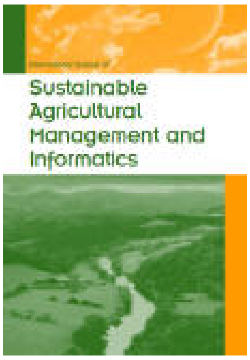

International Journal of Sustainable Agricultural Management and Informatics

ISSN online: 2054-5827 - ISSN print: 2054-5819
https://www.inderscience.com/ijsami

Automatic detection and diagnosis of cocoa diseases using mobile tech and deep learning

Richard K. Lomotey, Sandra Kumi, Rita Orji, Ralph Deters

DOI: 10.1504/IJSAMI.2024.10059217

Article History:

Received:	03 January 2023
Last revised:	21 March 2023
Accepted:	23 March 2023
Published online:	08 December 2023

Automatic detection and diagnosis of cocoa diseases using mobile tech and deep learning

Richard K. Lomotey*

The Pennsylvania State University – Beaver,
Information Sciences and Technology,
Monaca, PA, USA
Email: rkl5137@psu.edu
*Corresponding author

Sandra Kumi

Department of Computer Science,
University of Saskatchewan,
Saskatoon, SK S7N 5C9, Canada
Email: sandra.kumi@usask.ca

Rita Orji

Faculty of Computer Science,
Dalhousie University,
Halifax, NS B3H 4R2, Canada
Email: rita.orji@dal.ca

Ralph Deters

Department of Computer Science,
University of Saskatchewan,
Saskatoon, SK S7N 5C9, Canada
Email: deters@cs.usask.ca

Abstract: Cocoa is a cash crop that contributes about 3% to the gross domestic product (GDP) of Ghana's economy and makes up about 20% of total export receipts according to the Ghana statistical service. However, revenue has been hampered recently by the outbreak of cocoa diseases such as Swollen shoot and black pod thereby causing up to 11% loss of the crop. There is, therefore, a need for urgent intervention by all stakeholders within the cocoa production sector. In this research, we aim to employ mobile technology and machine learning (ML) techniques to enhance the early detection and diagnosis of the two major diseases that affect cocoa production namely – swollen shoot and black pod. Specifically, a distributed mobile application is developed that enables farmers to take a picture or video of the cocoa and the app will analyse and automatically detect the specific disease. The app consequently suggests the best treatment to undertake using an inbuilt-information guide. The automatic detection and diagnosis of diseases are based on deep convolutional neural networks (CNN) for image analysis, classification, and detection. The research analysed 2,828 cocoa images spread across three class labels. We built and trained four CNN models, namely CentreNet ResNet50 V2, EfficientDet D0, SSD MobileNet V2, and SSD ResNet50 V1 FPN. We found the best generalised and fastest model to be the SSD MobileNet V2 with a detection confidence score of approximately 88.0%.

Keywords: agriculture; mobile; deep learning; cocoa production; machine learning; ML; convolutional neural networks; CNN; classification; detection.

Reference to this paper should be made as follows: Lomotey, R.K., Kumi, S., Orji, R. and Deters, R. (2024) 'Automatic detection and diagnosis of cocoa diseases using mobile tech and deep learning', *Int. J. Sustainable Agricultural Management and Informatics*, Vol. 10, No. 1, pp.92–119.

Biographical notes: Richard k. Lomotey is currently an Associate Professor of Information Sciences and Technology (IST) at the Pennsylvania State University, USA. He is the Program Coordinator for the IST program and the Cybersecurity Analytics and Operations (CYAOP) program at Penn State Beaver Campus. He is researching the intersection of smart technologies, mobile computing, data science, machine learning (ML), and cybersecurity, with a focus on how these technologies are transforming enterprises and policies.

Sandra Kumi is currently a PhD student in Computer Science at the University of Saskatchewan, Canada. She received the BSc and MSc degrees in Computer Science from Kwame Nkrumah University of Science and Technology, Ghana and Dongseo University, South Korea respectively. Her research interests include data trust, blockchain, cybersecurity and ML.

Rita Orji is currently a Canada Research Chair in Persuasive Technology and a Computer Science Professor with Dalhousie University, Canada, where she directs the Persuasive Computing Laboratory. Her research interest includes the intersection of technology and human behaviour with a major focus on investigating user-centred approaches to designing technologies to improve lives and promote desirable behaviours. She applies her research to tackle real-life problems in various domains.

Ralph Deters received his PhD degree from the Federal Armed Forces University in Munich, Germany, in 1998. He joined the University of Saskatchewan, as a Research Associate, in 1998. He is currently a Full Professor with the Department of Computer Science, University of Saskatchewan. His research areas focus on distributed ledger technology, IoT, and cloud/edge computing.

This paper is a revised and expanded version of a paper entitled 'Cocoa companion: deep learning-based smartphone application for cocoa disease detection' presented at The 19th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), Niagara Falls, Canada, 9–11 August 2022.

1 Introduction

COCOA (Theobroma cacao) remains one of the preferred commodities for making edibles such as chocolates, beverages, drinks, ice creams, candies, and Mesoamerican foods. The same is also used in the manufacturing of other products and consumables such as medicines, cocoa butter, skin, beauty care products, etc. Today, Côte d'Ivoire and

Ghana remain the largest producers and exporters of the cocoa bean with an annual production capacity of over 2.50 million tons (Food and Agriculture Organization of the United Nations, 2022). Figure 1 shows the suitable regions where cocoa is grown in both countries. In Ghana, the cocoa sector supports over 4 million farm families and contributes around 3% to the gross domestic product (GDP) – valued at $2.00 billion annually through the foreign exchange of export crops (Abbadi et al., 2019).

The governments of Côte d'Ivoire and Ghana are projecting doubling production capacity in the coming years. However, production capacity in recent seasons has been on the decline due to the prevalence of cocoa pests (such as capsids, shield bugs, and caterpillars) and diseases (such as black pod and swollen shoot) (Dormon et al., 2004; Mbarga et al., 2020; Ofori et al., 2022). The issue of cocoa yield decline is also prevalent in all other production zones in the world including West Africa and Latin America.

Thus, this multi-national research partnership is formed to address this problem. The partnership is formed between researchers from the Pennsylvania State University – USA, the University of Saskatchewan – Canada, Dalhousie University – Canada, and stakeholders in the cocoa sectors in Ghana (specifically, cocoa farmers associations). The goal of this research is to enable cocoa farmers to use mobile technologies for early detection of cocoa pod infestation. Currently, farmers have to wait for the cocoa pod to show signs of infestation. And in the case of the Black Pod and Swollen Shoot diseases, the manual detection is done at the advanced stage of infestation, thereby making the treatment more difficult if not too late in some circumstances.

Over the years, the use of computing technology in the agriculture field has risen as well. Specifically, the use of mobile devices has greatly impacted the methods that farmers use to not only grow and sell their crops but also to determine the cause of any problems that may arise. For example, with the mobile application BioLeaf (Mendes et al., 2020), users can scan a picture of a leaf, and the system will identify the plant and diagnose the issue in real-time based on image analysis, machine learning (ML), and artificial intelligence (AI). Applications like these immensely help the agricultural field, as the ability to quickly identify the cause of the problem can lead to a speedier remedy for said problem. Other researchers such as Szegedy et al. (2015), Verma et al. (2019), Petrellis (2018), Chandra (2019), Garg et al. (2021), Park et al. (2018) and Jang et al. (2020) use image recognition for diagnosis. Precisely, Chandra (2019) designed a smartphone application for the diagnosis of white stem borer infection in coffee plants using image analysis. The studies in Esgario et al. (2021) also use a smartphone app to assist farmers in identifying if coffee leaves have diseases or pests based on ML. As well, Sanga et al. (2020) presented a study that is aimed at developing a mobile application with ML for the early detection of banana diseases.

Furthermore, there is a project that implemented an integrated and collaborative platform for automated disease diagnosis, tracking, and forecasting for farmers (Singh, 2018). The problem that farmers have is diagnosing diseases early enough in their crops to avoid losses. Also, some farmers are not experts in identifying diseases, so agricultural experts must examine the crops for them. To assist farmers in diagnosing diseases, mobile technology creates an opportunity for a low-cost solution that can be widely deployed (Singh, 2018). With such mobile apps, farmers can upload pictures of their plants to automatically get a diagnosis (Saleem et al., 2019).

Also, incorporating software into mobile computing devices provides agricultural info, improved accessibility, improve data visibility, offers motivation for diversity, and provides marketing information (Alagumariappan et al., 2020; Shafinah et al., 2020; Sibanda et al., 2021).

In this research paper, we posit that the integration of deep learning techniques (which is a subcategory of ML) into smart computing can aid with the automatic detection of black pod and swollen shoot diseases. In our estimation, this research is the first in the space to address cocoa-related infections using mobile and AI; and the successful deployment of such will greatly impact the production capacity and income-earning power of farmers. From our preliminary investigations, we did not find any existing solution for cocoa disease detection in any of the mobile app stores, scholarly databases, and online. Our initial interviews (administered through a questionnaire) with some of the cocoa farmers in Ghana reveal that no such mobile interventions exist currently for them though they are willing to use them should they be presented. In the same questionnaire, they indicated other features they would like to have in the mobile app such as weather info, humidity, local map, disease info, budget estimator, and market trends.

As a result, we designed and developed a smartphone application with the above features integrated. Regarding cocoa pod infestation, we employ deep learning techniques to enhance the detection and diagnosis of the two major diseases that affect cocoa production namely – swollen shoot and black pod. Specifically, a distributed mobile application is developed that enables farmers to take a picture or video of the cocoa and the app will upload the image to a software-as-a-service cloud computing platform. The analysis of the image is done on the cloud to detect the specific disease by comparing the image to a set of pre-trained image models. The result of the detection is returned to the mobile component of the app and consequently suggests the best treatment to undertake using an inbuilt-information guide. The automatic detection and diagnosis of diseases are based on convolutional neural networks (CNN) for image analysis and classification. We built and trained four CNN models, namely CentreNet ResNet50 V2, EfficientDet D0, SSD MobileNet V2, and SSD ResNet50 V1 FPN. The research analysed 2,828 cocoa images spread across 3 class labels. We found the best generalised and fastest model to be the SSD MobileNet V2 with a detection confidence score of 88.0%.

Overall, the work made the following contributions to both mobile computing and AI research:

- Proved that deep learning techniques can be integrated into a smartphone application to successfully detect the presence of Swollen Shoot and Black Pod diseases. This automated process can eliminate the guess work and reduce human error; subsequently, leading to cocoa yield maximisation.
- This precision agriculture app contains a cocoa disease treatment information guide that assists farmers to know which specific chemicals to apply.
- We have generated and provided cocoa disease datasets of three classes (for swollen shoot, black pod, and healthy) for future use by other researchers who may be interested in a pre-trained dataset within the space.
- While most of the works on plant disease recognition use image classification, our work used object detection, which is a combination of classification and localisation.

- The smartphone application has other desired features for the farmers that meet their daily needs. By consolidating these features in the app, farmers can do so many things from one platform without having to switch between multiple products.

The remaining sections of the paper are arranged as follows. Section 2 features existing works on mobile applications and plant disease diagnosis based on image analysis, and deep learning – CNN. Section 3 describes the architectural design of the mobile application while Section 4 details our ML technique. The evaluation of the system is carried out in Section 5. The paper concludes in Section 5 with our contributions and future research directions.

2 Background works

2.1 Precision agriculture and machine learning

Precision agriculture (a.k.a., smart farming, or smart agriculture) is the era where modern technologies are employed to increase agriculture produce yield, as well as enhance farm-market communication and supply chains. This era has witnessed the introduction of technologies to achieve plant disease identification, detection, and diagnosis in soft-real-time. As such, we have witnessed the development of technologies that can assist in the proper diagnosis of diseased crops through spectrometry (Owomugisha et al., 2020; Barh and Balakrishnan, 2018).

Plant diseases are one of the most difficult obstacles to pass when it comes to farming or gardening world-wide. For example, the USA loses one-third of crop production annually due to crop diseases (Ahmed and Reddy, 2021; Mrisho et al., 2020). Farmers in the USA face limited resources to observe plant symptoms. As a result, there is a huge need to improve the detection of diseases and monitor agriculture losses. Mobile phone applications can be used to detect diseases because smartphones are affordable for farmers. They can also use them to detect crop growth, find how to properly take care of them, and learn about controlling diseases or curing them. Mobile applications for crops also pave the way for research into finding reliable resources for plant health and other agricultural resources (Ahmed and Reddy, 2021).

More recently, however, AI techniques are being integrated into the agronomic sector to enhance early detection and diagnosis of crop infestation and diseases. The work presented by Elhassouny and Smarandache (2019), goes into depth about how researchers are using recently advancing neural network technology to better identify diseases in plants. Specifically, they inform us about using a deep CNN to identify several diseases in tomato plants. The convolutional neural network or CNN is a type of artificial neural network (an aspect of ML), which is widely used for image/object recognition and classification. Deep learning has a higher accuracy because it does not need to extract specific features, but it can search the whole image by extracting data in layers, using low-level features to obtain abstract high-level features (Liu and Wang, 2021; Gandhi et al., 2018; Chethan et al., 2021).

Diseased plants have always posed a problem in the agricultural industry and reducing the amount of waste produced from diseased plants is one of the main goals of precision agriculture. Up until now using mobile technology to determine whether a plant is healthy or not has not been possible, and farmers were required to send their plants to a lab to have them tested (Hampf et al., 2021). To fix this problem many researchers and scientists have developed several types of AI, most recently the Deep CNN variety, to allow for quick and accurate results using data from images captured on a mobile device to determine if a plant is infected with a disease. Elhassouny and Smarandache (2019) disclosed that they used data from over 7,000 pictures to train the AI and when put to the test it produced an accuracy of about 88.4%. Although the results of this experiment seem very promising, there are still a few hurdles to overcome before this technology is near perfect, the main limiting factor is that there is not enough training data supplied to the AI. This conclusion is similar throughout the entire agronomic-based deep learning spectrum. The reader is referred to Table 1, which details the ongoing works on deep learning and image analysis for crop disease detection.

The research presented by Yang et al. (2020) was based on a system that can detect the severity of crop diseases by taking photos on a smartphone. Most crop disease recognition applications work using a picture taken by the user, and a CNN to conduct image recognition. However, due to the ever-growing amount of crop diseases, a deep transfer learning approach called CDCNNv2 was implemented. CDCNNv2 works by essentially cleaning up the image significantly, by padding, pooling and finally flattening to help recognise diseases from images. The neural network then looks for similar symptoms and checks whether the plant from the image matches a known disease. If the neural network detects a plant disease, the user is informed about the severity, and the recommended steps going forward. Through CDCNNv2, the results were 92% accurate, and 13% faster. The work presented by Shamkuwar et al. (2018) also accomplished approximately 95% accuracy in classification.

Similarly, Pallagani et al. (2019) built a mobile-inspired CNN system called dCrop. Using approx. 54,306 image datasets, they created a CNN class of indexes, which are then put into rows and columns of a matrix. Then PyTorch and Tensorflow are used to serve as ML tools. The paper gives the confidence percentage of ResNet 34, AlexNet, and ResNet 50.

Also, Francis and Deisy (2021) researched and developed an architecture using neural networks and ML to identify crop diseases. The research used a training group of apple leaves, running through multiple methods of identification to find the most accurate ones. Their findings were that, of methods involving colour images and reconstructed images, the colour images were identified with the most accuracy while also having less size and computational cost.

Furthermore, Ramesh et al. (2018) used images of leaves and classifications of their characteristics to identify healthy or diseased crops through a dataset obtained under random forest. To gather leaf characteristic data, the use of a histogram of an oriented gradient (HOG) is a necessity. Leaf images are switched from RGB, into gray scale type images to help it find abnormalities through Random Forest. Although this system is very helpful, there are some limitations in terms of the system's accuracy. The system has approximately 70% accuracy, but a solution to this limitation is an increase in images in the system. This way the system has more data to choose from and compare characteristics with. The works by Petrellis (2019) and Kothari (2018) however utilised a smartphone application with machine vision to analyse the plants for diseases.

According to Chen et al. (2021), Taiwan is dealing with gaps in an aging agricultural workforce, whereby the younger generation of workers lack the experience to spot and treat pest outbreaks in their crops. To discover these varied pests, an AI-based pest detection system was implemented. The application would process images of infestation, and feed this information through an object detection model, which reaches a server via a cloud platform. Three object detection models in turn were created: the two-stage faster region-based convolutional network (Faster R-CNN), a one-step feed-forward single-shot detector (SSD), and a one-step you only look once v4 (YOLO v4). The YOLO v4 proved the superior network, identifying mealybugs with a 100% success rate, and the Coccidae and Diaspididae by 88% and 97% respectively.

2.2 The open issues

Despite the extensive literature reviewed and presented, we could not find any research or project(s) on addressing cocoa disease infestation identification and detection using mobile technology and AI. However, we could learn from existing literature on how mobile applications and AI research aid in the quicker identification of plant diseases in the USA, India, Taiwan, Brazil, and other parts of the world (Francis and Deisy, 2021;Chen et al., 2021; Hampf et al., 2021; Garg et al., 2021).

In Côte d'Ivoire and Ghana (the two major cocoa-producing countries in the world), the leading diseases that threaten cocoa yield capacity are the *Black Pod* and *Swollen Shoot*. The *Black Pod* disease is triggered by a fungus (Phytophthora) that persists under poor humidity conditions and excessive rain. On the other hand, Swollen Shoot is a viral disease transmitted to the cocoa plant by mealybugs (Dormon et al., 2004). In the case of the latter, infestation does not only reduce production yield but can kill the plant itself. And like the situation in Brazil and India with regards to traditional diagnosis [51], farmers were required to send their plants to a lab to have them tested (Dormon et al., 2004; Hampf et al., 2021). Our preliminary interviews with some Ghanaian cocoa farmers indicated that government agencies in charge of cocoa production infrequently send agronomic personnel to come and educate them on diseases and treatment. There could be logistical challenges too as well as bad road conditions linking cocoa farming communities; thereby delaying how often farmers could be assisted with plant disease diagnosis.

Most of these challenges could be overcome with a successful deployment of an automatic cocoa infestation and disease diagnosis tool such as a smartphone application with an AI capability.

2.3 The research goals

Based on the open issues and the general lack of research in precision cocoa farming, with its negative consequences affecting global production capacity, we formed this multi-institutional research partnership to address the same. Initially, an advanced party of researchers from the Pennsylvania State University traveled to Ghana in the Summer of 2021 with a questionnaire to interview cocoa farmers. Overall, 76 questionnaires were administered in the following regions – Volta, Eastern, and Western. And 72 were returned completed.

The questionnaire aims to understand what features cocoa farmers would like to have in a smartphone app and whether automatic disease detection and diagnosis is a preferred option. The main features they indicated to have in the mobile application are automatic disease detection and diagnosis (69/72 responses), market updates/trends (67/72 responses), disease info (63/72 responses), weather and humidity (61/72 responses), local map (61/72 responses), budget estimator (60/72 responses).

To address these needs, the following research questions are outlined specifically in this paper.

- How can cocoa infestation (black pod and swollen shoot) be detected automatically with minimal error with the use of a smartphone?
- With no existing cocoa-oriented image databases, how could enough images be generated for training the deep learning model?
- Which ML technique will work better for this work? Classification or object detection?
- How could the smartphone application be designed to guarantee faster results and what could be done in case of connectivity loss?

3 The mobile application design

The current version of the mobile application is designed in Android (since most of our test users on the cocoa farm own this mobile OS), and it will be made available for free download on the Google Play platform. The name of the app is *Cocoa Companion*. The iOS version is also under development to support wider adoption and usage. The architectural design of the app is shown in Figure 2. Overall, there are four main layers of the app which are: application layer, application programming interface (API) layer, image/file server, and ML component.

Due to the need to have access to some consumer services outside our implementation, the API Layer is designed to connect to external services using public gateways. Specifically, we interfaced three APIs which are: Weather API, News API, and the Google Map API. The weather information is gathered from the Weather API (https://openweathermap.org/api) which contains climatic information and weather forecasts (for hourly, daily, and 30 days). Some of the data available include humidity, visibility, temperature, pressure, clouds, etc. This information is retrieved as JSON via HTTP and displayed on the screen of the app for easy access by the ordinary user. The Map API is connected through the Google Maps SDK (https://developers.google.com/maps/documentation/android-sdk/overview). The SDK is available for Android, iOS, and JavaScript. Through the SDK, location information, local map, and tracking within the farmland can be achieved. The News API is meant to support cocoa farmers with current information on cocoa market trends and trades, news on ravaging diseases and pests, as well as their treatment. Multiple sources are being considered for this information and relevant news is pushed to the screen when they are available.

On the home screen (i.e., app interface), users have access to the application layer. There is a budget estimator that aids the farmers to compute their production cost including analysing their profit and loss. The budget estimator helps cocoa farmers to inventory their costs and receipts. And when sales are made and recorded, the profit margins are calculated. Production estimates can also be made based on the prior year's sales. Furthermore, there is an interface that helps farmers to have access to a disease Information guide which has been integrated into the app. The disease information guide contains information on symptoms and treatment of common cocoa diseases such as black pod and swollen shot.

There is also a button on the screen (labelled ‘diagnose infection’) which connects to the Image/camera component. This component aids the farmer to take pictures and upload the image as well. In case of no internet connectivity, the picture is kept in a local Storage module on the phone until connectivity is restored; then it is uploaded. If a farmer takes a video instead of a picture, there is a pre-process module that captures a still shot of the video frame and treats it as an image. This is because the ML layer currently supports only object detection in images. The image is uploaded via an HTPP gateway to a file server which is implemented in Erlang (https://www.erlang.org) programming language and hosted on an Amazon EC2 cloud computing platform. The ML component is connected to the Erlang file server in a bi-directional manner. Thus, when the file server sends an image to the ML layer, the latter process the image and does the detection. After that, the result is sent back to the Erlang File server. The file server then forwards the result to the mobile App. Based on the specific result that is returned, the app displays related information. This means that if the ML layer detects black pod disease, for instance, the app will open the black pod interface from the information guide. It is important to note that users can access the information guide anytime within the app.

The diagram in Figure 3 details the application process workflow of the mobile App. It clearly shows how to start the application and the various screens including which major actions to take on each screen. The process is streamlined so that it will not be cumbersome to use by an average cocoa farmer. Lastly, some screenshots of the application are shown in Figure 4.

4 Deep learning and image detection

Image classification predicts the class of an object in an image. Object localisation locates the presence of object categories in an image, along with an axis-aligned bounding box indicating the position and scale of one instance of each object category (Russakovsky et al., 2015). *Object detection* combines image classification and object localisation. Object detection locates the presence of objects with an axis-aligned bounding box and its class label. Some groupings of object detection methods are one-stage methods, two stage-methods, and anchorless.

Two-stage detectors such as region-based convolutional neural networks (R-CNN), faster R-CNN, and mask R-CNN use a region proposal network (RPN) to extract regions of interest (bounding box) in the first stage. In the second stage, the regions generated are used in the classification and localisation of the object. One-stage detectors such as YOLO (Redmon et al., 2016), single shot multibox detector (SSD) (Liu et al., 2016), EfficientDet (Tan et al., 2020), and RetinaNet (Lin et al., 2017b), predict the bounding box and class of the input image without generating region proposals. In one-stage detectors, backbone networks are used to extract features from the input image and output the bounding box and class of the image. Backbone networks can be densely connected networks like ResNet (He et al., 2016), AmoabaNet (Real et al., 2019), EfficientNet (Tan and Le, 2019) or lightweight networks such as MobileNet (Howard et al., 2017), MobileNetV2 (Sandler et al., 2018), Xception (Chollet, 2017), and SqueezeNet (Iandola et al., 2016).

Two-stage detectors have high localisation and object recognition accuracy, whereas one-stage detectors achieve high inference speed. Specific to our work, we shall discuss the following deep learning models as employed.

4.1 Single shot detector

The single shot detector (SSD) framework (Liu et al., 2016) is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes followed by a non-maximum suppression step to produce the final detections. As shown in Figure 5, the SSD model comprises two parts: extraction of feature maps and application of convolutional predictors for detection.

The SSD model uses the VGG-16 network (Simonyan and Zisserman, 2014) as a base network to extract feature maps. It then uses a $3 \times 3 \times p$ small convolution filter to predict the location and class score for a category. The default boxes of the SSD model are similar to the anchor boxes used in Faster R-CNN (Ren et al., 2015), however, applied to several feature maps of different resolutions. This is done to efficiently discretise the space of possible output box shapes.

In training the SSD model, ground truth information is assigned to specific outputs in the fixed set of detector outputs. The overall training objective loss function is a weighted sum of the localisation loss (loc.) and the confidence loss (conf.), with the weight term α set to 1 by cross-validation (as shown in equation (1)) (Liu et al., 2016).

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right) \tag{1}$$

where N is the number of matched default boxes, l is the predicted box, c is the class confidence, and g is the ground truth box.

In this work, we trained the SSD model with a MobileNet V2 and ResNet50 V1 FPN backbone.

4.1.1 SSD MobileNet V2

MobileNetV2 (Sandler et al., 2018) extends MobileNet (Howard et al., 2017) by introducing linear bottleneck layers and shortcut connections between the bottlenecks. The core idea is that bottlenecks encode the model's intermediate inputs and outputs while the inner layer encapsulates the model's ability to transform from lower-level concepts such as pixels to higher-level descriptors such as image categories (Mark and Andrew, 2018). MobileNet V2 is based on an inverted residual structure which reduces memory requirements to improve performance.

The MobileNet model is based on depthwise separable convolutions which is a form of factorised convolutions that factorise a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. MobileNetV2 also uses depthwise separable convolutions with ReLU6 except for the last linear layer. The diagram in Figure 5 further shows the difference between MobileNet V1 and MobileNet V2 architectures.

The authors of MobileNet V2 empirically proved that using non-linear layers in bottlenecks hurt the performance of a model.

4.1.2 SSD Resnet50 V1 FPN (RetinaNet50)

The RetinaNet (Lin et al., 2017b) architecture uses a feature pyramid network (Lin et al., 2017a) (FPN) backbone on top of a feedforward ResNet (He et al., 2016) architecture. ResNet50 (residual network) is a variant of the ResNet architecture that is 50 layers deep. It is based on a deep residual learning framework of stacked residual blocks to reduce the training of deeper models. Additional layers in neural networks improve accuracy and performance but deeper models have the issue of degradation (training error).

To solve this issue, residual networks use residual blocks to improve the accuracy of deeper models by learning residual functions about layer inputs instead of learning unreferenced functions. Residual networks are easy to optimise and can gain accuracy from increased depth. FPN is a feature extractor that uses a single-scale image of arbitrary size as input and generates multi-scale feature maps as output.

4.2 EfficientDet

The EfficientDet (Tan et al., 2020) architecture is designed to improve accuracy and efficiency in object detection. The EfficientDet model consists of three parts as shown in Figure 6.

4.2.1 Backbone

The backbone architecture uses the same width and depth of scaling coefficients of EfficientNet. EfficientNet (Tan and Le, 2019) is a convolutional neural network architecture targeted to maximise model accuracy for any given resource constraint. The authors of EfficientNet proposed a new compound scaling method, which uses a compound coefficient to uniformly scale network depth, network width, and input resolution in a principled way as shown in equations (2) (Tan and Le, 2019):

$$\left.\begin{array}{l} depth: d = \alpha^{\phi} \\ width: w = \beta^{\phi} \\ resolution: r = \gamma^{\phi} \\ s.t. \alpha \cdot \beta^{\phi} \cdot \gamma^{\phi} \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{array}\right\} \tag{2}$$

where α, β, γ are constants that can be determined by a small grid search. Increasing the network depth, width or input resolution improves accuracy but decreases the accuracy gain for bigger models. Experimental results show that EfficientNets transfer well and achieves a higher accuracy of 91.7% on CIFAR-100 datasets.

4.2.2 Neck (BiFPN)

The Bi-directional feature pyramid network (BiFPN) allows efficient multi-scale feature fusion by introducing learnable weights to learn the importance of different input features. As shown in Figure 6, BiFPN takes level 3–7 features $\{P_3, P_4, P_5, P_6, P_7\}$ from the backbone network and repeatedly applies top-down and bottom-up bidirectional feature fusion. BiFPN width and depth are scaled with equation (3) (Tan et al., 2020):

$$W_{bifpn} = 64(1.35^{\phi}), D_{bifpn} = 3 + \phi \tag{3}$$

where W_{bifpn} is the BiFPN width (number of channels), D_{bifpn} BiFPN depth (number of layers) and ϕ as the compound coefficient.

4.2.3 Head

The head of the model is made up of a box and class prediction network, which predicts the bound box coordinates and category of the object. The width is the same as BiFPN width, but the depth is linearly increased using equation (4) (Tan et al., 2020):

$$D_{box} = D_{class} = 3 + [\phi / 3] \tag{4}$$

4.3 CentreNet

The CentreNet (Zhou et al., 2019) is an anchorless object detection model based on a keypoint. It uses keypoint estimation to find the centre point of a bounding box and regresses to all other object properties such as bounding box size, 3D location, orientation, and pose. The inference process of CentreNet is a single network feedforward without non-maximal suppression for post-processing. The input image is fed to a fully convolutional network that outputs a heatmap. The image features at each peak in the heatmap predict the targeted object bounding box's height and weight. The overall training loss is computed using equation (5) (Zhou et al., 2019):

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \tag{5}$$

where L_k denotes keypoint heatmap, L_{size} is the object size loss and L_{off} is the offset trained with an L^1 loss. λ_{size} and λ_{off} is set to 0.1 and 1 respectively. The training object of the keypoint heatmap is a penalty-reduced pixel-wise logistic regression with focal loss.

In this work, we trained CentreNet with Resnet50 backbone.

4.4 System overview

In this work, the main aim is to detect diseases that affect cocoa using the deep learning models discussed. We train the following four deep learning models: CentreNet ResNet50 V2, EfficientDet D0, SSD MobileNet V2, and SSD ResNet50 V1 FPN. These models were trained on three classes – 2 classes of infected cocoa (Black Pod and Swollen Shoot) and one class for Healthy cocoa. Our system takes a cocoa image as input and outputs the classification and localisation information in the image.

The general overview of the system is shown in Figure 7 and detailed in the following subsections.

4.4.1 Data collection

Our dataset consists of two types of cocoa diseases (Black Pod and Swollen Shoot) and healthy cocoa. The image dataset of cocoa was collected from cocoa farms in Ghana. A simple camera was used to capture images with the background of objects surrounding the cocoa pod inclusive. The images were saved in JPEG format with a dimension of $3{,}120 \times 4{,}160$.

4.4.2 Data augmentation

The performance of deep learning models depends on the relevance, quality, and quantity of training data. Augmentation is essential when the training data quantity is insufficient. Data augmentation reduces overfitting during training and improves the generalisation ability of deep learning models. We apply different transformation techniques such as rotation, blurring, translation, noise, and vertical and horizontal flips to increase the size and diversity of the dataset. The scikit-image library in Python was used to transform the images.

In Table 2, the number of original images and augmented images generated for each class are shown and Figure 8 illustrates the various transformation applied in the augmentation process.

4.4.3 Data annotation

We manually annotate each image in our dataset to define a region (bounding box) and its corresponding class using LabelImg (Lin, 2015). We seek the knowledge of some experts in crop diseases to predetermine the classes and correctly label images. Our dataset contains three classes of images (black pod, swollen shoot, and healthy cocoa). Each image is annotated with one bounding box and its corresponding class.

We use the MS COCO evaluation to analyse our dataset. Table 3 shows the distribution of the aspect ratio and area of the bounding box across the dataset, and it is plotted in Figure 9. The MS COCO evaluation metrics use different object scales for object detection, the area of the bounding box below 32 × 32 pixels, the area of the bounding box from 32 × 32 pixels to 96 × 96 pixels, and the area of the bounding box above 96 × 96 pixels for SMALL, MEDIUM, and LARGE objects respectively. Our work focuses on the ability of deep learning models to detect large and medium objects. As shown in Figure 9, the number of medium and large-area objects takes up to 1.73% and 98.27%, respectively. We do not have small objects in our dataset.

4.4.4 Cocoa disease detection

We trained four pre-trained models using the TensorFlow object detection API (Kasper-Eulaers et al., 2021) based on the following models: SSD MobileNet v2, SSD ResNet50 v1 FPN, EfficientDet D0, and CentreNet ResNet50 v1. These four models were previously trained on the COCO dataset. We divided our dataset into two: training and testing sets. The training set contained 2,262 images with their annotations and the testing set contained 566 images with 566 annotations.

The pre-trained models were fine-tuned on our dataset and maintained the default values of their pipeline configuration. The SSD MobileNet V2, SSD ResNet50 V1 FPN, and EfficientDet D0 use weighted smoothed L^1 as the localisation loss and weighted sigmoid focal as the classification loss. CentreNet ResNet50 V1 uses L^1 for its localisation loss and penalty reduced logistic focal as the classification loss. The batch size of each model was adjusted. The training sessions were configured to run for 50,000 steps. However, our experiments proved that the models did not need 50,000 steps in training to reach minimum loss. In some models, the learning rate started decreasing after 6,000 steps. Hence to have a fair comparison of the different models, training was stopped at 8,000 steps. Table 4 shows the batch size and other hyperparameters used in training each model.

In Algorithm 1, we detail the steps for training a deep learning model using TensorFlow object detection API:

- Data preparation: collect images and annotate images for object detection.
- Split data into training and testing sets. Transform each dataset into TFRecord format.
- Create a label map for the three classes of cocoa and save it to a .pbtxt file.
- Select model to train and configure: Fine-tune the pre-trained model for cocoa disease detection (number of classes, batch size, label map, and number of training steps).
- Train model and save to checkpoint file.
- Evaluate the test dataset and save the results to a file.

Algorithm 1 for Cocoa Disease Detection

Input: Dataset D, Pre-trained DL model M_{pt}, Hyperparameters of DL model $H(M_{pt})$, label map of classes, C

Output: Trained model M_t and evaluation results, M_e

B ← Batch size

$n(C)$ ← Number of classes

N ← Number of Steps

ckpt ← Checkpoint (for every 1,000 steps model is saved)

for $i \leftarrow 1$ to number of images in D do
        Extract the image and annotate with specific class Cl
end for
Split D: $D_{train} = 80\%$, $D_{test} = 20\%$
Generate TFRecords TFR_{train}, TFR_{test}
Fine-tune $H(M_{pt})$ with TFR_{train}, TFR_{test}, C, B, $n(C)$, N
for $j \leftarrow 1$ to N
    $M_t \leftarrow$ Train M_{pt} with TFR_{train}
    if $j == ckpt$ **then**
    Save Mt
    end if
end for
for $k \leftarrow 1$ to N
    Evaluate Mt with TFR_{test}
    Save evaluation results to Me
end for

5 Evaluation

In this section, we evaluate the performance of the four models mentioned in our dataset. We divided the cocoa dataset into a training set (80% images) and a test set (20% images). The object detection models used implementation from the TensorFlow object detection API (Yu et al., 2020). Training and inference scripts were written in Python and compatible with only TensorFlow V2.

We trained SSD MobileNet V2, EfficientDet D0, and CentreNet ResNet50 V2 on *Google Colab GPU*, *NVIDIA Tesla K*80, and SSD ResNet50 V1 FPN on a *Mac operating system* with an Intel Core i7 3.8 GHz 8-Core Processor with a 32GB of memory. All models were trained with 8000 steps and different batch sizes as shown in Table 4. All models were evaluated using a batch size of 1. The Cocoa Dataset will be available for public access on our project GitHub website.

5.1 Evaluation metrics

We evaluate the performance of deep learning models on our dataset, we adopted the COCO detection evaluation metrics (Lin et al., 2014). Intersection over Union (IoU) is used to evaluate the prediction of a model. IoU is the overlap between the predicted bounding box B_p and the ground truth bounding box B_{gt} (Simonyan and Zisserman, 2014). The IoU must exceed 0.5 (50%) using equation (6):

$$IoU = \frac{area\left(B_p \cap B_{gt}\right)}{area\left(B_p \cup B_{gt}\right)} \tag{6}$$

where $B_p \cap B_{gt}$ denotes the intersection of the predicted and ground truth bounding boxes and $B_p \cup B_{gt}$ denotes their union.

The following metrics are used to evaluate the performance of the models:

5.1.1 Average precision

The AP summarises the area under the interpolated precision-recall curve. The AP is defined as shown in equation (7) (Everingham et al., 2010):

$$AP = \sum_{i=1}^{n-1} \left(r_{i+1} - r_i\right) p_{interp}\left(r_{i+1}\right) \tag{7}$$

where $r_1, r_2, \ldots, r_n$ is the recall level at which precision interpolate and denotes p_{interp} the interpolated precision. The AP can be computed for each class in the dataset. Traditionally, if the dataset contains K classes, the AP averaged over all classes is defined as the mean average precision (mAP). This can be formalised as shown in equation (8) (Everingham et al., 2010).

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{8}$$

However, the COCO evaluation metrics make no distinction between the AP and mAP, likewise for the average recall (AR) and mean average recall (mAR). The AP is the primary metric used to evaluate the performance of object detection models. The higher the score of the AP, the more accurate the model is. The AP is computed using 10 IoU thresholds of 0.50: 0.05: 0.95, that is AP averaged across all 10 IoU thresholds from 0.5 to 0.95 with a step size of 0.05. This is different from the evaluation metrics of the Pascal VOC dataset where the AP is computed at an IoU of 0.50 which is equivalent to $AP^{IoU=0.50}$.

5.1.2 Average recall

The AR is defined as the maximum recall given a fixed number of detections per image, averaged over classes and multiple IoUs. The AR is computed as shown in equation (9).

$$AR = \frac{1}{O}\sum_{o=1}^{O} k \mid Pr_{t(o)}(\tau(k)) > 0 \max\left\{Rc_{t(o)}(\tau(k))\right\} \tag{9}$$

where O is the IoU, $t(o)$ is a set of O IoU thresholds, and $Pr_{t(o)}$, $Rc_{t(o)}$ $(\tau(k))$ is the *precision* × *recall* points for confidence $\tau(k)$. $AR^{MAX=1}$ is AR given 1 detection per image, while $AR^{MAX=10}$ and $AR^{MAX=100}$ are AR given 10 and 100 detections per image respectively.

To evaluate the multi-scale object detection capabilities of deep learning models AP and AR across scales are taken into consideration:

- AP^{SMALL} and AR^{SMALL} evaluate the ground truth of small objects (area < 32^2)
- AP^{MEDIUM} and AR^{MEDIUM} evaluate the ground truth of medium objects (32^2 < area < 96^2)
- AP^{LARGE} and AR^{LARGE} evaluate the ground truth of large objects (area > 96_2).

5.2 Experimental results

The dataset for each class was split into an 80% training set and a 20% testing set. Tables 5, 6, 7, and 8 show the evaluation performance of the models on the testing dataset.

Table 5 shows the performance of the models in terms of average precision in each class (AP). In the detection of black pod disease, the SSD MobileNet V2 performed best with AP of 50.5%, and AP of 88.8%, 44.5%, and 50.5% for IoU thresholds of 0.5, 0.75, and large objects respectively. This performance is followed by the EfficientDet DO model with an AP of 35.3%. At the IoU threshold of 0.50, SSD Resnet50 V1 FPN and EfficientDet DO achieved an AP of 83.3% and 81% respectively for Black Pod.

In the case of the healthy class, SSD Resnet50 V1 FPN obtained the highest AP of 37.6%. At the IoU threshold of 0.50, SSD MobileNet V2 returns the highest AP of 83.6%. For the detection of swollen shoot, EfficientDet DO and SSD Resnet50 V1 FPN both obtained an AP of 32%.

CentreNet Resnet50 v2 has the lowest performance with an AP of 33% for both healthy and black pod classes and 24.4% for the swollen shoot. In the case of AP for small and medium objects of the black pod and swollen shoot category, all models have a result of –1. This means that the model could not detect the objects in these classes. This is because the bounding box of this class is large and as such, the small and medium bounding boxes will go undetected.

Furthermore, Table 6 shows the overall average precision (AP) of the CNN models on the three classes. The SSD Resnet50 V1 FPN outperformed the other models with an AP of 32.4%. The $AP^{IoU = 0.75}$ and AP^{LARGE} of the same model returned 19.4% and 32.7% respectively. The SSD MobileNet V2 achieved the lowest AP of 29.1. At a single IoU threshold of 0.50, all the models obtained an AP above 50% with EfficientDet D0 achieving the highest score of 81%. While the SSD MobileNet V2 model performs better than CentreNet Resnet50 v2 with an improvement of 2.6% in detecting medium objects, CentreNet Resnet50 v2 receives the highest AP of 20%.

As explained earlier, our dataset contains large and medium objects and has no small-sized objects. Hence the implemented models have an AP and AR of –1 for small objects.

As shown in Table 7 all the models achieved a higher AR in the detection of black pod disease compared to the other two classes. In the detection of black pod, SSD MobileNet V2 has the highest AR of 52.9%, 63.7%, and 64% for $AR^{MAX = 1}$, $AR^{MAX = 10}$, and $AR^{MAX = 100}$ detections per image respectively. The returned value for AR^{LARGE} is 58.8%. The SSD Resnet50 V1 FPN model performs best in the detection of Healthy cocoa with AR of 39.9%, 57%, 58.4 and 58.8% given a maximum of 1, 10 and 100 detections per image and large objects respectively.

Figure 13 illustrates the AP comparison of the models in the detection of each class. From our analysis, we observed that all models performed better in the detection of black pod and healthy categories than swollen shoot. A simple reason for this could be attributed to the fact that black pod and healthy cocoa classes are distinct and much larger. The worst performance of all models in the detection of swollen shoot is attributed to the limited diversity of images and lower lighting conditions.

Figure 14 shows the overall AP performance comparison of the models. SSD Resnet50 V1 FPN achieves the highest AP of 32.4%. This performance is due to the use of the FPN backbone to generate rich features and a focal loss function to eliminate the accuracy gap. EfficientDet D0 obtained the second-highest AP of 32%. This is attributed to the BiFPN backbone of EfficientDet architecture which improves the accuracy of the predicted bounding box of a target object. A further comparison of the overall AP shows that CentreNet ResNet50 V2 outperforms SSD MobileNet V2 because CentreNet ResNet50 V2 uses keypoint estimation to locate centre points and regresses object properties to achieve good accuracy.

Additionally, Table 8 shows the overall AR of the implemented models. In the detection of medium and large objects, SSD Resnet50 V1 FPN obtained the highest AR of 52.9% and 59.5% respectively. Given a maximum of 10 detections per image, EfficientDet D0 and CentreNet Resnet50 V2 achieved a similar AR of 51%. For one detection per image, CentreNet Resnet50 V2 has the highest AR of 37.1%.

To verify the detection capability of the CNN models, we tested all the models with previously unseen images, one image for each class. From our evaluation results in Table 6 and Table 8, SSD MobileNet V2 performed worst among all the models, however, it performed best in the prediction of new images followed by EfficientDet D0. SSD ResNet50 V1 FPN with the highest mAP performed slightly better by detecting two out of three classes. CentreNet ResNet50 V2 performed worst in detecting new images as it was able to detect only one out of three classes. Figure 10 shows the detection results in comparison for black pod as returned by the three models. The SSD MobileNet V2 has an 88% confidence score for prediction while EfficientDet D0 and SSD Resnet50 v1 FPN have confidence scores of 66% and 55% respectively. Figure 11 shows the detection results in comparison for the Healthy cocoa pod as returned by all four models. The SSD MobileNet V2 achieves the highest confidence score of 74%, 61% for both EfficientDet D0 and SSD Resnet50 v1 FPN, and CentreNet Resnet50 V2 has 54%. Figure 13 on the other hand presents the detection results in comparison for Swollen Shoot from only two models. As shown in Figure 12, only SSD MobileNet V2 and EfficientDet D0 were able to detect Swollen Shoots with a confidence score of 54% and 55% respectively.

Table 9 shows the detection capability of the models for specific classes and their confidence score; where black pod, swollen shoot, and healthy are denoted as BP, SS, and H respectively.

5.3 Model training

In this section, we discuss the training performance of the mentioned models. The *TensorBoard* is used to track the training loss and learning rates of the implemented models. The performance of the models is observed based on the following metrics: classification loss, learning rate, localisation loss, regularisation loss, and total loss. These metrics are explained as follows:

- Classification loss – determines the prediction of the target categories (classes) in a dataset.
- Localisation loss – indicates the difference between the predicted bounding box and the true values
- Regularisation loss – is generated by the model's regularisation function (L1 and L2 regularisation). Regularisation optimises the generalisation ability of object detection models by preventing overfitting.
- Total loss – evaluates the training performance of models. It is the sum of classification, localisation, and regularisation loss.
- Learning rate – this curve indicates the rate at which the model learns during training.

5.3.1 Results: SSD MobileNet V2

Figure 15 shows the training performance of SSD MobileNet V2 for each class. After training for 8,000 steps all classes had a learning rate of 0.07696. Black pod has the highest localisation, classification, and total loss of 0.09812, 0.1591, and 0.328 respectively. The classification loss of swollen shoot starts at 0.45 and ends at a value of 0.1399. The swollen shoot has the highest regularisation loss 0.07167. The healthy class has the lowest localisation, classification, regularisation, and total loss of 0.08433, 0.09975, 0.06635, and 0.2504 respectively. The losses of each class decrease as the training steps increase.

5.3.2 Results: EfficientDet D0

Figure 16 illustrates the training losses and learning rate of the EfficientDet D0 model in each class. In the learning rate graph, the model starts at 0.004 for all classes and significantly increases to 0.08 at 2500 steps. It then decreases to 0.07993 after training for 8,000 steps.

The classification, localisation, and total loss of the Healthy class decline at step 4000 compared to the other classes. The regularisation loss for each class increased after training. Consequently, the Healthy class has the highest value of 0.03044.

5.3.3 Results: SSD ResNet$_{50}$ V1 FPN

Figure 17 shows SSD ResNet50 V1 FPN training performance in each class. The learning rate for each class decreased noticeably after 2,000 steps. The classification loss, localisation loss, and total loss of the black pod class dramatically drop at step 5,700 at a value of 0.1826, 0.0909, and 0.3844 respectively.

The swollen shoot class has the least regularisation loss of 0.0812. The total loss for the Healthy class is the least compared to the black pod and swollen shoot classes.

5.3.4 Results: CentreNet ResNet50 V2

Figure 18 shows the training performance of the CentreNet ResNet50 V2 model in each class. The training performance is slightly worse across all the classes. The CentreNet ResNet50 V2 uses slightly different metrics from the other models. It uses the loss box offset, loss box scale, loss object centre, learning rate, and total loss. The box loss is how well the algorithm can locate the centre of an object and how well the predicted bounding box covers an object (Magalhães et al., 2021).

The loss box offset for the black pod class decreased at step 6,900 but at the end of the training process, the swollen shoot class has the least value of 0.4831. The healthy class has the highest loss box scale of 0.8472. The loss object centre for the swollen shoot class is the least with a value of 1.387. The total loss of black pod, swollen shoot, and healthy classes started at 3.947, 4.251, and 4.256. Then ended at 2.376, 2.287, and 3.104 respectively. We observed that the total loss did not minimise after training for 8,000 steps.

5.3.5 Results: combined classes

In Figure 19, we show the training performance of the models trained on all three classes combined. All the models were trained for 8,000 steps to make a fair comparison. At the end of the training process, the SSD MobileNet V2, EfficientDet D0, SSD ResNet50 V1 FPN, and CentreNet Resnet50 V2 each have a learning rate of 0.07696, 0.07993, 0.03365, and 1.00E-03 respectively.

The SSD ResNet50 V1 FPN has the least value of training classification at 0.2387, while SSD MobileNet V2 has a classification loss of 0.2495. EfficientDet D0 starts training with a classification loss of approximately 1.079 and ends with a value of 0.2861. EfficientDet D0 has the least value of approximately 0.00859 for localisation loss. This is attributed to the BiFPN backbone of EfficientDet architecture which improves the accuracy of the predicted bounding box of a target object.

The SSD MobileNet V2 model has a regularisation loss value of 0.09284 while that of EfficientDet D0 is 0.03271. The highest regularisation loss was recorded for SSD ResNet50 V1 FPN with a value of 0.1326.

Moreover, the SSD MobileNet V2, EfficientDet D0, SSD ResNet50 V1 FPN, and CentreNet Resnet50 V1 have a total loss of 0.4701, 0.3274, 0.6563, and 2.679 respectively. The higher value of total loss for CentreNet Resnet50 V1 is attributed to the higher value of object centre loss compared to the loss of box offset and box scale as shown in Figure 20. The CentreNet Resnet50 V2 has a loss value of 1.577, 0.4497, and 0.6523 for the object centre, box offset, and box scale respectively.

5.3.6 Training and validation of total loss

Table 10 shows the training and validation total loss of SSD MobileNet V2, EfficientDet D0, and SSD Resnet50 V1 FPN models on individual classes. The SSD MobileNet V2 model has a validation total loss of 0.79226, 1.41879, and 1.20369 for the black pod, swollen shoot, and healthy classes respectively. The validation total loss for each class is over 0.5 higher than the training total loss which indicates overfitting.

For swollen shoot disease, EfficientDet D0 has a value of 0.2897 and 0.36138 respectively for training and validation total loss. In the case of healthy and black pod classes, EfficientDet D0 converges at a value of approximately 0.36 for both training and validation total loss. EfficientDet D0 does not show overfitting or underfitting in the detection of each class.

On the other hand, the training and validation total loss of SSD Resnet50 V1 FPN is approximately 0.65 for the swollen shoot. The validation total loss is slightly lower than the training total loss, indicating underfitting in the Black pod class. In the healthy class, the validation total loss is slightly higher than the training total loss, which indicates that the model does not experience overfitting or underfitting issues.

Table 11 illustrates the training and validation loss of CentreNet Resnet50 V2 in individual classes and overall classes. The model experience overfitting except in the healthy class which has the value of 3.24173 and 3.104 for validation and training total loss. In the black pod, swollen shoot, and overall classes, the difference in the validation and training loss is more than 0.4. From our result, we deduce that CentreNet Resnet50 V2 does not generalise well with the testing data, and hence requires more training steps and data.

Table 12 shows the training and validation loss of SSD MobileNet V2, EfficientDet D0, and SSD Resnet50 V1 FPN models on overall classes. The validation total loss for EfficientDet D0 is slightly higher than its training total loss at a value of 0.37007. The training and validation loss of SSD Resnet50 V1 FPN has a value of approximately 0.65. Both EfficientDet D0 and SSD Resnet50 V1 FPN did not overfit or underfit the training data. On the other hand, the SSD MobileNet V2 has a training total loss of 0.4701 and a higher validation total loss of 1.16025.

6 Conclusions

In this section, we conclude the paper with discussions on the undeniable need to address the issue of cocoa production yield losses with the use of modern tools such as smartphones and ML. This need is highlighted by the rate at which the major cocoa production zones (across Africa and Latin America) are experiencing cocoa pod infestation due to black pod and swollen shoot diseases.

6.1 Discussion of results

In this work, we demonstrated that mobile technology and deep learning can be employed to automatically diagnose cocoa pod infestations. Specifically, we focused on the detection of swollen shoot and black pod diseases – the two most prevalent diseases that affect cocoa production. Like the situation in other jurisdictions, some farmers (especially the younger demographic) find it harder to identify crop diseases, particularly if the diseases are symptomatically similar. There could also be logistical challenges and bad road conditions linking cocoa farming communities can become unmotorable, thereby delaying how often farmers could be assisted with plant disease diagnosis. It is therefore our position that most of these challenges could be overcome with a successful deployment of an automated cocoa infestation and disease diagnosis tool such as a smartphone application with a ML capability.

Though cocoa disease detection is an understudied area, we could learn from existing literature. Thus, we designed and develop the Cocoa Companion application which enables cocoa farmers to take a picture of an infested pod and upload the picture for automatic disease detection and diagnosis. The diagnosis is done on a back-end cloud server and the result is returned to the mobile endpoint. This process is done in a soft-real-time when there is stable network connectivity. In case of connectivity loss, the application stores the photo on the local device until connectivity is restored.

With regards to the ML technique, we employed the deep CNN with the following four models – SSD MobileNet V2, EfficientDet D0, CentreNet Resnet50 V2, and SSD Resnet50 V1 FPN. Then three classes are trained consisting of black pod, swollen shoot, and healthy cocoa.

SSD MobileNet V2 was the best-generalised model for detecting new samples. The model achieved the highest confidence score of 88% when predicting Black Pod disease as shown in Table 9. The SSD MobileNet V2 has the least model size of 284.6 MB and an inference time of 4.2 ms on GPU. It outperforms the other implemented models in detection speed but had the lowest AP. This unique performance is attributed to the depthwise separable convolution which reduces computation. In addition, the use of input image resolution and width multiplier hyperparameters improves the performance of the SSD MobileNet V2 model. Also, the SSD Resnet50 V1 FPN model achieved the highest mAP and infers images in 23 ms. It is a deeper model and hence will have a higher AP and inference latency. The EfficientDet D0 model had an impressive performance with a confidence score of above 50% in detecting all cocoa classes and inferring images in 9.4 ms. Moreover, the CentreNet Resnet50 V2 model performed slightly better than the SSD MobileNet V2 model, however, it failed to detect some classes of cocoa.

We optimised two models, the SSD MobileNet V2 and EfficientDet D0, for the detection of cocoa diseases on smartphones. These models have the fastest inference latency compared to the other models, hence appropriate for use in the real world. Also, the SSD MobileNet V2 limits the number of parameters and does fewer calculations, therefore can run efficiently on mobile devices. The EfficientDet D0 employs model scaling to obtain better accuracy of predictions.

We used the MS COCO dataset evaluation metrics to comprehensively evaluate the performance of models on our cocoa dataset. The SSD MobileNet V2, EfficientDet D0, CentreNet Resnet50 V2, and SSD Resnet50 V1 FPN achieved an AP of 20.2%, 33.6%, 29.5%, and 34.3% respectively on COCO 2017 dataset (Kasper-Eulaers et al., 2021). Most researchers working in plant disease detection use the IoU threshold of 0.50 to determine the mAP of the models. This is equivalent to our $AP^{IoU = 0.50}$ metric. Magalhães et al. (2021) evaluated SSD and YOLO models for the detection of tomatoes in a greenhouse using the Pascal VOC challenge metrics. The SSD MobileNet V2 achieved an mAP of 51.46%, which is similar to our results

Also, Yan et al. (2021) proposed a lightweight apple detection for picking robots using improved YOLOv5. The authors compared their work with the EfficientDet D0 network. At a confidence threshold of 0.5, the EfficientDet D0 obtained an mAP of 80%. In comparison to our work, the SSD MobileNet V2 and EfficientDet D0 reached an mAP of 71.7% and 81% at the IoU threshold of 0.50 as shown in Table 6. Our results in this study showed that object detection models developed through transfer learning can accurately detect cocoa diseases.

It is important to note however that most of the works found in the area of automatic plant disease diagnosis using ML rely on classification techniques. This is why the basic metrics used in their reporting are accuracy, epochs, precision, recall, and F1-score. Our work on the other hand is based on the object detection technique (i.e., the combination of classification and localisation). Thus, the basic reporting metrics are the mAP, AP, AR, and steps.

The work can be extended to make disease identification in other plants much easier and more efficient based on object detection techniques. This is a huge deviation from the traditional classification technique that has been used in the past but with several limitations. Our results could be used as a benchmark to improve the object detection technique. Moreover, our dataset will be made available publicly for use by other researchers who may be interested in cocoa disease detection.

6.2 Limitations

Our experimental results proved that deep learning models can detect cocoa diseases. However, due to low image quality and insufficient data augmentation techniques used, the performance of the models could be limited. There were instances of overfitting and generalisation in some models due to less quantity of data to represent classes of disease. Moreover, the annotation of images was done by one annotator, which could lead to fewer variations and the possibility of bias in the labeling of images.

It is however important to state that the limitations observed in this work have also been experienced in prior works as highlighted in Table 1.

6.3 Future works

The immediate next phase of this project collaboration is to do field training in the Summer of 2023 for the cocoa farmers in Ghana. Educational Workshops will be organised in collaboration with the Ghana COCOBOD (the main government agency responsible for the management of cocoa affairs) for farmers in Ghana. The research team will perform usability training on a mass scale in the following cocoa-producing regions in Ghana – Volta, Eastern, Western, and Ashanti. The educational workshop will highlight the usefulness, need for adoption, and affordances of the smartphone app. This training will be extended later to Côte d'Ivoire.

Furthermore, as reported earlier, this paper focused on cocoa disease detection using deep learning. Object detection is the combination of classification and localisation techniques. The future technical exploration of this work will consider cocoa disease diagnosis using classification techniques and transfer learning. In this case, there is a need to upgrade the quantity and quality of the diseased cocoa image database. Then, the performance of the future classification technique can be compared to that of the current detection technique.

Moreover, due to the deployment of an architecture that detects black pod and swollen shoot diseases successfully with commendable precision, the future upgrade can include other disease classes such as caterpillar infection, etc. This step is currently under full consideration. The images needed are already available and the augmentation process has been completed.

Figures, Tables and the Appendix are available on request by emailing the corresponding author.

Acknowledgements

Grateful to the Pennsylvania State University, USA, University of Saskatchewan, Canada, Dalhousie University, Canada, and Ghana Cocoa Board.

Special thanks to the following undergraduate students who contributed immensely to the mobile user interface design – Dylan Kelly and Jonathan Woodstuff.

References

Abbadi, S., Senadza, B., Lieuw-Kie-Song, M. and Abebe, H. (2019) 'Assessing the employment effects of processing Cocoa in Ghana', *STRENGTHEN Publication Series. International Labour Office* [online] https://www. ILO. org/wcmsp5/groups/public/---ed_emp/---ifp_skills/documents/publication/wcms_673136. pdf (Data obrashcheniya 11.03. 2021).

Abu, I.O., Szantoi, Z., Brink, A., Robuchon, M. and Thiel, M. (2021) 'Detecting cocoa plantations in Côte d'Ivoire and Ghana and their implications on protected areas', *Ecological Indicators*, Vol. 129, No. 1, p.107863.

Ahmed, A.A. and Reddy, G.H. (2021) 'A mobile-based system for detecting plant leaf diseases using deep learning', *AgriEngineering*, Vol. 3, No. 3, pp.478–493.

Alagumariappan, P., Dewan, N.J., Muthukrishnan, G.N., Raju, B.K.B., Bilal, R.A.A. and Sankaran, V. (2020) 'Intelligent plant disease identification system using machine learning', in *Engineering Proceedings*, Multidisciplinary Digital Publishing Institute, Vol. 2, No. 1, p.49.

Barh, A. and Balakrishnan, M. (2018) 'Smart phone applicaGons: role in agri-informaGon disseminaGon', *Agricultural Reviews*, Vol. 39, No. 1, pp.82–85.

Boulent, J., Foucher, S., Théau, J. and St-Charles, P.L. (2019) 'Convolutional neural networks for the automatic identification of plant diseases', *Frontiers in Plant Science*, Vol. 10, p.941.

Chandra, A. (2019) 'Diagnosing the health of a plant in a click', in *Research into Design for a Connected World*, pp.593–601, Springer, Singapore.

Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y.A. (2020) 'Using deep transfer learning for image-based plant disease identification', *Computers and Electronics in Agriculture*, Vol. 173, No. 1, p.105393.

Chen, J-W., Lin, W-J., Cheng, H-J., Hung, C-L., Lin, C-Y. and Chen, S-P. (2021) 'A smartphone-based application for scale pest detection using multiple-object detection methods', *Electronics*, Vo. 10, No. 4, p.372, MDPI AG.

Chethan, K.S., Donepudi, S., Supreeth, H.V. and Maani, V.D. (2021) 'Mobile application for classification of plant leaf diseases using image processing and neural networks', in *Data Intelligence and Cognitive Informatics*, pp.287–306, Springer, Singapore.

Chollet, F. (2017) 'Xception: deep learning with depthwise separable convolutions', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp.1251–1258.

Delnevo, G., Girau, R., Ceccarini, C. and Prandi, C. (2021) 'A deep learning and social iot approach for plants disease prediction toward a sustainable agriculture', *IEEE Internet of Things Journal*, Vol. 9, No. 10, pp.7243–7250.

Dharmasiri, S.B.D.H. and Jayalal, S. (2019) 'passion fruit disease detection using image processing', in *2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, March, pp.126–133, IEEE

Diah, N.M., Jahim, M.L.M., Roslan, N.A. M., Ibrahim, Z. and Abdullah, A. (2021) 'Development of mobile application for plant disease recognition using convolutional neural network method', in *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Vol. 6, No. 1, pp.1–6, IEEE.

Dormon, E.V., Van Huis, A., Leeuwis, C., Obeng-Ofori, D. and Sakyi-Dawson, O. (2004) 'Causes of low productivity of cocoa in Ghana: farmers' perspectives and insights from research and the socio-political establishment', *NJAS: Wageningen Journal of Life Sciences*, Vol. 52, Nos. 3–4, pp.237–259.

Elhassouny, A. and Smarandache, F. (2019) 'Smart mobile application to recognize tomato leaf diseases using convolutional neural networks', in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, July, pp.1–4, IEEE.

Erlang [online] https://www.erlang.org/ (accessed 10 August 2023).

Esgario, J.G., de Castro, P.B., Tassis, L.M. and Krohling, R.A. (2021) 'An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning', *Information Processing in Agriculture*, Vol. 9, No. 1, pp.38–47.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A. (2010) 'The pascal visual object classes (VOC) challenge', *International Journal of Computer Vision*, Vol. 88, No. 2, pp.303–338.

Ferentinos, K.P. (2018) 'Deep learning models for plant disease detection and diagnosis', *Computers and Electronics in Agriculture*, Vol. 145, No. 1, pp.311–318.

Food and Agriculture Organization of the United Nations, 'FAOSTAT' (2022) [online] www.fao.org. (accessed 25 February 2022).

Francis, M. and Deisy, C. (2021) 'Mathematical and visual understanding of a deep learning model towards m-agriculture for disease diagnosis', *Archives of Computational Methods in Engineering*, February, Vol. 28, No. 3, pp.1129–1145.

Fuentes, A., Yoon, S., Kim, S.C. and Park, D.S. (2017) 'A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition', *Sensors*, Vol. 17, No. 9, p.2022.

Gandhi, R., Nimbalkar, S., Yelamanchili, N., and Ponkshe, S. (2018) 'Plant disease detection using CNNs and GANs as an augmentative approach', in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, May, pp.1–5, IEEE.

Garg, G., Gupta, S., Mishra, P., Vidyarthi, A., Singh, A. and Ali, A. (2021) 'CROPCARE: an intelligent real-time sustainable IoT system for crop disease detection using mobile vision', *IEEE Internet of Things Journal*, Vol. 10, No. 4.

Geetharamani, G. and Pandian, A. (2019) 'Identification of plant leaf diseases using a nine-layer deep convolutional neural network', *Computers and Electrical Engineering*, Vol. 76, No. 1, pp.323–338.

Google Maps SDK [online] https://developers.google.com/maps/documentation/android-sdk/overview (accessed 10 August 2023).

Hampf, A.C., Nendel, C., Strey, S. and Strey, R. (2021) 'Biotic yield losses in the Southern Amazon, Brazil: making use of smartphone-assisted plant disease diagnosis data', *Frontiers in Plant Science*, Vol. 12, No. 1, p.548.

Hassan, S.M., Maji, A.K., Jasiński, M., Leonowicz, Z. and Jasińska, E. (2021) 'Identification of plant-leaf diseases using CNN and transfer-learning approach', *Electronics*, Vol. 10, No. 12, p.1388.

He, K., Zhang, X., Ren, S. and Sun, J. (2016) 'Deep residual learning for image recognition', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp.770–778.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... and Adam, H. (2017) 'Mobilenets: efficient convolutional neural networks for mobile vision applications', arXiv preprint arXiv:1704.04861.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016) 'SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size', arXiv preprint arXiv:1602.07360.

Jang, G., Lee, J.W., Lee, J.G. and Liu, Y. (2020) 'Distributed fine-tuning of CNNs for image retrieval on multiple mobile devices', *Pervasive and Mobile Computing*, Vol. 64, No. 1, p.101134.

Kasper-Eulaers, M., Hahn, N., Berger, S., Sebulonsen, T., Myrland, Ø. and Kummervold, P.E. (2021) 'Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5', *Algorithms*, Vol. 14, No. 4, p.114.

Kothari, J.D. (2018) 'Plant disease identification using artificial intelligence: machine learning approach', *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 7, No. 11, pp.11082–11085.

Krizhevsky, A. (2014) 'One weird trick for parallelizing convolutional neural networks', arXiv preprint arXiv:1404.5997.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'Imagenet classification with deep convolutional neural networks', *Advances in Neural Information Processing Systems*, Vol. 25, No. 1, pp.1097–1105.

Li, Y., Nie, J. and Chao, X. (2020) 'Do we really need deep CNN for plant diseases identification?', *Computers and Electronics in Agriculture*, Vol. 178, No. 1, p.105803.

Lin, T. (2015) *LabelImg. Git code* [online] https://pypi.org/project/labelImg/ (accessed 10 August 2023)

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017a) 'Feature pyramid networks for object detection', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2117–2125.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017b) 'Focal loss for dense object detection', in *Proceedings of the IEEE International Conference on Computer Vision,* pp.2980–2988.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... and Zitnick, C.L. (2014) 'Microsoft coco: common objects in context', in *European Conference on Computer Vision*, September, pp.740–755, Springer, Cham.

Liu, J. and Wang, X. (2021) 'Plant diseases and pests detection based on deep learning: a review', *Plant Methods*, Vol. 17, No. 1, pp.1–18.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. (2016) 'SSD: single shot multibox detector', in *European Conference on Computer Vision,* pp.21–137, Springer, Cham.

Loyani, L. and Machuve, D. (2021) 'A deep learning-based mobile application for segmenting Tuta Absoluta's damage on tomato plants', *Engineering, Technology and Applied Science Research*, Vol. 11, No. 5, pp.7730–7737.

Magalhães, S. A., Castro, L., Moreira, G., Dos Santos, F.N., Cunha, M., Dias, J. and Moreira, A.P. (2021) 'Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse', *Sensors*, Vol. 21, No. 10, p.3569.

Mark, S. and Andrew, H. (2018) 'MobileNetV2: the next generation of on-device computer vision networks', *Google Research* [online] https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html (accessed 10 August 2023).

Mbarga, J.B., Begoude, B.A.D., Ambang, Z., Meboma, M., Kuaté, J., Ewbank, W. and Ten Hoopen, G.M. (2020) 'Field testing an oil-based Trichoderma asperellum formulation for the biological control of cacao black pod disease, caused by Phytophthora megakarya', *Crop Protection*, Vol. 132, No. 1, p.105134.

Mendes, J., Pinho, T.M., Neves dos Santos, F., Sousa, J.J., Peres, E., Boaventura-Cunha, J., ... and Morais, R. (2020) 'Smartphone applications targeting precision agriculture practices–a systematic review', *Agronomy*, Vol. 10, No. 6, pp.855.

Militante, S.V., Gerardo, B.D. and Medina, R.P. (2019) 'Sugarcane disease recognition using deep learning', in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, October, pp.575–578, IEEE.

Mohanty, S.P., Hughes, D.P. and Salathé, M. (2016) 'Using deep learning for image-based plant disease detection', *Frontiers in Plant Science*, Vol. 7, No. 1, p.1419.

Mrisho, L.M., Mbilinyi, N.A., Ndalahwa, M., Ramcharan, A.M., Kehs, A.K., McCloskey, P.C., ... and Legg, J.P. (2020) 'Accuracy of a smartphone-based object detection model, PlantVillage Nuru, in identifying the foliar symptoms of the viral diseases of cassava – CMD and CBSD', *Frontiers in Plant Science*, Vol. 11, No. 1, pp.1–14.

Ng, H.F., Lin, C.Y., Chuah, J.H., Tan, H.K. and Leung, K.H. (2021) 'Plant disease detection mobile application development using deep learning', in *2021 International Conference on Computer and Information Sciences (ICCOINS)*, July, pp.34–38, IEEE.

Ofori, A., Padi, F.K., Ameyaw, G.A., Dadzie, A.M., Opoku-Agyeman, M., Domfeh, O. and Ansah, F.O. (2022) 'Field evaluation of the impact of cocoa swollen shoot virus disease infection on yield traits of different cocoa (Theobroma cacao L.) clones in Ghana', *Plos One*, Vol. 17, No. 1, p.e0262461.

Owomugisha, G., Mugagga, P.K., Melchert, F., Mwebaze, E., Quinn, J.A. and Biehl, M. (2020) 'A low-cost 3-D printed smartphone add-on spectrometer for diagnosis of crop diseases in field', *in Proceedings of the 3rd ACM SIGCAS Conference on Computing and Sustainable Societies*, June, pp.331–332.

Oyewola, D.O., Dada, E.G., Misra, S. and Damaševičius, R. (2021) 'Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing', *PeerJ Computer Science*, Vol. 7, No. 1, p.e352.

Pallagani, V., Khandelwal, V., Chandra, B., Udutalapally, V., Das, D. and Mohanty, S.P. (2019) 'DCrop: a deep-learning-based framework for accurate prediction of diseases of crops in smart agriculture', in *2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*, December, pp.29–33, IEEE.

Park, H., JeeSook, E. and Kim, S.H. (2018) 'Crops disease diagnosing using image-based deep learning mechanism', in *2018 International Conference on Computing and Network Communications (CoCoNet)*, August, pp.23–26, IEEE.

Petrellis, N. (2018) 'A review of image processing techniques common in human and plant disease diagnosis', *Symmetry*, Vol. 10, No. 7, p.270.

Petrellis, N. (2019) 'Plant disease diagnosis for smart phone applications with extensible set of diseases', *Applied Sciences*, Vol. 9, No. 9, p.1952.

Pinto, L.A., Mary, L. and Dass, S. (2021) 'The real-time mobile application for identification of diseases in coffee leaves using the CNN model', in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, August, pp.1694–1700, IEEE.

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J. and Hughes, D.P. (2017) 'Deep learning for image-based cassava disease detection', *Frontiers in Plant Science*, Vol. 8, No. 1, p.1852.

Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., Ndalahwa, M., ... and Hughes, D.P. (2019) 'A mobile-based deep learning model for cassava disease diagnosis', *Frontiers in Plant Science*, Vol. 10, No. 1, p.272.

Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N. and Vinod, P.V. (2018) 'Plant disease detection using machine learning', in *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*, IEEE, April, pp.41–45.

Real, E., Aggarwal, A., Huang, Y. and Le, Q.V. (2019) 'Regularized evolution for image classifier architecture search', in *Proceedings of the AAAI Conference on Artificial Intelligence*, July, Vol. 33, No. 1, pp.4780–4789.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) 'You only look once: unified, real-time object detection', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779–788.

Ren, S., He, K., Girshick, R. and Sun, J. (2015) 'Faster r-cnn: towards real-time object detection with region proposal networks', *Advances in Neural Information Processing Systems*, Vol. 28, No. 1, p.1.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... and Fei-Fei, L. (2015) 'Imagenet large scale visual recognition challenge', *International Journal of Computer Vision*, Vol. 115, No. 3, pp.211–252.

Saleem, M.H., Potgieter, J. and Arif, K.M. (2019) 'Plant disease detection and classification by deep learning', *Plants*, Vol. 8, No. 11, p.468.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C. (2018) 'Mobilenetv2: Inverted residuals and linear bottlenecks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp.4510–4520.

Sanga, S., Mero, V., Machuve, D. and Mwanganda, D. (2020) 'Mobile-based deep learning models for banana diseases detection', arXiv preprint arXiv:2004.03718.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y. (2013) 'Overfeat: integrated recognition, localization and detection using convolutional networks', arXiv preprint arXiv:1312.6229.

Shafinah, K., Sahari, N., Sulaiman, R., Saif, A.S. and Ikram, M.M. (2020) 'm-Naslada: a mobile application for black pepper crop disease information', in *Journal of Physics: Conference Series*, April, Vol. 1529, No. 2, p.22041, IOP Publishing.

Shamkuwar, D.O., Thakre, G., More, A.R., Gajakosh, K.S. and Yewale, M.O. (2018) 'An expert system for plant disease diagnosis by using neural network', *International Research Journal of Engineering and Technology*, Vol. 5, No. 1, pp.369–372.

Shrimali, S. (2021) 'PlantifyAI: a novel convolutional neural network based mobile application for efficient crop disease detection and treatment', *Procedia Computer Science*, January, Vol. 191, No. 1, pp.469–474.

Sibanda, B.K., Iyawa, G.E. and Gamundani, A.M. (2021) 'Systematic review of plant pest and disease identification strategies and techniques in mobile apps', in *World Conference on Information Systems and Technologies*, Springer, Cham, March, pp.491–502.

Simonyan, K. and Zisserman, A. (2014) 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556.

Singh, K.K. (2018) 'An artificial intelligence and cloud based collaborative platform for plant disease identification, tracking and forecasting for farmers', in *2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pp.49–56, IEEE.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016) 'Deep neural networks based recognition of plant diseases by leaf image classification', *Computational Intelligence and Neuroscience*, No. 1, pp.1–12.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. and Rabinovich, A. (2015) 'Going deeper with convolutions', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp.1–9.

Tan, M. and Le, Q. (2019) 'Efficientnet: rethinking model scaling for convolutional neural networks', in *International Conference on Machine Learning,* May, pp.6105–6114, PMLR.

Tan, M., Pang, R. and Le, Q.V. (2020) 'Efficientdet: Scalable and efficient object detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp.10781–10790.

Tang, Z., Yang, J., Li, Z. and Qi, F. (2020) 'Grape disease image classification based on lightweight convolution neural networks and channelwise attention', *Computers and Electronics in Agriculture*, Vol. 178, No. 1, p.105735.

TensorBord [online] https://www.tensorflow.org/tensorboard (accessed 10 August 2023).

Valdoria, J.C., Caballeo, A.R., Fernandez, B.I.D. and Condino, J.M.M. (2019) 'iDahon: an android based terrestrial plant disease detection mobile application through digital image processing using deep learning neural network algorithm', in *2019 4th International Conference on Information Technology (InCIT)*, October, pp.94–98, IEEE.

Verma, S., Chug, A., Singh, A.P., Sharma, S. and Rajvanshi, P. (2019) 'Deep learning-based mobile application for plant disease diagnosis: a proof of concept with a case study on tomato plant', in *Applications of Image Processing and Soft Computing Systems in Agriculture*, pp.242–271, IGI Global.

Weather API [online] https://openweathermap.org/api (accessed 10 August 2023).

Yan, B., Fan, P., Lei, X., Liu, Z. and Yang, F. (2021) 'A real-time apple targets detection method for picking robot based on improved YOLOv5', *Remote Sensing*, Vol. 13, No. 9, p.1619.

Yang, M., Li, D., Chen, M., Bouras, A., Tang, Y. and Yu, X. (2020) 'The implementation of a crop diseases APP based on deep transfer learning', in *2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, May, pp.22–28, IEEE.

Yu, H. et al. (2020) *TensorFlow Model Garden* [online] https://www.tensorflow.org/guide/model_garden (accessed 10 August 2023).

Zhou, X., Austin, U.T., Wang, D., Berkeley, U.C. and Austin, U.T. (2019) 'Object as point', *Cvpr.*, [online] arxiv.org (accessed 10 August 2023).