study". In: *Energy Conversion and Management* 208, p. 112582. ISSN: 0196-8904. DOI: 10.1016/J.ENCONMAN.2020.112582.

Legg, J. and C. Fauquet (2004). "Cassava mosaic geminiviruses in Africa". In: *Plant Molecular Biology 2004 56:4* 56 (4), pp. 585–599. ISSN: 1573-5028. DOI: 10.1007/S11103-004-1651-7. URL: https://link.springer.com/article/10.1007/s11103-004-1651-7.

Legg, J. and S. Ogwal (1998). "Changes in the incidence of African cassava mosaic virus disease and the abundance of its whitefly vector along south–north transects in Uganda". In: *Journal of Applied Entomology* 122 (1-5), pp. 169–178. ISSN: 1439-0418. DOI: 10.1111/J.1439-0418.1998.TB01480.X. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1439-0418.1998.tb01480.x.

Legg, J. et al. (2001). "Spread into Rwanda of the severe cassava mosaic virus disease pandemic and the associated Uganda variant of East African cassava mosaic virus (EACMV-Ug)". In: *Plant Pathology* 50 (6), pp. 796–796. ISSN: 1365-3059. DOI: 10.1046/J.1365-3059.2001.00619.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1046/j.1365-3059.2001.00619.x.

Legg, J. et al. (2011). "Comparing the regional epidemiology of the cassava mosaic and cassava brown streak virus pandemics in Africa". In: *Virus Research* 159 (2), pp. 161–170. ISSN: 01681702. DOI: 10.1016/J.VIRUSRES.2011.04.018.

Legg, J. et al. (2015). "Cassava Virus Diseases: Biology, Epidemiology, and Management". In: *Advances in Virus Research* 91 (1), pp. 85–142. ISSN: 0065-3527. DOI: 10.1016/BS.AIVIR.2014.10.001.

Li, J. et al. (2018). "Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion". In: *Sensors (Basel, Switzerland)* 18 (7). ISSN: 14248220. DOI: 10.3390/S18072080. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6068932/.

Li, M. et al. (2017). "Hyperspectral Image Classification Based on Deep Forest and Spectral-Spatial Cooperative Feature". In: *International Conference on Image and Graphics (ICIG)* 10668, pp. 325–336. ISSN: 16113349. DOI: 10.1007/978-3-319-71598-8_29. URL: https://link.springer.com/chapter/10.1007/978-3-319-71598-8_29.

Li, Y., J. Nie, and X. Chao (2020). "Do we really need deep CNN for plant diseases identification?" In: *Computers and Electronics in Agriculture* 178, p. 105803. ISSN: 0168-1699. DOI: 10.1016/J.COMPAG.2020.105803.

Liang, H. et al. (2017). "Text feature extraction based on deep learning: a review". In: *EURASIP Journal on Wireless Communications and Networking* 2017 (1), pp. 1–12. ISSN: 16871499. DOI: 10.1186/S13638-017-0993-1. URL: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-017-0993-1.

Little, T. et al. (2010). "The Coevolution of Virulence: Tolerance in Perspective". In: *PLOS Pathogens* 6 (9), e1001006. ISSN: 1553-7374. DOI: 10.1371/JOURNAL.PPAT.1001006. URL: https://journals.plos.org/plospathogens/article?id=10.1371/journal.ppat.1001006.

Liu, F. and A. Yang (2019). "Application of gcForest to visual tracking using UAV image sequences". In: *Multimedia Tools and Applications* 78.19, pp. 27933–27956. ISSN: 15737721. DOI: 10.1007/S11042-019-07864-Y. URL: https://link.springer.com/article/10.1007/s11042-019-07864-y.

Liu, H. et al. (2021a). "Small sample color fundus image quality assessment based on gcforest". In: *Multimedia Tools and Applications* 80.11, pp. 17441–17459. ISSN: 15737721. DOI: 10.1007/S11042-020-09362-Y. URL: https://link.springer.com/article/10.1007/s11042-020-09362-y.

Liu, J. and X. Wang (2021). "Plant diseases and pests detection based on deep learning: a review". In: *Plant Methods* 17 (1). ISSN: 17464811. DOI: 10.1186/S13007-021-00722-9.

Liu, J. et al. (2008). "A spatially explicit assessment of current and future hotspots of hunger in Sub-Saharan Africa in the context of global change". In: *Global and Planetary Change* 64 (3-4), pp. 222–235. ISSN: 0921-8181. DOI: 10.1016/J.GLOPLACHA.2008.09.007.

Liu, K. et al. (2021b). "An intelligent fault diagnosis method for transformer based on IPSO-gcForest". In: *Mathematical Problems in Engineering* 2021. ISSN: 15635147. DOI: 10.1155/2021/6610338.

Liu, Q. et al. (2019a). "Gcforest-Based Fault Diagnosis Method for Rolling Bearing". In: *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018*, pp. 572–577. DOI: 10.1109/PHM-CHONGQING.2018.00103.

Liu, X. et al. (2019b). "Deep Multigrained Cascade Forest for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 57 (10), pp. 8169–8183. ISSN: 15580644. DOI: 10.1109/TGRS.2019.2918587.

Liu, X. et al. (2019c). "Emotion Recognition Based on Multi-Composition Deep Forest and Transferred Convolutional Neural Network". In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 23.5, pp. 883–890. ISSN: 18838014. DOI: 10.20965/JACIII.2019.P0883.

Loh, W. (2009). "Improving the precision of classification trees". In: *The Annals of Applied Statistics* 3 (4), pp. 1710–1737. DOI: 10.1214/09-AOAS260.

Loh, W. and Y. Shih (1997). "Split Selection Methods For Classification Trees". In: *Statistica Sinica* 7, pp. 815–840.

Loh, W. and N. Vanichsetakul (1988). "Tree-Structured Classification Via Generalised Discriminant Analysis". In: *Journal of the American Statistical Association* 83 (403), pp. 715–725.

Long, J., E. Shelhamer, and T. Darrell (2015). "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Lötschert, W. and G. Beese (1983). *Collins Guide to Tropical Plants*. Vol. 1. Collins. ISBN: 0002191121.

Lowder, S., M. Sánchez, and R. Bertini (2021). "Which farms feed the world and has farmland become more concentrated?" In: *World Development* 142, p. 105455. ISSN: 0305-750X. DOI: 10.1016/J.WORLDDEV.2021.105455.

Lu, H. et al. (2019). "A 3D Convolutional Neural Network for Volumetric Image Semantic Segmentation". In: *Procedia Manufacturing* 39, pp. 422–428. ISSN: 2351-9789. DOI: 10.1016/J.PROMFG.2020.01.386.

Léotard, G. et al. (2009). "Phylogeography and the Origin of Cassava: New Insights from the Northern Rim of the Amazonian Basin". In: *Molecular Phylogenetics and Evolution* 53, pp. 329–334. DOI: 10.1016/j.ympev.2009.05.003.

López, C. and A. Bernal (2012). "Cassava Bacterial Blight: Using Genomics for the Elucidation and Management of an Old Problem". In: *Tropical Plant Biology* 5 (1), pp. 117–126. ISSN: 1935-9764. DOI: 10.1007/S12042-011-9092-3. URL: https://link.springer.com/article/10.1007/s12042-011-9092-3.

López-Villavicencio, M. et al. (2011). "Competition, Cooperation Among Kin, and Virulence in Multiple Infections". In: *Evolution* 65 (5), pp. 1357–1366. ISSN: 1558-5646. DOI: 10.1111/J.1558-5646.2010.01207.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1558-5646.2010.01207.x.

Ma, P. et al. (2022). "DBC-Forest: Deep forest with binning confidence screening". In: *Neurocomputing* 475, pp. 112–122. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2021.12.075.

Ma, W. et al. (2019). "Change Detection Based on Multi-Grained Cascade Forest and Multi-Scale Fusion for SAR Images". In: *Remote Sensing* 11.2, p. 142. ISSN: 2072-4292. DOI: 10.3390/RS11020142. URL: https://www.mdpi.com/2072-4292/11/2/142.

MacFadyen, S. et al. (2018). "Cassava whitefly, Bemisia tabaci (Gennadius) (Hemiptera: Aleyrodidae) in East African farming landscapes: a review of the factors determining abundance". In: *Bulletin of Entomological Research* 108 (5), pp. 565–582. ISSN: 0007-4853. DOI:

10.1017/S0007485318000032. URL: https://www.cambridge.org/core/journals/bulletin-of-entomological-research/article/cassava-whitefly-bemisia-tabaci-gennadius-hemiptera-aleyrodidae-in-east-african-farming-landscapes-a-review-of-the-factors-determining-abundance/D43B1B34908112063495739456B6E98A.

Maruthi, M. et al. (2005). "Transmission of Cassava brown streak virus by Bemisia tabaci (Gennadius)". In: *Journal of Phytopathology* 153 (5), pp. 307–312. ISSN: 1439-0434. DOI: 10.1111/J.1439-0434.2005.00974.X. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1439-0434.2005.00974.x.

Maruthi, M. et al. (2017). "The role of the whitefly, Bemisia tabaci (Gennadius), and farmer practices in the spread of cassava brown streak ipomoviruses". In: *Journal of Phytopathology* 165 (11-12), pp. 707–717. ISSN: 1439-0434. DOI: 10.1111/JPH.12609. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/jph.12609.

Maryum, A., M. Akram, and A. Salam (2021). "Cassava Leaf Disease Classification using Deep Neural Networks". In: pp. 32–37. DOI: 10.1109/HONET53078.2021.9615488.

Mathur, A., S. Das, and S. Sircar (2006). "Status of Agriculture in India: Trends and Prospects". In: *Economic and Political Weekly* 41 (52), pp. 5327–5336. URL: https://www.jstor.org/stable/4419078.

Mauck, K. et al. (2012). "Transmission mechanisms shape pathogen effects on host–vector interactions: evidence from plant viruses". In: *Functional Ecology* 26 (5), pp. 1162–1175. ISSN: 1365-2435. DOI: 10.1111/J.1365-2435.2012.02026.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1365-2435.2012.02026.x.

Mbanjo, E. et al. (2021). "Technological Innovations for Improving Cassava Production in Sub-Saharan Africa". In: *Frontiers in Genetics* 11, p. 1829. ISSN: 16648021. DOI: 10.3389/FGENE.2020.623736.

McCallum, E., R. Anjanappa, and W. Gruissem (2017). "Tackling agriculturally relevant diseases in the staple crop cassava (Manihot esculenta)". In: *Current Opinion in Plant Biology* 38, pp. 50–58. ISSN: 1369-5266. DOI: 10.1016/J.PBI.2017.04.008.

Mechiche-Alami, A. and A. Abdi (2020). "Agricultural productivity in relation to climate and cropland management in West Africa". In: *Scientific Reports* 10 (1), pp. 1–10. ISSN: 2045-2322. DOI: 10.1038/s41598-020-59943-y. URL: https://www.nature.com/articles/s41598-020-59943-y.

Mello Prado, R. de (2021). *Visual and Leaf Diagnosis*. Springer, Cham, pp. 279–312. DOI: 10.1007/978-3-030-71262-4_19. URL: https://link.springer.com/book/10.1007/978-3-030-71262-4.

Messenger, R. and L. Mandell (1972). "A Modal Search Technique for Predictibe Nominal Scale Multivariate Analys". In: *Source: Journal of the American Statistical Association* 67 (340), pp. 768–772.

Miao, X. et al. (2011). "Applying tree-based ensemble algorithms to the classification of ecological zones using multi-temporal multi-source remote-sensing data". In: *International Journal of Remote Sensing* 33 (6), pp. 1823–1849. ISSN: 13665901. DOI: 10.1080/01431161.2011.602651. URL: https://www.tandfonline.com/doi/abs/10.1080/01431161.2011.602651.

Mikaberidze, A. and B. McDonald (2020). "A tradeoff between tolerance and resistance to a major fungal pathogen in elite wheat cultivars". In: *New Phytologist* 226 (3), pp. 879–890. ISSN: 1469-8137. DOI: 10.1111/NPH.16418. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/nph.16418.

Miller, S., F. Beed, and C. Harmon (2009). "Plant Disease Diagnostic Capabilities and Networks". In: *Annual Review of Phytopathology* 47, pp. 15–38. ISSN: 00664286. DOI: 10.1146/ANNUREV-PHYTO-080508-081743. URL: https://www.annualreviews.org/doi/abs/10.1146/annurev-phyto-080508-081743.

Mojjada, R. et al. (2020). "Detection of plant leaf disease using digital image processing". In: *Materials Today: Proceedings*. ISSN: 2214-7853. DOI: 10.1016/J.MATPR.2020.11.115.

Morgan, J. and J. Sonquist (1963). "Problems in the Analysis of Survey Data, and a Proposal". In: *Journal of the American Statistical Association* 58 (302), pp. 415–434. DOI: 10.1080/01621459.1963.10500855.

Moyer, J. and D. Bohl (2019). "Alternative pathways to human development: Assessing trade-offs and synergies in achieving the Sustainable Development Goals". In: *Futures* 105, pp. 199–210. ISSN: 0016-3287. DOI: 10.1016/J.FUTURES.2018.10.007.

Mtunguja, M. et al. (2019). "Opportunities to commercialize cassava production for poverty alleviation and improved food security in Tanzania". In: *African Journal of Food, Agriculture, Nutrition and Development* 19 (1), pp. 13928–13946. ISSN: 1684-5358. DOI: 10.4314/ajfand.v19i1.. URL: https://www.ajol.info/index.php/ajfand/article/view/185568.

Mumford, R., R. Macarthur, and N. Boonham (2015). "The role and challenges of new diagnostic technology in plant biosecurity". In: *Food Security* 8 (1), pp. 103–109. ISSN: 1876-4525. DOI: 10.1007/S12571-015-0533-Y. URL: https://link.springer.com/article/10.1007/s12571-015-0533-y.

Mwatuni, F. et al. (2015). "Distribution of Cassava Mosaic Geminiviruses and their Associated DNA Satellites in Kenya". In: *Journal of Experimental Agriculture International* 9 (3), pp. 1–12. ISSN: 2457-0591. DOI: 10.9734/AJEA/2015/18473. URL: https://journaljeai.com/index.php/JEAI/article/view/3403.

Mwebaze, E. and G. Owomugisha (2017). "Machine Learning for Plant Disease Incidence and Severity Measurements from Leaf Images". In: *2016 15th IEEE International Conference on Machine Learning and Applications*, pp. 158–163. DOI: 10.1109/ICMLA.2016.0034.

Mwebaze, E. et al. (2019). "iCassava 2019 Fine-Grained Visual Categorization Challenge". In: *arXiv*. DOI: 10.48550/arxiv.1908.02900. URL: https://arxiv.org/abs/1908.02900v2.

Nair, V. and G. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814.

Najafabadi, M. et al. (2015). "Deep learning applications and challenges in big data analytics". In: *Journal of Big Data* 2 (1), pp. 1–21. ISSN: 21961115. DOI: 10.1186/S40537-014-0007-7. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-014-0007-7.

Nazarov, P. et al. (2020). "Infectious Plant Diseases: Etiology, Current Status, Problems and Prospects in Plant Protection". In: *Acta Naturae* 12 (3), pp. 46–59. ISSN: 20758251. DOI: 10.32607/ACTANATURAE.11026.

Neuenschwander, P. et al. (2002). "Occurrence of the Uganda variant of East African cassava mosaic virus (EACMV-Ug) in western Democratic Republic of Congo and the Congo Republic defines the westernmost extent of the CMD pandemic in East/Central Africa". In: *Plant Pathology* 51 (3), pp. 385–385. ISSN: 1365-3059. DOI: 10.1046/J.1365-3059.2002.00698.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1046/j.1365-3059.2002.00698.x.

Nguyen, T., S. Gheewala, and S. Garivait (2007). "Full Chain Energy Analysis of Fuel Ethanol from Cassava in Thailand". In: *Environmental Science and Technology* 41, pp. 4135–4142. DOI: 10.1021/es0620641. URL: https://pubs.acs.org/sharingguidelines.

Nicaise, V. (2014). "Crop immunity against viruses: Outcomes and future challenges". In: *Frontiers in Plant Science* 5, p. 660. ISSN: 1664462X. DOI: 10.3389/FPLS.2014.00660.

Nicolopoulou-Stamati, P. et al. (2016). "Chemical Pesticides and Human Health: The Urgent Need for a New Concept in Agriculture". In: *Frontiers in Public Health* 4, p. 148. ISSN: 22962565. DOI: 10.3389/FPUBH.2016.00148.

Nidhis, A. et al. (2019). *Cluster Based Paddy Leaf Disease Detection, Classification and Diagnosis in Crop Health Monitoring Unit*. Ed. by J. Peter et al. Vol. 31. Springer, Cham, pp. 281–291. DOI: 10.1007/978-3-030-04061-1_29. URL: https://link.springer.com/chapter/10.1007/978-3-030-04061-1_29.

Nie, X. et al. (2021). "Online Multiview Deep Forest for Remote Sensing Image Classification via Data Fusion". In: *IEEE Geoscience and Remote Sensing Letters* 18 (8), pp. 1456–1460. ISSN: 15580571. DOI: 10.1109/LGRS.2020.3002848.

Niehl, A. et al. (2016). "Double-stranded RNAs induce a pattern-triggered immune signaling pathway in plants". In: *The New Phytologist* 211 (3), pp. 1008–1019. ISSN: 1469-8137. DOI: 10.1111/NPH.13944. URL: https://pubmed.ncbi.nlm.nih.gov/27030513/.

Nishad, R. et al. (2020). "Modulation of Plant Defense System in Response to Microbial Interactions". In: *Frontiers in Microbiology* 11, p. 1298. ISSN: 1664302X. DOI: 10.3389/FMICB.2020.01298.

Njoroge, M. et al. (2017). "Whitefly species efficiency in transmitting cassava mosaic and brown streak virus diseases". In: *Cogent Biology* 3 (1), p. 1311499. ISSN: 2331-2025. DOI: 10.1080/23312025.2017.1311499. URL: https://www.tandfonline.com/doi/abs/10.1080/23312025.2017.1311499.

Nwezeobi, J. et al. (2020). "Cassava whitefly species in eastern Nigeria and the threat of vector-borne pandemics from East and Central Africa". In: *PLoS ONE* 15 (5), e0232616. ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0232616. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7205266/.

Oerke, E. and H. Dehne (2004). "Safeguarding production—losses in major crops and the role of crop protection". In: *Crop Protection* 23 (4), pp. 275–285. ISSN: 0261-2194. DOI: 10.1016/J.CROPRO.2003.10.001.

Oldfield, E., M. Bradford, and S. Wood (2019). "Global meta-analysis of the relationship between soil organic matter and crop yields". In: *SOIL* 5 (1), pp. 15–32. ISSN: 2199398X. DOI: 10.5194/SOIL-5-15-2019.

Olsen, K. (2004). "SNPs, SSRs and inferences on cassava's origin". In: *Plant Molecular Biology* 56 (4), pp. 517–526. ISSN: 0167-4412. DOI: 10.1007/S11103-004-5043-9. URL: https://pubmed.ncbi.nlm.nih.gov/15630616/.

Ossama, A. et al. (2014). "Convolutional Neural Networks for Speech Recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22 (10). DOI: 10.1109/TASLP.2014.2339736. URL: http://www.ieee.org/publications_standards/publications/rights/index.html.

Otsuka, K., Y. Nakano, and K. Takahashi (2016). "Contract Farming in Developed and Developing Countries". In: *Annual Review of Resource Economics* 8 (1), pp. 353–376. ISSN: 19411359. DOI: 10.1146/ANNUREV-RESOURCE-100815-095459. URL: https://www.annualreviews.org/doi/abs/10.1146/annurev-resource-100815-095459.

Pagán, I. and F. García-Arenal (2018). "Tolerance to Plant Pathogens: Theory and Experimental Evidence". In: *International journal of molecular sciences* 19 (3). ISSN: 1422-0067. DOI: 10.3390/IJMS19030810. URL: https://pubmed.ncbi.nlm.nih.gov/29534493/.

Pang, M. et al. (2018). "Improving Deep Forest by Confidence Screening". In: *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 1194–1199. ISSN: 15504786. DOI: 10.1109/ICDM.2018.00158.

Pathak, A., M. Pandey, and S. Rautaray (2018). "Application of Deep Learning for Object Detection". In: *Procedia Computer Science* 132, pp. 1706–1717. ISSN: 1877-0509. DOI: 10.1016/J.PROCS.2018.05.144.

Patil, B. et al. (2015). "Cassava brown streak disease: a threat to food security in Africa". In: *Journal of General Virology* 96, pp. 956–968. DOI: 10.1099/jgv.0.000014. URL: http://vir.sgmjournals.org.

Paudel, D. and H. Sanfaçon (2018). "Exploring the diversity of mechanisms associated with plant tolerance to virus infection". In: *Frontiers in Plant Science* 871, p. 1575. ISSN: 1664462X. DOI: 10.3389/FPLS.2018.01575.

Perilla-Henao, L. and C. Casteel (2016). "Vector-borne bacterial plant pathogens: Interactions with hemipteran insects and plants". In: *Frontiers in Plant Science* 7 (AUG2016), p. 1163. ISSN: 1664462X. DOI: 10.3389/FPLS.2016.01163.

Pita, J. et al. (2001). "Recombination, pseudorecombination and synergism of geminiviruses are determinant keys to the epidemic of severe cassava mosaic disease in Uganda". In: *Journal of General Virology* 82, pp. 655–665. DOI: 10.1099/0022-1317-82-3-655.

Prechelt, L. (2012). *Early stopping - But when?* Ed. by M. G., O. G.B., and K. Müller. Vol. 7700. Springer, pp. 53–67. ISBN: 9783642352881. DOI: 10.1007/978-3-642-35289-8_5. URL: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_5.

Quinlan, J. (1986). "Induction of Decision Trees". In: *Machine Learning* 1 (1), pp. 81–106. ISSN: 1573-0565. DOI: 10.1023/A:1022643204877. URL: https://link.springer.com/article/10.1007/BF00116251.

— (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc. ISBN: 1-55860-238-0. URL: https://www.sciencedirect.com/book/9780080500584/c4-5.

Ramcharan, A. et al. (2017). "Deep learning for image-based cassava disease detection". In: *Frontiers in Plant Science* 8, p. 1852. ISSN: 1664462X. DOI: 10.3389/FPLS.2017.01852.

Ravi, V., V. Acharya, and T. D. Pham (2022). "Attention deep learning-based large-scale learning classifier for Cassava leaf disease classification". In: *Expert Systems* 39.2. ISSN: 14680394. DOI: 10.1111/EXSY.12862.

Ren, S. et al. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *International Conference on Neural Information Processing Systems*, pp. 91–99. URL: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed3Paper.pdf.

Ren, T., Y. Zhang, and C. Wang (2019). "Identification of Corn Leaf Disease Based on Image Processing". In: *2019 2nd International Conference on Information Systems and Computer Aided Education, ICISCAE 2019*, pp. 165–168. DOI: 10.1109/ICISCAE48440.2019.221610.

Ricciardi, V. et al. (2018). "How much of the world's food do smallholders produce?" In: *Global Food Security* 17, pp. 64–72. ISSN: 2211-9124. DOI: 10.1016/J.GFS.2018.05.002.

Rodoni, B. (2009). "The role of plant biosecurity in preventing and controlling emerging plant virus disease epidemics". In: *Virus Research* 141 (2), pp. 150–157. ISSN: 0168-1702. DOI: 10.1016/J.VIRUSRES.2008.11.019.

Rogers, D. and S. Appan (1973). "Manihot Manihotoides (Euphorbiaceae)". In: *Flora Neotropica* 13, pp. 1–272. URL: https://www.jstor.org/stable/4393691.

Rokach, L. and O. Maimon (2005). *Decision Trees*. Ed. by O. Maimon and L. Rokach. Springer, pp. 165–192. DOI: 10.1007/0-387-25465-X_9. URL: https://link.springer.com/chapter/10.1007/0-387-25465-X_9.

Ronneberger, O., P. Fischer, and T. Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention* 9351, pp. 234–241. ISSN: 16113349. DOI: 10.1007/978-3-319-24574-4_28. URL: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

Rosenblatt, F. (1960). "Perceptron Simulation Experiments". In: *Proceedings of the IRE* 48 (3), pp. 301–309. ISSN: 00968390. DOI: 10.1109/JRPROC.1960.287598.

Rumelhart, D., G. Hinton, and R. Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323 (6088), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: https://www.nature.com/articles/323533a0.

Russakovsky, O. et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115 (3), pp. 211–252. ISSN: 15731405. DOI: 10.1007/

S11263-015-0816-Y. URL: https://link.springer.com/article/10.1007/s11263-015-0816-y.

Sambasivam, G. and G. Opiyo (2021). "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks". In: *Egyptian Informatics Journal* 22 (1), pp. 27–34. ISSN: 1110-8665. DOI: 10.1016/J.EIJ.2020.02.007.

Sanchez, P. (2002). "Soil fertility and hunger in Africa". In: *Science* 295 (5562), pp. 2019–2020. ISSN: 00368075. DOI: 10.1126/SCIENCE.1065256.

Sandler, M. et al. (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. ISSN: 10636919. DOI: 10.48550/arxiv.1801.04381. URL: https://arxiv.org/abs/1801.04381v4.

Sarica, A., A. Cerasa, and A. Quattrone (2017). "Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: A systematic review". In: *Frontiers in Aging Neuroscience* 9 (OCT), p. 329. ISSN: 16634365. DOI: 10.3389/FNAGI.2017.00329.

Sarker, I. (2021). "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions". In: *SN Computer Science* 2 (6), pp. 1–20. ISSN: 26618907. DOI: 10.1007/S42979-021-00815-1. URL: https://link.springer.com/article/10.1007/s42979-021-00815-1.

Sastry, K. and T. Zitter (2014). *Plant virus and viroid diseases in the tropics: Volume 2: Epidemiology and management*. Ed. by K. S. Sastry and T. A. Zitter. 1st ed. Vol. 2. Springer Netherlands, pp. 149–165. ISBN: 9789400778207. DOI: 10.1007/978-94-007-7820-7.

Savary, S. et al. (2006). "Quantification and Modeling of Crop Losses: A Review of Purposes". In: *Annual Review of Phytopathology* 44 (1), pp. 89–112. DOI: 10.1146/annurev.phyto.44.070505.143342. URL: www.annualreviews.org.

Savary, S. et al. (2019). "The global burden of pathogens and pests on major food crops". In: *Nature Ecology & Evolution* 3 (3), pp. 430–439. ISSN: 2397-334X. DOI: 10.1038/s41559-018-0793-y. URL: https://www.nature.com/articles/s41559-018-0793-y.

Savatin, D. et al. (2014). "Wounding in the plant tissue: the defense of a dangerous passage". In: *Frontiers in Plant Science* 5 (SEP). ISSN: 1664462X. DOI: 10.3389/FPLS.2014.00470. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4165286/.

Shackelford, G. et al. (2018). "Cassava farming practices and their agricultural and environmental impacts: A systematic map protocol". In: *Environmental Evidence* 7 (1), pp. 1–7. ISSN: 20472382. DOI: 10.1186/S13750-018-0142-2. URL: https://environmentalevidencejournal.biomedcentral.com/articles/10.1186/s13750-018-0142-2.

Shen, D., G. Wu, and H. Suk (2017). "Deep Learning in Medical Image Analysis". In: *Annual Review of Biomedical Engineering* 19, p. 221. ISSN: 15454274. DOI: 10.1146/ANNUREV-BIOENG-071516-044442. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5479722/.

Shen, F., J. Liu, and P. Wu (2020). "Low-Resolution Facial Expression Recognition Based on Texture Mapping-Based GcForest". In: *Proceedings of 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2020*, pp. 289–294. DOI: 10.1109/ICCASIT50869.2020.9368700.

Shin, H. et al. (2016). "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning". In: *IEEE Transactions on Medical Imaging* 35 (5), p. 1285. ISSN: 1558254X. DOI: 10.1109/TMI.2016.2528162. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4890616/.

Simonyan, K. and A. Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. DOI: 10.48550/arxiv.1409.1556. URL: https://arxiv.org/abs/1409.1556v6.

Smith, P. et al. (2010). "Competition for land". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 365 (1554), pp. 2941–2957. ISSN: 14712970. DOI: 10.1098/RSTB.2010.0127. URL: https://royalsocietypublishing.org/doi/full/10.1098/rstb.2010.0127.

Song, Y. and Y. Lu (2015). "Decision tree methods: applications for classification and prediction". In: *Shanghai Archives of Psychiatry* 27 (2), p. 130. ISSN: 10020829. DOI: 10.11919/J.ISSN.1002-0829.215044. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/.

Srivastava, N. et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958. ISSN: 1532-4435.

Sseruwagi, P. et al. (2004). "Methods of surveying the incidence and severity of cassava mosaic disease and whitefly vector populations on cassava in Africa: a review". In: *Virus Research* 100 (1), pp. 129–142. ISSN: 0168-1702. DOI: 10.1016/J.VIRUSRES.2003.12.021.

Storey, H. (1936). "Virus diseases of East African plants. VI. A progress report on studies of disease of cassava." In: *East African Agricultural Journal* 2, p. 39.

Strobl, C., J. Malley, and G. Tutz (2009). "An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests". In: *Psychological methods* 14 (4), p. 323. ISSN: 1082989X. DOI: 10.1037/A0016973. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2927982/.

Sultan, B. and M. Gaetani (2016). "Agriculture in West Africa in the twenty-first century: Climate change and impacts scenarios, and potential for adaptation". In: *Frontiers in Plant Science* 7 (AUG2016), p. 1262. ISSN: 1664462X. DOI: 10.3389/FPLS.2016.01262.

Sundström, J. et al. (2014). "Future threats to agricultural food production posed by environmental degradation, climate change, and animal and plant diseases - a risk analysis in three economic and climate settings". In: *Food Security* 6 (2), pp. 201–215. ISSN: 18764525. DOI: 10.1007/S12571-014-0331-Y. URL: https://link.springer.com/article/10.1007/s12571-014-0331-y.

Szegedy, C. et al. (2016). "Rethinking the Inception Architecture for Computer Vision". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826. URL: https://doi.org/10.1109/CVPR.2016.308.

Szegedy, C. et al. (Feb. 2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31 (1). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11231.

Thongsuwan, S. et al. (2021). "ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost". In: *Nuclear Engineering and Technology* 53 (2), pp. 522–531. ISSN: 1738-5733. DOI: 10.1016/J.NET.2020.04.008.

Thresh, J. and R. Cooter (2005). "Strategies for controlling cassava mosaic virus disease in Africa". In: *Plant Pathology* 54 (5), pp. 587–614. ISSN: 1365-3059. DOI: 10.1111/J.1365-3059.2005.01282.X. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1365-3059.2005.01282.x.

Thresh, J., D. Fargette, and G. Otim-Nape (2016). "The viruses and virus diseases of cassava in Africa". In: *African Crop Science Journal* 2 (4), pp. 459–478. ISSN: 1021-9730. DOI: 10.4314/acsj.v2i4.. URL: https://www.ajol.info/index.php/acsj/article/view/135950.

Tollenaere, C., H. Susi, and A. Laine (2016). "Evolutionary and Epidemiological Implications of Multiple Infection in Plants". In: *Trends in Plant Science* 21 (1), pp. 80–90. ISSN: 1360-1385. DOI: 10.1016/J.TPLANTS.2015.10.014.

Tomlinson, K. et al. (2018). "Cassava brown streak disease: historical timeline, current knowledge and future prospects". In: *Molecular Plant Pathology* 19 (5), pp. 1282–1294. ISSN: 1364-3703. DOI: 10.1111/MPP.12613. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/mpp.12613.

Tonukari, N. et al. (2015). "White Gold: Cassava as an Industrial Base". In: *American Journal of Plant Sciences* 6 (7), pp. 972–979. ISSN: 2158-2742. DOI: 10.4236/AJPS.2015.67103. URL: http://www.scirp.org/Journal/Paperabs.aspx?paperid=55817.

Torkpo, S. et al. (2021). "Occurrence of cassava mosaic begomovirus-associated satellites on cassava in Ghana". In: *Cogent Food & Agriculture* 7 (1), p. 1963929. ISSN: 23311932. DOI: 10.1080/23311932.2021.1963929. URL: https://www.tandfonline.com/doi/abs/10.1080/23311932.2021.1963929.

Trebicki, P. (2020). "Climate change and plant virus epidemiology". In: *Virus Research* 286, p. 198059. ISSN: 0168-1702. DOI: 10.1016/J.VIRUSRES.2020.198059.

Tumwegamire, S. et al. (2018). "Exchanging and managing in-vitro elite germplasm to combat Cassava Brown Streak Disease (CBSD) and Cassava Mosaic Disease (CMD) in Eastern and Southern Africa". In: *Food Security* 10 (2), pp. 351–368. ISSN: 18764525. DOI: 10.1007/S12571-018-0779-2. URL: https://link.springer.com/article/10.1007/s12571-018-0779-2.

UN (2015). *Transforming our world: the 2030 Agenda for Sustainable Development Transforming our world: the 2030 Agenda for Sustainable Development Preamble*. United Nations General Assembly. URL: https://www.un.org/en/development/desa/population/migration/generalassembly/docs/globalcompact/A_RES_70_1_E.pdf.

— (2018). *The 2030 Agenda and the Sustainable Development Goals An opportunity for Latin America and the Caribbean Goals, Targets and Global Indicators*. United Nations. URL: www.cepal.org/en/suscripciones.

van Bavel, J. (2013). "The world population explosion: causes, backgrounds and projections for the future". In: *Facts, Views & Vision in ObGyn* 5 (4), p. 281. ISSN: 2032-0418. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3987379/.

van Dijk, M. et al. (2021). "A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050". In: *Nature Food 2021 2:7* 2 (7), pp. 494–501. ISSN: 2662-1355. DOI: 10.1038/s43016-021-00322-9. URL: https://www.nature.com/articles/s43016-021-00322-9.

van Ittersum, M. et al. (2016). "Can sub-Saharan Africa feed itself?" In: *Proceedings of the National Academy of Sciences of the United States of America* 113 (52), pp. 14964–14969. ISSN: 10916490. DOI: 10.1073/PNAS.1610359113. URL: www.pnas.org/cgi/doi/10.1073/pnas.1610359113.

van Munster, M. (2020). "Impact of Abiotic Stresses on Plant Virus Transmission by Aphids". In: *Viruses* 12 (2). ISSN: 19994915. DOI: 10.3390/V12020216. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7077179/.

Vasconcelos, L. et al. (2016). "Polymorphism of starch pathway genes in cassava". In: *Genetics and Molecular Research* 15 (4), p. 15049082. DOI: 10.4238/gmr15049082. URL: http://dx.doi.org/10.4238/gmr15049082.

Velásquez, A., C. Castroverde, and S. He (2018). "Plant and pathogen warfare under changing climate conditions". In: *Current biology* 28 (10), R619–R634. ISSN: 09609822. DOI: 10.1016/J.CUB.2018.03.054. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5967643/.

Vågsholm, I., N. Arzoomand, and S. Boqvist (2020). "Food Security, Safety, and Sustainability—Getting the Trade-Offs Right". In: *Frontiers in Sustainable Food Systems* 4, p. 16. ISSN: 2571581X. DOI: 10.3389/FSUFS.2020.00016.

Waisundara, V. (2018). "Introductory Chapter: Cassava as a Staple Food". In: *Cassava*. DOI: 10.5772/INTECHOPEN.70324. URL: https://www.intechopen.com/chapters/56558.

Wang, G., Y. Sun, and J. Wang (2017). "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning". In: *Computational Intelligence and Neuroscience*. ISSN: 16875273. DOI: 10.1155/2017/2917536.

Wang, H. et al. (2012). "Application of neural networks to image recognition of plant diseases". In: *2012 International Conference on Systems and Informatics, ICSAI 2012*, pp. 2159–2164. DOI: 10.1109/ICSAI.2012.6223479.

Wang, T. et al. (2021). "Transient Voltage Stability Assessment Method based on gcForest". In: *Journal of Physics: Conference Series* 1914.1, p. 012025. ISSN: 1742-6596. DOI: 10.1088/1742-6596/1914/1/012025. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1914/1/012025.

Ward, E. et al. (2004). "Plant pathogen diagnostics: immunological and nucleic acid-based approaches". In: *Annals of Applied Biology* 145 (1), pp. 1–16. ISSN: 1744-7348. DOI: 10.1111/J.1744-7348.2004.TB00354.X. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1744-7348.2004.tb00354.x.

Weng, L. et al. (2020). "Land use/land cover recognition in arid zone using A multi-dimensional multi-grained residual Forest". In: *Computers & Geosciences* 144, p. 104557. ISSN: 0098-3004. DOI: 10.1016/J.CAGEO.2020.104557.

Wielkopolan, B., M. Jakubowska, and A. Obrępalska-Stęplowska (2021). "Beetles as Plant Pathogen Vectors". In: *Frontiers in Plant Science* 12, p. 2241. ISSN: 1664462X. DOI: 10.3389/FPLS.2021.748093.

Willsey, T., S. Chatterton, and H. Cárcamo (2017). "Interactions of root-feeding insects with fungal and oomycete plant pathogens". In: *Frontiers in Plant Science* 8, p. 1764. ISSN: 1664462X. DOI: 10.3389/FPLS.2017.01764.

Winter, S. et al. (2010). "Analysis of cassava brown streak viruses reveals the presence of distinct virus species causing cassava brown streak disease in East Africa". In: *Journal of General Virology* 91, pp. 1365–1372. DOI: 10.1099/vir.0.014688-0. URL: http://pfam.sanger.ac.uk.

Wolfe, M. et al. (2016). "Genome-Wide Association and Prediction Reveals Genetic Architecture of Cassava Mosaic Disease Resistance and Prospects for Rapid Genetic Improvement". In: *The Plant Genome* 9 (2). ISSN: 1940-3372. DOI: 10.3835/PLANTGENOME2015.11.0118. URL: https://onlinelibrary.wiley.com/doi/full/10.3835/plantgenome2015.11.0118.

Wójtowicz, A. et al. (2021). "A random forest model for the classification of wheat and rye leaf rust symptoms based on pure spectra at leaf scale". In: *Journal of Photochemistry and Photobiology B: Biology* 223, p. 112278. ISSN: 1011-1344. DOI: 10.1016/J.JPHOTOBIOL.2021.112278.

Xia, L. et al. (2017). "Research on Potential Damage Estimation of Household Appliances Based on gcForest Model". In: *Proceedings of the 2017 International Conference on Software and e-Business - ICSEB 2017*. DOI: 10.1145/3178212. URL: https://doi.org/10.1145/3178212.3178216.

Xia, M. et al. (2018). "Cloud/snow recognition for multispectral satellite imagery based on a multidimensional deep residual network". In: *International Journal of Remote Sensing* 40.1, pp. 156–170. ISSN: 13665901. DOI: 10.1080/01431161.2018.1508917. URL: https://www.tandfonline.com/doi/abs/10.1080/01431161.2018.1508917.

Xie, W. et al. (2022). "Evaluation of Different Bearing Fault Classifiers in Utilizing CNN Feature Extraction Ability". In: *Sensors* 22 (9), p. 3314. ISSN: 1424-8220. DOI: 10.3390/S22093314. URL: https://www.mdpi.com/1424-8220/22/9/3314.

Yamashita, R. et al. (2018). "Convolutional neural networks: an overview and application in radiology". In: *Insights into Imaging* 9 (4), pp. 611–629. ISSN: 18694101. DOI: 10.1007/S13244-018-0639-9. URL: https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9.

Yang, D. et al. (2002). "Short RNA duplexes produced by hydrolysis with Escherichia coli RNase III mediate effective RNA interference in mammalian cells". In: *Proceedings of the*

National Academy of Sciences of the United States of America 99 (15), pp. 9942–9947. ISSN: 00278424. DOI: 10.1073/PNAS.152327299. URL: https://www.pnas.org.

Yang, S. et al. (2020). "Crop Classification Method Based on Optimal Feature Selection and Hybrid CNN-RF Networks for Multi-Temporal Remote Sensing Imagery". In: *Remote Sensing* 12 (19), p. 3119. ISSN: 2072-4292. DOI: 10.3390/RS12193119. URL: https://www.mdpi.com/2072-4292/12/19/3119.

Yao, Y., L. Rosasco, and A. Caponnetto (2007). "On Early Stopping in Gradient Descent Learning". In: *Constructive Approximation* 26 (2), pp. 289–315. ISSN: 1432-0940. DOI: 10.1007/S00365-006-0663-2. URL: https://link.springer.com/article/10.1007/s00365-006-0663-2.

Yeom, J. et al. (2021). "Segmentation of experimental datasets via convolutional neural networks trained on phase field simulations". In: *Acta Materialia* 214, p. 116990. ISSN: 1359-6454. DOI: 10.1016/J.ACTAMAT.2021.116990.

Yin, X. et al. (2018). "Deep Forest-Based Classification of Hyperspectral Images". In: *Chinese Control Conference, CCC* 2018-July, pp. 10367–10372. ISSN: 21612927. DOI: 10.23919/CHICC.2018.8483767.

Yuan, Z. and P. Zhao (2019). "An improved ensemble learning for imbalanced data classification". In: *Proceedings of 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2019*, pp. 408–411. DOI: 10.1109/ITAIC.2019.8785887.

Zarei, M. (2017). "Portable biosensing devices for point-of-care diagnostics: Recent developments and applications". In: *TrAC - Trends in Analytical Chemistry* 91, pp. 26–41. ISSN: 18793142. DOI: 10.1016/J.TRAC.2017.04.001.

Zeiler, M. and R. Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8689 LNCS (PART 1), pp. 818–833. ISSN: 16113349. DOI: 10.1007/978-3-319-10590-1_53. URL: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53.

Zhang, C. et al. (2015). "Biogenesis, function, and applications of virus-derived small RNAs in plants". In: *Frontiers in Microbiology* 6, p. 1237. ISSN: 1664302X. DOI: 10.3389/FMICB.2015.01237.

Zhang, J. et al. (2022). "Bearing Fault Diagnosis Based on a Novel Adaptive ADSD-gcForest Model". In: *Processes* 10.2, p. 209. ISSN: 2227-9717. DOI: 10.3390/PR10020209. URL: https://www.mdpi.com/2227-9717/10/2/209.

Zhang, Z. and M. Sabuncu (2018). "Generalised Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels". In: *32nd Conference on Neural Information Processing Systems*, pp. 8792–8802.

Zheng, L. et al. (2022). "Determination of adulteration in wheat flour using multi-grained cascade forest-related models coupled with the fusion information of hyperspectral imaging". In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 270, pp. 1386–1425. DOI: 10.1016/j.saa.2021.120813. URL: https://doi.org/10.1016/j.saa.2021.120813.

Zhou, X. et al. (1997). "Evidence that DNA-A of a geminivirus associated with severe cassava mosaic disease in Uganda has arisen by interspecific recombination". In: *Journal of General Virology* 78, pp. 2101–2111. DOI: 10.1099/0022-1317-78-8-2101.

Zhou, Z. and J. Feng (2017). "Deep Forest". In: *National Science Review* 6 (1), pp. 74–86. ISSN: 2053714X. DOI: 10.1093/nsr/nwy108. URL: https://arxiv.org/abs/1702.08835v4.

Zhu, Y., J. Duan, and T. Wu (2021). "Animal fiber imagery classification using a combination of random forest and deep learning methods:" in: *Journal of Engineered Fibers and Fabrics* 16. ISSN: 15589250. DOI: 10.1177/15589250211009333. URL: https://journals.sagepub.com/doi/full/10.1177/15589250211009333.

Zhuang, L. (2021). "Deep-Learning-Based Diagnosis of Cassava Leaf Diseases Using Vision Transformer". In: *2021 4th Artificial Intelligence and Cloud Computing Conference (AICCC '21)*, pp. 74–79. DOI: 10.1145/3508259.3508270. URL: https://doi.org/10.1145/3508259..

Zinga, I. et al. (2013). "Epidemiological assessment of cassava mosaic disease in Central African Republic reveals the importance of mixed viral infection and poor health of plant cuttings". In: *Crop Protection* 44, pp. 6–12. ISSN: 0261-2194. DOI: 10.1016/J.CROPRO.2012.10.010.

Zipfel, C. (2014). "Plant pattern-recognition receptors". In: *Trends in Immunology* 35 (7), pp. 345–351. ISSN: 1471-4906. DOI: 10.1016/J.IT.2014.05.004.

Zárate-Chaves, C. et al. (2021). "Cassava diseases caused by Xanthomonas phaseoli pv. manihotis and Xanthomonas cassavae". In: *Molecular Plant Pathology* 22 (12), pp. 1520–1537. ISSN: 1364-3703. DOI: 10.1111/MPP.13094. URL: https://onlinelibrary.wiley.com/doi/full/10.1111/mpp.13094.