Sustainable Plant Pathogen Detection: Balancing Accuracy and Energy Efficiency in Deep Learning Models

MSc Research Project
MSc in Artificial Intelligence

Aishwarya Bediskar
Student ID: x23144441

School of Computing
National College of Ireland

Supervisor: Dr. Mayank Jain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aishwarya Bediskar
Student ID:	x23144441
Programme:	MSc in Artificial Intelligence
Year:	2023-24
Module:	MSc Research Project
Supervisor:	Dr. Mayank Jain
Submission Due Date:	12/08/2024
Project Title:	Sustainable Plant Pathogen Detection: Balancing Accuracy and Energy Efficiency in Deep Learning Models
Word Count:	4908
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Sustainable Plant Pathogen Detection: Balancing Accuracy and Energy Efficiency in Deep Learning Models

Aishwarya Bediskar
x23144441

Abstract

In the realm of sustainable agriculture, accurate and energy-efficient plant pathogen detection is crucial for crop health, environmental sustainability, and economic viability. This study investigates sophisticated machine learning methods for the identification of plant pathogens from images, with a particular emphasis on deep learning models such as DenseNet121, VGG19, and Convolutional Neural Networks (CNN). In order to maintain environmental sustainability, I built and assessed these models to determine their accuracy in diagnosing plant illnesses as well as their carbon footprint. A dataset consisting of 2,025 photos that were classified as bacteria, viruses, fungus, healthy, and pests was used in the study. According to the results, DenseNet121 performs better than CNN and VGG19, obtaining the maximum accuracy of 98% with the least amount of loss (0.068). DenseNet121 is the most ecologically friendly model as it also shows the lowest carbon emissions. DenseNet121 has a rather low carbon emission of 0.0003 kg CO2 per epoch, proving the efficiency of the model for large-scale applications in precision agriculture. The results depicted DenseNet121 as the most reliable and greenest among the models studied.

Keywords- Plant Pathogen Detection, Deep Learning, Energy Efficiency, Carbon Footprint, Sustainability

1 Introduction

Plant pathogen detection has become a critical focus area in sustainable agriculture, demanding accuracy and efficiency, because it guarantees crop health, lowers disease-related losses, and uses less harmful pesticides, all of which support environmental and financial sustainability. Early response and efficient treatment of plant diseases depend on accurate diagnosis. There is a lot of opportunity to improve detection accuracy and response times by integrating advanced machine learning and transfer learning models into this domain. But these models are quite challenging, especially when it comes to how much power they need for RAM, GPUs, and CPUs. As agriculture sector moves towards precision and digital farming, it is necessary to develop pathogen detection systems that are not only effective but also sustainable in terms of energy usage.

The significance of energy efficiency is highlighted by the growing dependence of plant pathogen detection models on high-performance computational resources. Because of their capacity for parallel processing, GPUs are essential for managing complicated calculations, but they also use a lot of energy. In the same way, the constant use of CPUs and RAM for model training and inference raises the total energy load. To mitigate these effects, sustainable computing methods are essential. These practices include the adoption of optimised model designs, effective training algorithms, and real-time power monitoring.

Deep learning technologies have made image processing and categorisation easier to manage and have produced improved accuracy results. Deep neural networks (DNNs) have shown ground-breaking efficacy in a number of tasks that were previously thought to be unachievable in recent years LeCun et al. (2015); Alom et al. (2019). It costs a lot of money to train a DNN effectively on a big number of samples. The energy consumption of machine learning training tasks rises with the length of the model training period. Because high-performance GPUs are frequently used to train DNN workloads yet are less energy-efficient than their CPU equivalents, this is a significant challenge for them. Therefore, it is imperative to lower GPU energy usage when training machine learning workloads. Reaching this objective will enable consumers to train larger models for the same amount of money and lower maintenance costs.

1.1 Research Questions and Objectives

1.1.1 Research Questions:

1. How to develop a plant pathogen detection model from images which has high accuracy?

2. How to ensure that the plant pathogen detection model works and has a low carbon footprint?

3. How to strike a balance between accuracy and energy efficiency of such models?

1.1.2 Objectives:

The primary objectives of this research are:

- **Develop an Accurate Plant Pathogen Detection Model:** To develop a high-accuracy machine learning model for plant pathogen identification from photos that would enable early and accurate diagnosis and promote sustainable agriculture by lowering crop losses and the requirement for dangerous chemicals.

- **Minimize the Carbon Footprint of Detection Models:** To guarantee that the plant pathogen detection model runs with a minimal carbon footprint, energy efficiency in computational procedures and the application of sustainable computing techniques are crucial.

Deep Learning Models are becoming increasingly important for efficient disease identification as agricultural technology moves towards more automated and accurate systems. These models provide better accuracy and efficiency by utilising powerful GPUs and large amounts of processing power. However, there is an environmental cost associated with these technologies' energy requirements. The total greenhouse gas emissions of the detecting systems are directly impacted by the carbon footprint associated with the operation of GPUs, CPUs, and RAM.

The concept of carbon footprint is essential, when evaluating and controlling the environmental effect of many activities such as agricultural advancements in technology. The whole amount of greenhouse gas (GHG) emissions from an entity's activities or processes is measured by its carbon footprint, mostly from carbon dioxide (CO) but also from methane (CH) and nitrous oxide (NO). This parameter is becoming more important in the area of plant pathogen detection since machine learning algorithms require a large amount of energy to train and implement.

Several crucial methods are involved in integrating carbon reduction measures into the creation and use of machine learning models:

- **Optimising Model Efficiency:** In order to minimise computing resources, energy consumption and related emissions, it is possible to develop lightweight model structures and energy-efficient methods.
- **Implementing Sustainable Computing Practices:** Machine learning operations may drastically reduce their carbon footprint by utilising energy-efficient hardware, enhancing power management, and integrating renewable energy sources to power data centres.
- **Monitoring and Reporting:** Building up reliable systems for constant assessment and modification of activities to reduce environmental effect is made possible by the real-time tracking of energy use and carbon emissions.

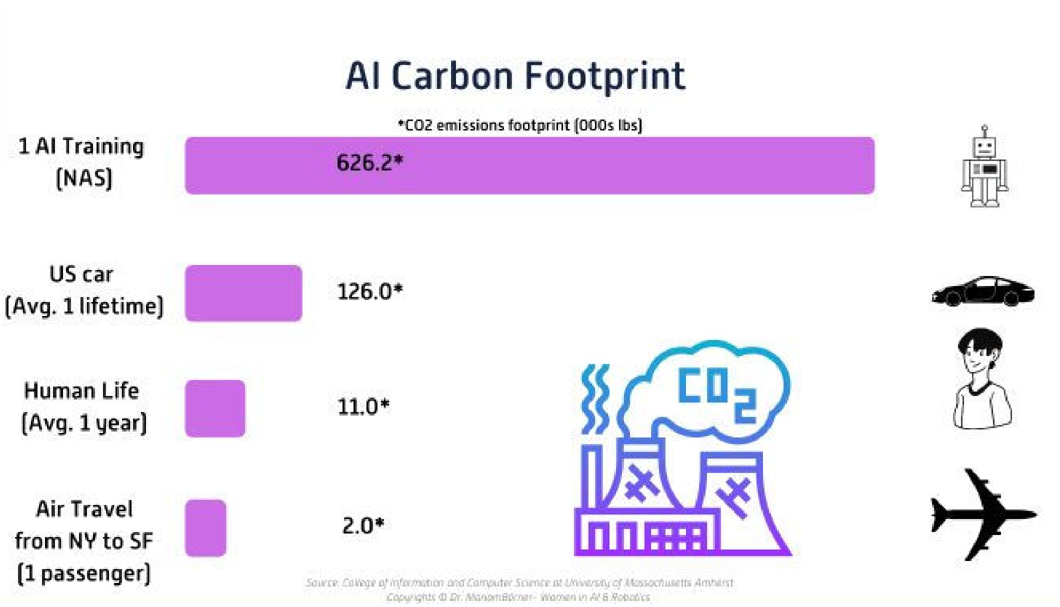

Figure 1: AI Carbon Footprint yahoo.com (n.d.)

2 Related Work

Sustainable plant pathogen detection is becoming more crucial as the agriculture sector finding out to balance technological advancements with environmental responsibility. In order to find out the model which takes lower energy consumption of RAM, GPUs and CPUs in plant pathogen detection, this paper focusing on advanced machine learning techniques which is more sustainable.

In this study, researcher Yadav et al. (2024) compares three of the most powerful convolutional neural network (CNN) architectures—VGG16, VGG19, and ResNet—in general for the classification of leaf diseases, with a focus on black rot, ESCA, leaf blight, and healthy leaves. This study examines the capacity of the algorithms to accurately identify the illnesses from the visual symptoms using a very large dataset of grapevine leaf photos collected from various locations. Based on the study findings, it is evident that the ResNet model outperforms its competitors with an exceptional accuracy of 95%. VGG19 and VGG16 follow closely behind with 93.5% and 92%, respectively. These findings highlight the effectiveness of deep residual learning and the significance of architectural depth in guaranteeing CNN accuracy for tasks involving the categorisation of plant diseases.

Arathi and Dulhare (2023) In this paper, in order to detect plant diseases, CNN is utilised for feature extraction. There is a problem with these conventional CNN algorithms' accuracy. The experimental findings shown that the pre-trained DenseNet-121 model, which is the suggested model, can categorise various leaf pictures in the dataset with a better classification accuracy of 91%. With the use of ImageNet weights, this transfer learning method can effectively identify cotton plant illnesses.

Rithik P et al. (2023) In the proposed study, transfer learning based on the Convolutional Neural Network model VGG-19 is used for predicting the classes of illnesses. This can help reduce yield losses by up to 30% if the identification of illness is done early. Low-level spatial information can be retrieved using CNN-based models such as VGG-19. The VGG-19 was pre-trained using the CIFAR 100 dataset and the ImageNet dataset, which has more than a thousand classifications. It achieves an accuracy of 94% for VGG-19.

The study carried out by J. Chen et al. is among the studies connected to transfer learning in the field of plant pathology. The plants under study were rice and maize Chen et al. (2020b). The study's findings show that, in situations with extremely complicated object backgrounds, their suggested model achieves an average accuracy of 92%. The identification of abnormalities in millet may also be resolved with the use of transfer learning Coulibaly et al. (2019). An accuracy of 95%, precision of 90.50%, recall of 94.50%, and f1-score of 91.75% are obtained from the identification results using the created model. The application of transfer learning to improve both the quality and quantity of production in the field of plant pathology is still the subject of several research Thenmozhi and Reddy (2019); Aravind et al. (2019); Kamal et al. (2019). In order to categorise abnormalities discovered in potato leaves, researchers in this work also want to employ transfer learning from a trained CNN architecture, such as VGG-16.

Wang et al. (2022) This paper tackles the growing energy requirements of contemporary models by introducing an online GPU energy optimisation framework (GPOEO) for machine learning workloads. GPOEO dynamically optimises energy setups by utilising GPU performance counters in conjunction with an analytical model to significantly reduce profiling overhead. To balance execution time and energy usage, the framework makes use of local search algorithms and multi-objective prediction modelling. Test results reveal that, as compared to NVIDIA's default scheduling, GPOEO provides a 16.2% mean energy savings with just a 5.1% increase in execution time.

Priya et al. (2023) In this study researchers discuss the substantial greenhouse gas emissions from India, along with the anticipated rise in emissions in spite of attempts to reduce them. The authors focus on the carbon footprint which is essential and effectively tackle climate change. They draw attention to India's pledge under the Paris Agreement to cut its carbon GHG emission intensity by 33–35% by 2030. The study also looks at machine learning models and IoT devices used to monitor and forecast greenhouse gas emissions in real time. Birch and LSTM models are used to analyse the environmental conditions of the present and the future.

Huang and Mao (2024) This paper suggests a thorough framework for analysing, optimising, and tracking carbon footprints at different supply chain stages using artificial intelligence. The suggested method collects, processes, and analyses enormous volumes of data about carbon emissions, including industrial and transportation, using AI algorithms. The framework identifies important areas for emission reduction and creates plans to minimise environmental effect while preserving operational efficiency by utilising machine learning and optimisation approaches. This strategy facilitates proactive decision-making through real-time monitoring and predictive analytics, enabling businesses to swiftly adjust to shifting market dynamics and environmental requirements. The use of artificial intelligence (AI) not only improves the precision and dependability of carbon footprint evaluations but also offers insights for ongoing development and monitoring sustainability performance.

3 Methodology

3.1 Data Acquisition

The dataset utilised in this work was gathered from the Kaggle website and is openly accessible to everyone KaggleLink (n.d.). 2,025 photographs total from this collection were gathered for plant pathogen categorisation. The five classifications that make up the plant pathogen dataset are: Bacteria, Viruses, Fungi, Healthy and Pests. The collection includes a variety of images of plant leaves that have been impacted by various diseases. Every image in the collection shows a leaf of a plant, either healthy or diseased. To accurately identify the illness on the leaf, proper labelling will be utilised. A number of methods, including Transfer Learning and Convolutional Neural Networks, will be used to identify plant diseases. The example leaf photos from the collection that are infected with different diseases are shown below, along with a table that display the number of images for each class in the plant pathogen dataset.

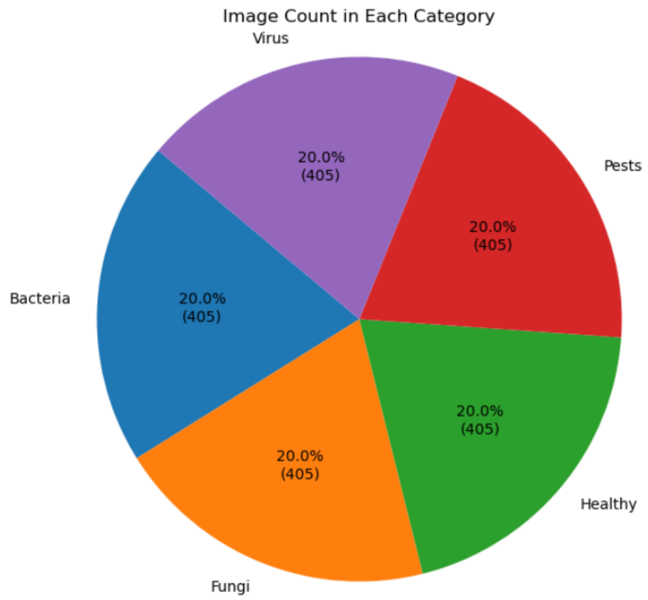

Figure 2: Distribution of Image Categories for Plant Pathogen Detection

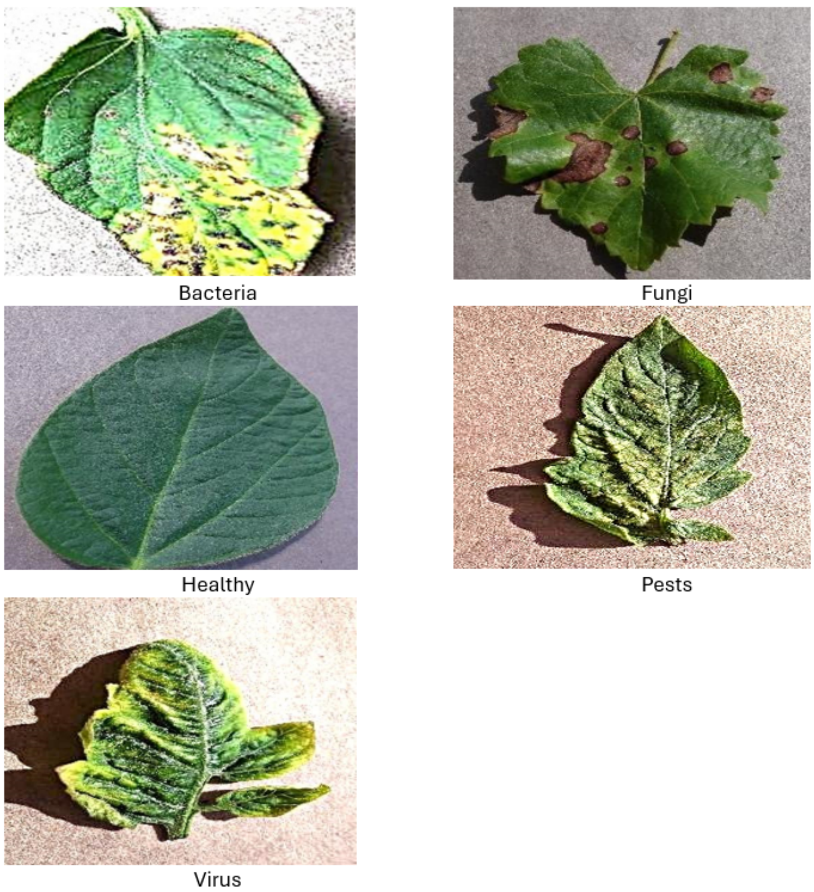

Figure 3: Sample Photos of Leaves from the Plant Pathogen Database

3.2 Data Preprocessing

The preprocessing stage involves a number of transformations applied to the image dataset in order to prepare it for model training. Following dataset import, photos are initially scaled up to a standard target size. To make sure that every image is in the same proportions, which the neural network needs. Following that, the pixel value for a picture is divided by 225 to normalise to a range of 0 or 1, which expedites the training procedure and improves the performance of the model.

The dataset is split into 80-20 training testing, make sure the model is trained on one part of the data and verified in another. Categorical labels are converted into numbers by encoding the labels suitable for classification tasks. Finally, once the data has been scaled propagated however it needed to be predicted for the neural network using training and test splits. This preprocessing pipeline ensures the input data is how best it could be used during training to further support model performance.

3.3 Data Modelling

In the data modelling phase, I implemented VGG19, DenseNet121 and Convolutional Neural Network (CNN) to classified plant pathogen images into five disease categories. Pixel values were normalised, and images were scaled to 64x64 pixels as part of the dataset preparation. I improved the model's generality by adding flipping, rotating, shifting, and zooming to the training set. Global average pooling, dropout, and batch normalisation were used in the model's construction to reduce overfitting. Categorical cross entropy loss function and Adam optimiser were used to put the method together. For adjusting the learning rate and preventing overfitting, I employed callbacks such as ReduceLROnPlateau and Early Stopping. A batch size of 64 was used during the model's 50 epochs of training. Final performance indicated how well the model distinguished between different pathogen types. Evaluation measures included loss and accuracy on a different validation set.

3.4 Model Evaluation

Model evaluation will assist in determining the model's actual capacity to provide the intended result. In the evaluation step, several important indicators were added in the assessment of plant pathogen detection algorithm to guarantee its accuracy and robustness. A thorough breakdown of the right and wrong predictions made in each of the five classes was provided via a confusion matrix. I also plotted accuracy and loss throughout the training process, plotted these metrics to see the learning curve and identify any signs of overfitting or underfitting.

3.5 Carbon Emission Analysis

Once the model has been evaluated, I calculated the energy usage of the RAM, GPUs, and CPUs utilised for training and inference to assess the model's environmental effect. The carbon footprint of the model is then evaluated by estimating the related CO2 emissions. This stage emphasises how crucial it is to take sustainability into account while working on machine learning models.

4 Design Specification

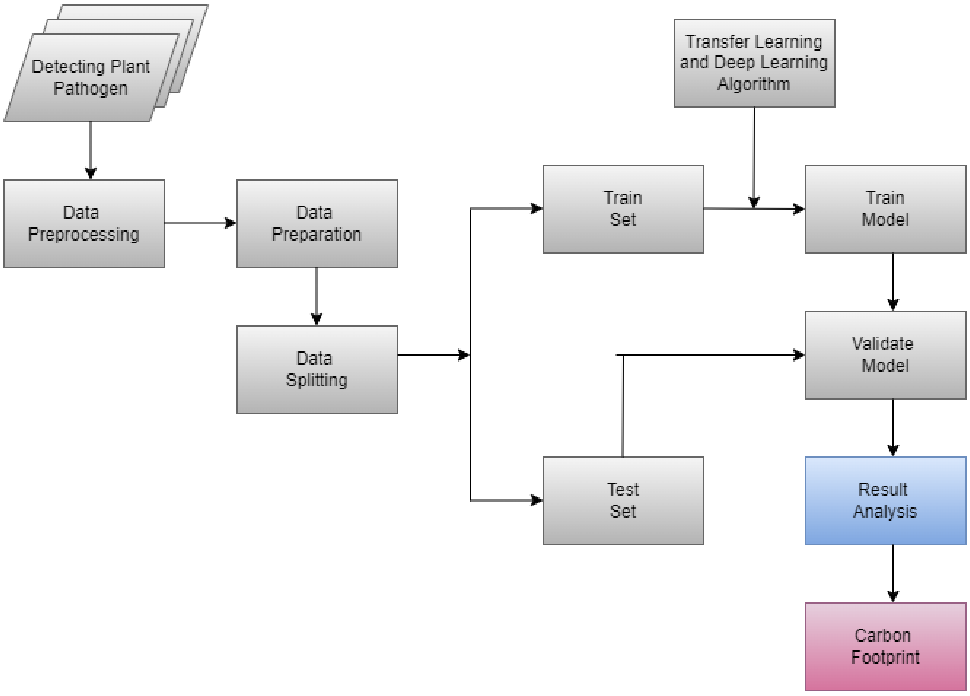

Figure 4: Design Specification for Plant Pathogen Detection

Figure 4 presents the detailed workflow of the plant pathogen detection project. Part 3 provides specifics on the preliminary steps, which include data collection, data preprocessing, and data splitting. For model training, validation, and assessment, the next steps entail using Transfer Learning and Deep Learning algorithms, as described in section 5. To guarantee both precision and energy economy in pathogen detection, the project closes with a result analysis and carbon footprint evaluation.

5 Implementation

A system with a 2.30 GHz 12th Gen Intel(R) Core(TM) i7-12700H, 16 GB of RAM, 6 GB Nvidia GeForce RTX3060 graphics card, and 512 GB SSD storage is being used for this research. Given the complexity of processing pictures with the above architectures, this system configuration is appropriate for the research since it requires a significant amount of computational power to run CNN, DenseNet121, and VGG19 models and process images. Python programming language is used to create code, and Jupyter Notebook is the tool used to run the code with a separate environment established for the project with needed libraries so that there is no issue of other conflicting libraries in the environment. This section will go into great depth on the implementations and assessments of the specially-made optimised CNN, DenseNet121, and VGG19.

5.1 Implementation of CNN Architecture

CNN, commonly known as the CovNet method, falls under DL to identify plant diseases LeCun et al. (2015). Thus, deep learning, often known as DL LeCun et al. (2015), is a subfield of artificial intelligence (AI) in which a computer or system imitates human behaviour and the process by which people learn.

According to the labels, the preprocessed and altered images that are kept in the train and test folders are read and saved with a 150 pixel image size in this CNN architecture. Because the validation set is sufficiently large, it is used at a ratio of 0.2 for the training set.

- **Input Shape:** The input data has a constant shape of (150,150,3), with three channels since the image is coloured. This shape was decided after examining how various picture dimensions affected the model's performance.
- **Convolutional Layer:** There are four convolutional layers in the CNN architecture. To maintain the input dimensions, the first layer has the 'same' amount of padding. The first two levels employ 32 filters, each measuring 3x3. The third layer has the 'same' amount of padding, whereas the next two levels employ 64 filters of size 3x3. MaxPooling2D is performed after the second layer to minimise spatial dimensions, while the activation function 'relu' is employed throughout to add non-linearity.
- **Pooling Layer:** Max pooling is used to try to minimise overfitting in any scenario, and it will also assist to shorten the total amount of time the model runs. In order to capture the most important aspects of the picture, a stride size of two is used here.
- **Rectified Linear Unit Activation Function(ReLU):** ReLU is being utilised to increase the model's ability to comprehend images. In order to transform the current negative values to zero and eliminate their influence, ReLU is also employed to maintain the resultant output at non-zero values. Early stages will also gain from learning important information throughout the network's backpropagation.
- **Flatten Layer** This layer is used to create a flat array from the n-dimensional array output seen above. The input for the following layer will be this array that has been flattened.
- **Fully Connected Layers:** There are two fully connected layers at the end to obtain the prediction as either diseased or healthy. In the initial fully connected layer, 512 neurones are employed, and inconsistencies are eliminated using ReLU. Each layer is generated as a matrix by multiplying the input layer by the weights, which are once again determined dynamically in both levels based on the information that is provided to them.
- **Adam Optimizer:** Its ability to manage sparse gradients on photos with noisy data has led to the employment of this optimiser.

Above layers include various tunable parameters, a cross entropy function to assist eliminate minute mistakes, and an early stopping feature to enable the model to halt the process if the validation scores did not increase. Several runs of the model have been conducted with varying batch size settings (32, 64, and 512), optimiser functions, input picture sizes, and weight and bias calculation methodologies.

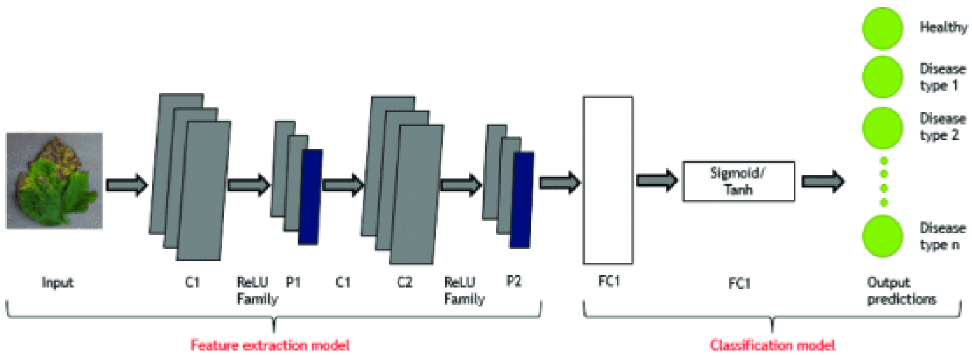

Figure 5: Convolutional Neural Network Architecture Biswas and Yadav (2023)

5.2 VGG19

The VGG 19 model developed by the Visual Graphics Group at Oxford has been one of most impactful Convolutional Neural Network (CNN) architecture opengenus (n.d.). It has 19 layers with trainable weights, and is deeper than its predecessor VGG16. VGG19 is trained on more than a million images from the ImageNet database, and can distinguish between 1000 different categories of objects. It is a simple and effective model of architecture where it applies small 3×3 convolution filters throughout the network in general, along with layers for spatial mass reduction such as max-pooling stratum which repeatedly shrinks by 2 times. Because the VGG19 architecture is praised for its balance between depth and computational efficiency, it has become a popular option to fine-tuning pre-trained models geeksforgeeks (2024).

The VGG19-based model leverages a pre-trained VGG19 network from Keras, dropping its top layers and use ImageNet-trained weights. The input shape is set to (150, 150, 3). To operate as feature extractors, all of the basic model's layers are frozen. The expanded model consists of a dropout layer (0.3), flattening, and dense layers of 512, 256, and 128 units using ELU activation. Five units with softmax activation are present in the output layer for categorisation. In order to avoid overfitting, the model is trained with early stopping and constructed using the Adam optimiser and categorical cross-entropy loss.

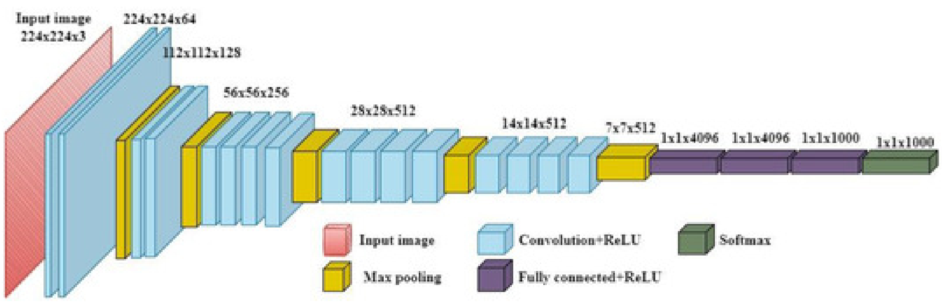

Figure 6: VGG19 Architecture Nguyen et al. (2022)

5.3 DenseNet121

DenseNet, also known as the Dense Convolutional Network, is a noteworthy development in deep learning techniques. This approach uses a convolutional procedure, which enables more in-depth picture analysis and a quicker image recognition computation process. As a transfer learning model, DenseNet121 integrates pre-trained models learnt using large datasets like ImageNet and C. DenseNet121 transmits the output feature map from the previous layer to the next input layer by connecting each convolutional layer to the one after it, a technique known as a dense block Chen et al. (2020a).

In the implementation of DenseNet121 model, DenseNet121 is used as a feature extractor using pre-trained ImageNet weights in the architecture of the DenseNet121 model, excluding the top classification layer which is not being used in the implementation. The DenseNet121 backbone is added to this model after a convolutional layer with three filters and a 3x3 kernel size. The output is subjected to global average pooling by following batch normalisation and dropout for regularisation. Afterwards a fully linked dense layer with 256 units and ReLU activation comes a batch normalisation and dropout layer. There are 15 units in the final output layer that have a softmax activation function for multi-class classification. Categorical crossentropy loss and the Adam optimiser are used to construct the model, and data augmentation approaches are used to train it to increase generalisation.

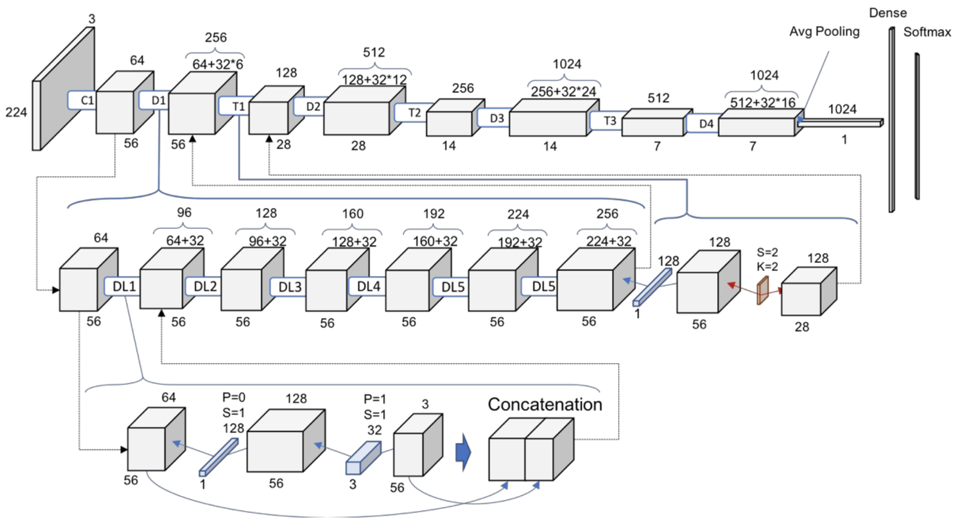

Figure 7: Full schematic representation of DenseNet-121 Ruiz (2018)

5.4 Carbon Footprint

In the implementation of carbon emission, CodeCarbon Library is used for measuring carbon emission during a machine learning model is being trained. EmissionsTracker class starts first by initializing and then starting before fitting the model to the training set of data. The tracker does the tracking of CO2 emissions as the model trains, with its performance evaluated every epoch and modified. Finally, when the training is done, the tracker stops and computes the total amount of CO2 emissions. The number of epochs in the analysis of the emissions data is used to obtain the average per epoch by dividing the total amount by epoch size. The data is presented graphically on a plot, where the emitted values for every epoch are constants. The graph represents the CO2 emissions related to model training visible shows in section 6, which helps to comprehend the understanding the environmental effect of machine learning processes is made easier by the map, which shows the CO2 emissions linked to model training. This method aids in measuring and evaluating the sustainability of computational techniques.

6 Evaluation

6.1 Evaluation for CNN

The accuracy graph indicates how well the model is learning and if there is any overfitting. The graph represents how the model's validation and training accuracy changed across epochs. The model's training and validation loss trends are displayed on the loss graph, which makes it easier to see how effectively the model is converging and generalising. Increased accuracy and reduced loss during the training time usually signal the proper fine-tuning of the model, it means that those during the training period the model will become more accurate and have lower loss at the end.

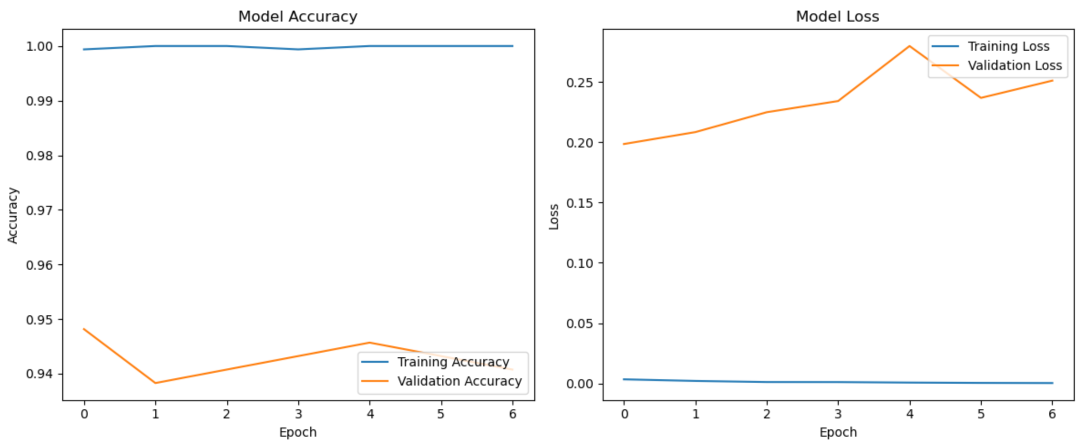

Figure 8: Model Accuracy and Loss for CNN

The model's performance is displayed in the confusion matrix as true positives, true negatives, false positives, and false negatives. Many metrics that indicate how successfully the classes are discriminated are computed, including precision, recall, and the F1-score. The confusion matrix shows the performance of a classification model across five classes: Bacteria, Fungi, Healthy, Pests, and Virus. The diagonal values indicate correct classifications, with high accuracy observed for Fungi (83) and Healthy (79).

The ROC curve illustrate the model's performance in distinguishing between classes and it plots the true positive rate against the false positive rate across different thresholds. Greater overall discrimination and classification capabilities of the model is shown by a bigger area under the curve (AUC).

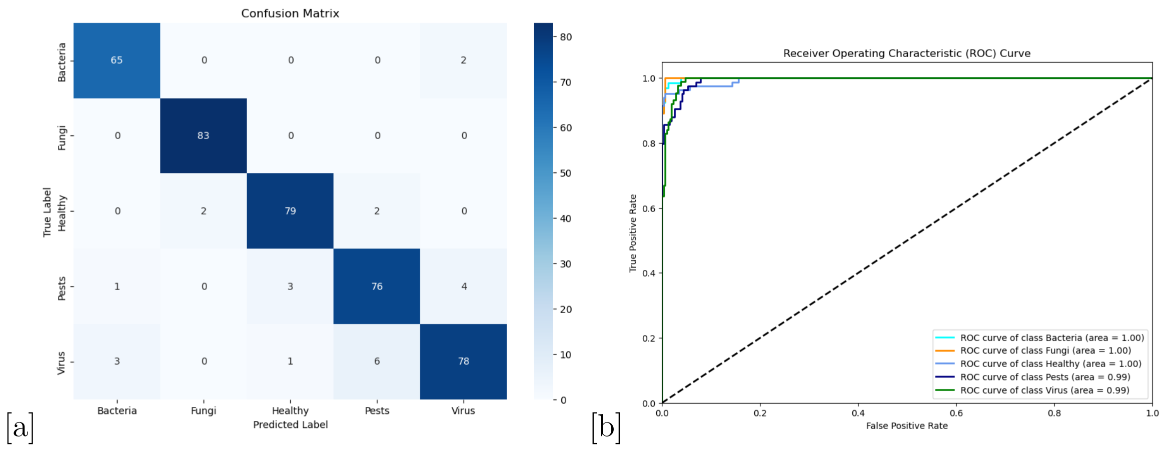

Figure 9: (a) Confusion Metrix for CNN (b) ROC Curve for CNN

6.2 Evaluation for DenseNet121

In terms of plant disease classification, the DenseNet121 model performs well, the confusion matrix indicates that most classes — most notably Bacteria (97) and Fungi (84)—have excellent accuracy. There is some misunderstanding between the Virus and Healthy classes while some Virus instances are mistakenly assigned to the Healthy class.

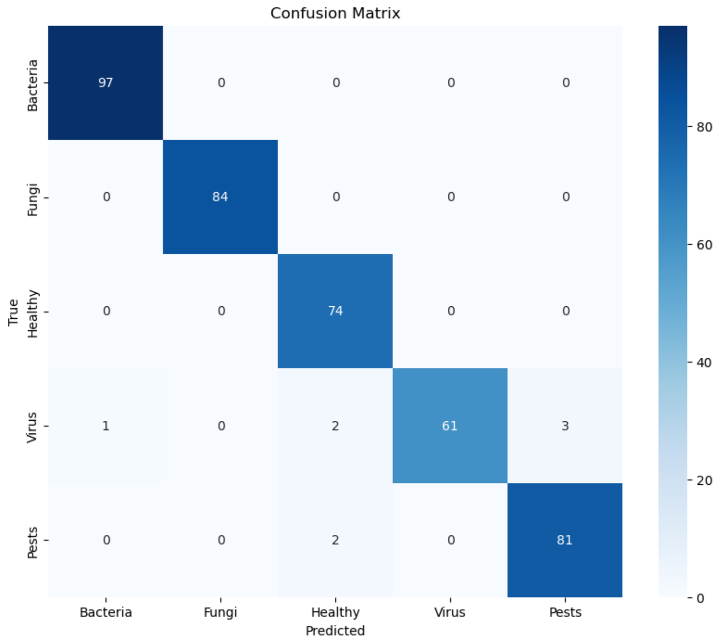

Figure 10: Confusion Matrix for DenseNet121

In the case of DenseNet121, there is a sharp increase of both train and validation accuracy to over 90% in about 20 epochs. However, the validation loss increases from 30 epochs onwards, which is an obvious case of overfitting, while the training loss linearly continues to drop.

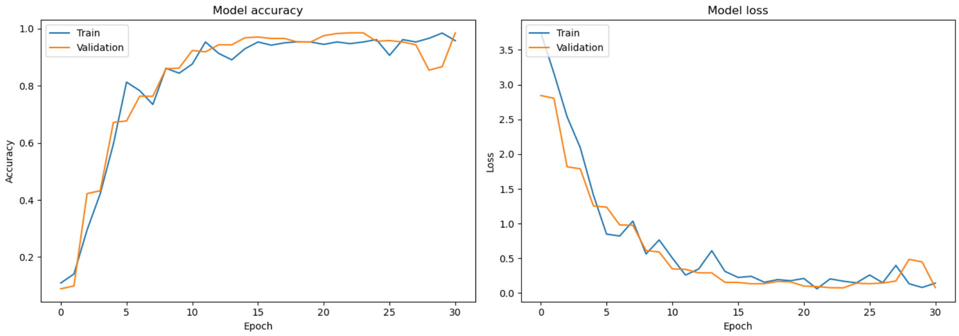

Figure 11: Validation Accuracy and Loss graph for DenseNet121

In the training phase, DenseNet121 emitted approximately 0.0003 kg of CO2, which is negligible. Carbon impact was measured and tracked during the completion of the model's 50 epochs using the CodeCarbon package. Stability in emissions across epochs showed how the model appropriately used its computer resources.

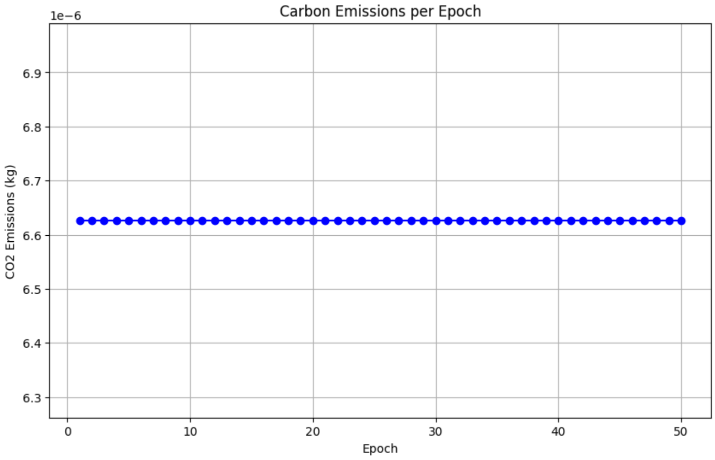

Figure 12: Carbon Emission for DenseNet121

6.3 Evaluation for VGG19

The image 13 is a graphical representation of actual versus predicted classes for the different categories of leaf diseases: fungi, bacteria, pests, viruses, and healthy leaves. Most of the predictions were similar to the actual labels, thus showing strong model performance, except in one case of misclassification, where a virus-affected leaf was misclassified as being affected by pests. This gives a good representation of both the overall accuracy and an area of improvement for the predictive model.

Figure 13: Plant Disease Prediction using VGG19

The confusion matrix clearly portrays the performance of the VGG19 model in the classification of the diseases on the leaves. It fits most of the classes accurately and shows minimal misclassification, mostly between the "Pests" and "Virus" classes.

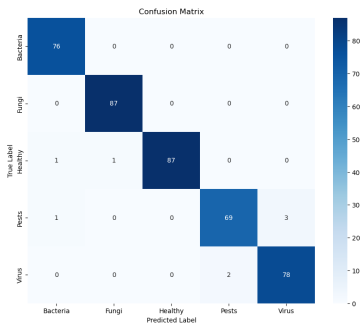

Figure 14: Confusion Matrix for VGG19

During the model training procedure, about 0.0071 kg of CO2 were released overall. To get an estimate for each period, the carbon emissions are averaged over all epochs. The environmental impact of the model's training is reflected in this graphic, which shows the emissions for each epoch.

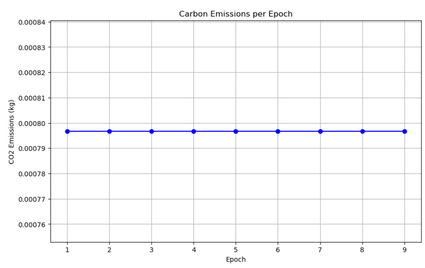

Figure 15: Carbon Emission for VGG19

6.4 Discussion

The three models—CNN, VGG19, and DenseNet121—were evaluated, and the results provide significant additional details on how effectively they work in agricultural applications, especially when it comes to correctly recognising plant diseases. The results highlight the advantages of the present concept and execution, despite the experiments' meticulous planning to guarantee that every model was evaluated using the same dataset and under the same conditions.

	Accuracy	Loss	CO2 Emission
CNN	95%	0.177	0.0012 Kg
VGG19	98%	0.079	0.0071 Kg
DenseNet121	98%	0.068	0.0003 Kg

This CNN model, though simpler and more computationally efficient, has very poor accuracy and loss compared to VGG19 and DenseNet121. In particular, CNN was able to achieve an accuracy of 95% with a loss of 0.177, which is far higher in comparison to the losses obtained in the more advanced models. That means that even though CNN might be faster to put into action, it still lacks reliability in very important tasks such as pathogen detection, where high accuracy is required. The higher loss does give an indication that the CNN model is less confident in its predictions, which could lead to more misclassifications in a real-world scenario.

On the other hand, DenseNet121 and VGG19 both show higher accuracy of 98%, while DenseNet121 has the least amount of loss, 0.068. Hence, DenseNet121 is now the most reliable model among the three.

Moreover, DenseNet121 is also more environmentally efficient, with CO emissions of 0.0003 Kg, in comparison with VGG19, which has a 0.0071 Kg emission. Reduced carbon emissions show DenseNet121 to be much more accurate, more reliable, and more sustainable for large-scale deployment. Where energy efficiency and environmental impact are critical, DenseNet121 is much more balanced in terms of high performance with sustainability and would be preferable over both VGG19 and CNN.

7 Conclusion and Future Work

In conclusion, this paper gives an overview of the main role advanced deep learning models can play in detecting accurate and energy-efficient plant pathogens for sustainable agriculture. In this study, VGG19 and DenseNet121 provided an outstandingly high accuracy in plant pathogen detection. Among the tested models in this research, DenseNet121 showed top performance, providing the highest accuracy of 98% and having the lowest carbon emissions at the same time. First, this makes DenseNet121 the most reliable model in relation to disease detection; it is also the most environmentally friendly, having its carbon emissions per epoch standing at 0.0003 kg CO2e. In contrast, VGG19 emitted 0.0071 kg CO2 every epoch, which is way more in comparison to the 0.0003 kg CO2 by DenseNet121, whereas the CNN model, although more energy-efficient, returned an accuracy of just 95%.

Future studies could focus on the application of DenseNet121-based models in the detection of multiple other stresses in plants apart from pathogens, including lack of nutrients, drought, and/or pest-related infestations. Multi-spectral or hyperspectral image data can be used to improve the accuracy of stress detection at presymptomatic stages of the disease process. Even better diagnostic tools for farmers could be made available through a coupling of machine learning models with expert systems or knowledge-based approaches, enabling the performance of precise and timely interventions against agricultural threats.

References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. and Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures, *electronics* **8**(3): 292.

Arathi, B. and Dulhare, U. N. (2023). Classification of cotton leaf diseases using transfer learning-densenet-121, *Proceedings of third international conference on advances in computer engineering and communication systems: ICACECS 2022*, Springer, pp. 393–405.

Aravind, K., Raja, P., Aniirudh, R., Mukesh, K., Ashiwin, R. and Vikas, G. (2019). Grape crop disease classification using transfer learning approach, *Proceedings of the International conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, Springer, pp. 1623–1633.

Biswas, B. and Yadav, R. K. (2023). A review of convolutional neural network-based approaches for disease detection in plants, *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 514–518.

Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y. (2020a). Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture* **173**: 105393.
URL: *https://www.sciencedirect.com/science/article/pii/S0168169919322422*

Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y. A. (2020b). Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture* **173**: 105393.

Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D. and Traore, D. (2019). Deep neural networks with transfer learning in millet crop images, *Computers in industry* **108**: 115–120.

geeksforgeeks (2024). VGG-Net Architecture Explained - GeeksforGeeks — geeksforgeeks.org, `https://www.geeksforgeeks.org/vgg-net-architecture-explained/`. [Accessed 11-08-2024].

Huang, R. and Mao, S. (2024). Carbon footprint management in global supply chains: A data-driven approach utilizing artificial intelligence algorithms, *IEEE Access* **12**: 89957–89967.

KaggleLink (n.d.). https://www.kaggle.com/datasets/kanishk3813/pathogen-dataset.

Kamal, K., Yin, Z., Li, B., Ma, B. and Wu, M. (2019). Transfer learning for fine-grained crop disease classification based on leaf images, *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE, pp. 1–5.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *nature* **521**(7553): 436–444.

Nguyen, T.-H., Nguyen, T.-N. and Ngo, B.-V. (2022). A vgg-19 model with transfer learning and image segmentation for classification of tomato leaf disease, *AgriEngineering* **4**(4): 871–887.
URL: *https://www.mdpi.com/2624-7402/4/4/56*

opengenus (n.d.). Understanding the VGG19 Architecture — iq.opengenus.org, `https://iq.opengenus.org/vgg19-architecture/`. [Accessed 11-08-2024].

Priya, N., Srinidhi, K. and Kousalya, T. (2023). Carbon footprint monitoring system using machine learning and deep learning techniques, *2023 12th International Conference on Advanced Computing (ICoAC)*, pp. 1–8.

Rithik P, J., Devi R S, S., Brendan A, M. and R, S. (2023). Plant disease classification using deep learning approach (vgg19), *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1, pp. 1715–1718.

Ruiz, P. (2018). Understanding and visualizing DenseNets — towardsdatascience.com, `https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a`. [Accessed 08-08-2024].

Thenmozhi, K. and Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning, *Computers and Electronics in Agriculture* **164**: 104906.

Wang, F., Zhang, W., Lai, S., Hao, M. and Wang, Z. (2022). Dynamic gpu energy optimization for machine learning training workloads, *IEEE Transactions on Parallel and Distributed Systems* **33**(11): 2943–2954.

Yadav, A. P. S., Thapliyal, N., Aeri, M., Kukreja, V. and Sharma, R. (2024). Advanced deep learning approaches: Utilizing vgg16, vgg19, and resnet architectures for enhanced grapevine disease detection, *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1–4.

yahoo.com (n.d.). AI Carbon Footprint — in.images.search.yahoo.com, `https://in.images.search.yahoo.com/images/view;_ylt=AwrtgxVikrhmTEA7iTC9HAx.;_ylu=c2VjA3NyBHNsawNpbWcEb2lkA2MwYTE2Y2M5NTg4MWM0ZTc3ZmU4MjVlZjY1ZjcwMWZjBGdwb3MDNARpdAN`
`back=https%3A%2F%2Fin.images.search.yahoo.com%2Fsearch%2Fimages%3Fp%3DAI%2Bcarbon%2Bfootprint%26type%3DE210IN714G0%26fr%3Dmcafee%26fr2%3Dpiv-web%26tab%3Dorganic%26ri%3D4&w=800&h=450&imgurl=static.wixstatic.com%2Fmedia%2F91c4fe_5ba992ce0fe44c6eaa4f2aaa4b96ed38%7Emv2.jpg%2Fv1%2Ffit%2Fw_800%252Ch_450%252Cal_c%252Cq_80%2Ffile.jpg&rurl=https%3A%2F%2Fwww.womeninairobotics.de%2Fpost%2Fai-carbon-footprint&size=40.3KB&p=`

AI+carbon+footprint&oid=c0a16cc95881c4e77fe825ef65f701fc&fr2=piv-web&fr=mcafee&tt=AI+Carbon+Footprint&b=0&ni=21&no=4&ts=&tab=organic&sigr=V4umzP2SCATy&sigb=wFY4ONabNBLX&sigi=gehItWzXrtmW&sigt=QlPt0ohCrVpC&.crumb=3STUpDwfDYy&fr=mcafee&fr2=piv-web&type=E210IN714G0. [Accessed 11-08-2024].