HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 3 (2024)

Sustainable Farming: Automating Apple Disease Detection with Max-Voting Ensemble Deep Learning

Rajarshi Mandal[1] and Mirna Kheir Gouda[#]

[1]Lexington High School, USA
[#]Advisor

ABSTRACT

The Food and Agriculture Organization of the United Nations (FAO) estimates that 20 to 40 percent of crops produced around the world are lost to pests, costing $220 billion dollars annually. With a high nutritional and medicinal value, apples are one of the key fruits for healthy living today. Most industrialized apple orchards currently rely on human vision for disease detection, which causes a significant lag in the tracking of apple diseases resulting in poor crop yields and fruit quality. Automating apple disease detection using machine learning will lead to sustainable farming. Deep learning, a branch of machine learning, is well-suited for learning from image data. In this paper, Convolutional Neural Networks (CNN), a type of deep learning model, are investigated to accurately classify healthy and disease-affected apple leaf images. A baseline CNN model was developed along with other CNN models, namely DenseNet121, EfficientNetB7, MobileNetV2, ResNet50, and VGG16. A max-voting ensemble that included the most accurate CNN models was then deployed on the web. Consequently, apple farmers and other users can detect the three common apple leaf diseases – apple scab, black rot, and cedar apple rust, as well as healthy leaves by uploading their own images from the orchard.

Introduction

With an ever-growing gap between food demand and supply, it is imperative that we put all our efforts into maximizing crop yields[1]. Currently, the Food and Agriculture Organization of the United Nations (FAO) estimates that 20 to 40 percent of crops produced worldwide are lost to pests, costing $220 billion annually[2]. Consequently, disease detection is crucial for sustainable agriculture.

An apple contains 95 calories, 25 grams of carbohydrates, 19 grams of naturally occurring sugar, 4.4 grams of dietary fiber, and no fat. In addition, apples are a reliable source of vitamin C and phytochemicals such as quercetin, catechin, chlorogenic acid, and anthocyanin[3]. Pectin, a soluble fiber found in apples, prevents constipation and reduces bad cholesterol (LDL). With a high nutritional and medicinal value, apples are one of the key fruits today, and our dependence on apples will only increase over time. Furthermore, apples can be used to make a diverse array of products such as apple cider, apple crisps, apple juice, apple pies, applesauce, apple strudels, apple tarts, candy apples, caramel apples, and so much more! Unfortunately, apple diseases cause large crop losses. Pesticides are sprayed in excess amounts across the entire orchard as a preventative measure. This has catastrophic impacts on the environment and may render some apples inedible[4]. Thus, the timely and accurate detection of apple diseases is essential for the sustainable development of the apple industry.

The major diseases that affect the apple plants are apple scab, black rot, and cedar apple rust. Details of these diseases are below.

Apple Scab

Apple scab is a disease caused by the ascomycete fungus *Venturia inaequalis*[5]. The first symptoms of the disease are dark, irregularly shaped lesions on foliage and blossoms. As apple scab causes fruit deformation along with early leaf and fruit drop, farmers may experience crop losses of up to 70%. Chemical control, biological control, and resistance breeding programs have all been used to prevent apple scab. Organic production systems use copper or sulfur-based protectant sprays, but they do not treat pre-existing infections. In addition, the trees that are sprayed may be damaged. Serenade® ASO, a microbial biofungicide whose active ingredient is Bacillus subtilis, reduces disease incidence by 41 to 94%[6]. However, it is imperative that it is used shortly after the apple plant is infected. Finally, resistance breeding treatments have been used. Researchers have isolated the resistance gene Vf (Rvi6) and developed many cultivars for commercial sale. Unfortunately, *Venturia inaequalis* breaks down resistance genes, making this option's effectiveness questionable. In summary, treatment may be possible given an early diagnosis of apple scab.

Black Rot

Black rot is a disease caused by the plant pathogen *Botryosphaeria obtusa*. Symptoms include circular, dark brown lesions with purple margins on the leaves[7]. As black rot causes large brown rotten areas to form on the blossom end of an apple, farmers have reported crop losses of over 50% on apple trees. Physical control, biological control, and resistance breeding programs have all been used to prevent black rot. One of the most common types of physical control is pruning and burning infected areas[8]. This method requires farmers to manually find parts of their trees that are infected. Fungicides can also be used to prevent infection. Captan, a phthalimide fungicide, is a 3a,4,7,7a-tetrahydrophthalimide. It can reduce disease incidence in black rot and apple scab. However, it does not treat pre-existing infections. In addition, some fungicides are problematic for the environment and expensive. Finally, resistance breeding treatments have been used. There are many cultivars with a variety of susceptibility levels to black rot, but there is no cultivar that is completely immune to this disease. In summary, treatment may be possible given an early diagnosis of black rot.

Cedar Apple Rust

Cedar apple rust is a disease caused by the plant pathogen *Gymnosporangium juniperi-virginianae*. Symptoms of the disease include circular yellow spots on the upper part of the leaf with brownish tube-like structures sticking out on the lower part of the leaf[9]. As cedar apple rust causes infections at the blossom end of the fruit along with premature leaf drop, farmers may experience significant crop losses. Physical control, biological control, and resistance breeding programs have all been used to prevent cedar apple rust. One option is to prune infected areas to prevent the disease's spread to other trees. Farmers usually do this when the infection has reached a state where it is impossible to cure the plant of cedar apple rust. Myclobutanil, a conazole (triazole) fungicide, can treat cedar apple rust at its initial stages. Currently, farmers manually locate where to apply this fungicide. Alternatively, they might spray all of their trees with fungicides. This reduces economic profit and is problematic for the environment. Finally, resistance breeding treatments have been used. Some cultivars such as William's Pride and Liberty are highly resistant to cedar apple rust[10]. Currently, they are rather expensive, so most farmers cannot profit from these cultivars. In summary, treatment may be possible given an early diagnosis of cedar apple rust.

Figure 1. Images showing apple leaves that are infected with apple scab, black rot, cedar apple rust, or are healthy.

Most industrialized orchards currently rely on human vision for disease detection, resulting in a huge need for disease experts. Identification of diseases is a difficult undertaking as judgments may be influenced by subjective perception and visual fatigue[11]. Moreover, there exists a large gap between the demand and supply of disease experts. Additionally, continuously scrutinizing all plants for diseases in a large area is inefficient. These obstacles cause a significant lag in the tracking of apple diseases, which results in the overuse of pesticides and lower fruit quality. Therefore, the automatic identification of apple diseases is necessary to increase crop yield and quality.

Machine learning is a subfield of artificial intelligence that is devoted to developing and comprehending computer algorithms that leverage data to minimize a cost function with gradient descent[12]. Deep learning is a branch of machine learning where practitioners build models comprised of many processing layers to understand data with multiple levels of abstraction[13]. A Convolutional Neural Network (CNN), a type of deep learning model, could accurately classify images of apple leaves whether they are healthy or have diseases when it is trained with an appropriate dataset. For example, when an apple leaf image is provided as input, the system can automatically detect a healthy leaf (control group) or a leaf with a specific disease, e.g., Apple Scab, Black Rot, and Cedar Apple Rust.

In this paper, our goal was to develop an accurate deep learning model that implemented Convolutional Neural Networks (CNN) to automatically detect specific apple leaf diseases, if present, when an apple leaf image was provided as input. This model, when deployed on the web, would help apple farmers to efficiently detect apple diseases quickly for remedial actions. To achieve the goal, first we investigated prior work in this area.

Literature Review

Many attempts have been made to detect apple diseases, and they each have their advantages and disadvantages. Hyperspectral remote sensing, support vector machines, convolutional neural networks, and long short-term memory network are just a few of the many ways researchers are trying to combat apple diseases.

Hyperspectral remote sensing, also known as imaging spectroscopy, is a technique where images are created by collecting data on a wide spectrum of light. This starkly contrasts with the usual RGB color scheme, where data is collected only on red, green, and blue wavelengths of light. This technology has been previously used in the fields of astronomy, biotechnology, and environmental science. Recently, researchers have used this technology to identify Fire Blight in apple plants. Spectral signatures from 400 to 2500 nm were used to evaluate indices such as ARI (Anthocyanin Reflectance Index), RDVI (Renormalized Difference Vegetation Index), MSR (Modified Simple Ratio), and NRI (Nitrogen Reflectance Index). In addition, the researchers invented their own index, called the QFI. They found that a SWIR band at 1900 nm enabled them to distinguish between healthy, infected, and dry leaves of apple trees[14].

A Support Vector Machine (SVM) is a supervised learning technique that performs nonlinear classification using the kernel trick. This mathematical operation projects inputs into high-dimensional spaces, allowing hyperplanes to separate classes. This technology has been previously used in the fields of bioinformatics, computational biology, and geoscience. Recently, researchers have used this technology to classify apple plants as being infected with black rot, cedar apple rust, or being healthy. The multiclass SVM model had better accuracy than the hyperspectral remote sensing techniques that had been used[15].

A Convolutional Neural Network (CNN) is an artificial neural network that utilizes convolutions instead of matrix multiplications in at least a few layers. As a result of this mathematical operation, CNNs can accurately classify images. This technology has been previously used for anomaly detection, drug discovery, and image recognition. Recently, researchers have used this technology to classify apple plants as infected with apple scab, black rot, cedar apple rust, or healthy. They modeled their CNN architecture off GoogLeNet, a CNN with 22 layers[16].

A Long Short-Term Memory (LSTM) is an artificial neural network with feedback connections. This gives the model the ability to process data sequences, such as a video. This technology has been previously used for image generation, language translation, and speech recognition. Recently, researchers have used this technology to classify apple diseases and pests from images. The LSTM model consists of AlexNet, GoogLeNet, and DenseNet201. The extracted deep features are then input into three LSTM layers. The outputs of these layers are then used in a majority voting classifier. This model outperforms deep CNNs, giving it state-of-the-art accuracy[17].

Table 1. A table depicting the technique used, dataset size, and results for three different publications.

Publication (Researchers)	**Technique Used**	**Dataset Size**	**Results**
Chakraborty, S., Paul, S., Zaman, R. (2021)	Support Vector Machine	500 images	96.0%
Baranwal, S., Khandelwal, S., Arora, A. (2019)	Convolutional Neural Network	2561 images	98.4%
Turkoglu, M., Hanbay, D., Sengur, A. (2019)	Long Short-Term Memory Network	1192 images	99.2%

Transfer learning models based on CNNs (e.g., DenseNet, EfficientNet, MobileNet, ResNet, and VGG) have become immensely popular[18-22]. In this project, we focused on finding the model that could yield better accuracies than those from prior research. Moreover, we looked for models that can maintain those higher accuracies when dataset sizes were much larger than the ones used in the existing research.

Dataset

The original dataset used for this project was published by the Kaggle user "Pushpa Lama" on June 27, 2021 at 06:56:29 (Eastern Daylight Time) and was titled "Apple_disease_sp." It contained four classes (apple scab, black rot, cedar apple rust, and healthy) with about 2,000 training images and 500 testing images each. The images were stored as Joint Photographic Experts Group (JPG) files with a size of 256 pixels by 256 pixels. Consequently, the aspect ratio was 1:1. In total, there were 7,771 training images and 1,943 testing images.

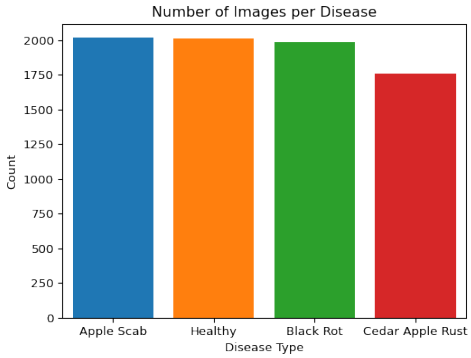

Figure 2. A bar graph depicting the class distribution for the dataset.

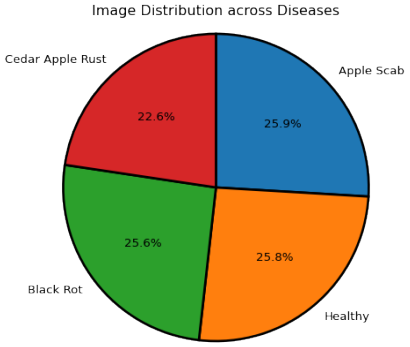

Figure 3. A pie chart depicting the class distribution for the dataset.

Next, we performed data preprocessing. Resizing the images to 224 pixels by 224 pixels allowed us to utilize a variety of transfer learning models. We also augmented our data to make our models more accurate on low-quality images and those with unusual backgrounds. Some of the augmentations used include zooming, rotating, illuminating, dimming, and shearing. The final dataset used to train our model contained 1,464 original images and 1,536 augmented images for a total of 3,000 images per class. Augmented images were not included in the testing data to understand the model's performance on high-quality images. This dataset has also been uploaded to Kaggle for the community to access.

Methods

In the proposed framework, apple leaf images were used as input. They were normalized, resized, and augmented during data preprocessing procedures. Afterward, a model classified the plant as having apple scab, black rot, cedar apple rust, or being healthy. The proposed work involved eight major steps:

Data Collection

To find a suitable dataset, we navigated to kaggle.com/datasets to look for a dataset with at least three different classes, a healthy class, and at least 500 images per class. Once the dataset was found, analysis was done on the images in the dataset. It appeared that leaves were taken off apple trees and photographed with a plain background. Afterward, a Kaggle notebook was created, and the dataset was added to the input directory.

Exploratory Data Analysis

To get a better feel for the data, we read all the image files and store them in a dataframe. Next, we plotted a bar graph and pie chart showing the image distribution across classes. In addition, eight images for each class were plotted on a grid. After that, we divided all images' pixel values by 255 so that the model trained faster.

Data Augmentation

We decided to perform data augmentation to ensure that the final model could handle images with natural backgrounds. The augmentations below were done randomly on the entire dataset:

- Zoom in or out of the image.
- Flip the image horizontally or vertically.
- Rotate the image.
- Increase or decrease the brightness of the image.
- Shear the image.

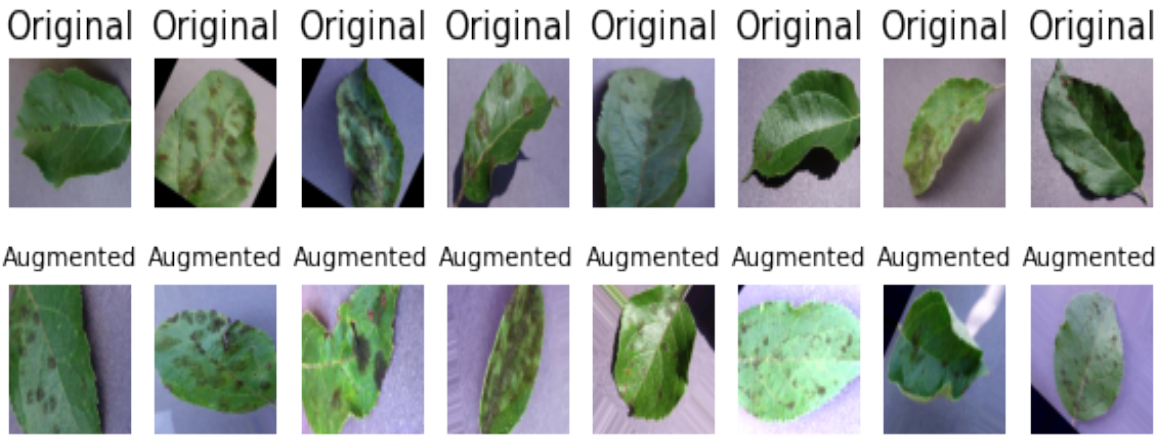

Figure 4. Images depicting the augmentation of apple leaves infected with apple scab.

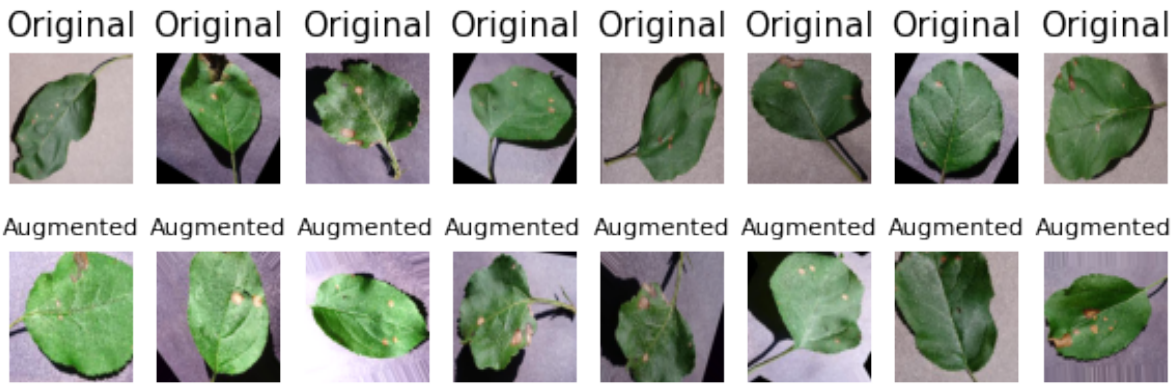

Figure 5. Images depicting the augmentation of apple leaves infected with black rot.

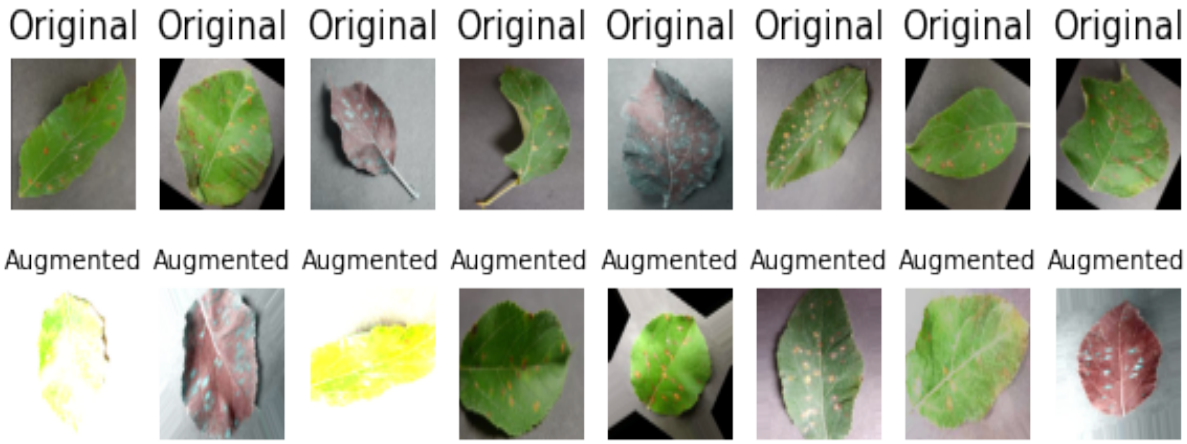

Figure 6. Images depicting the augmentation of apple leaves infected with cedar apple rust.

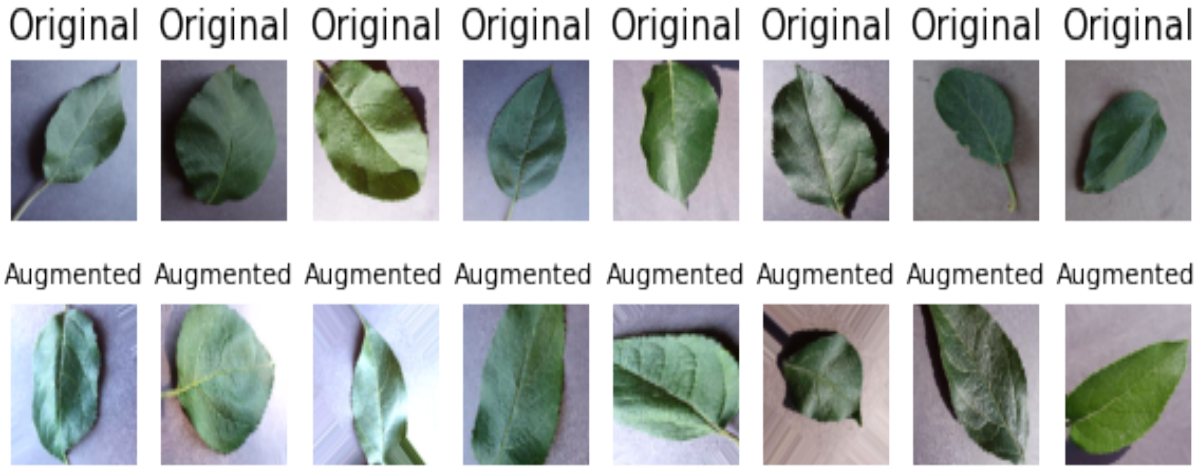

Figure 7. Images depicting the augmentation of healthy apple leaves.

Finally, we combined 1,464 images from the original dataset with 1,536 augmented images for each class. This led to each class having 3,000 images, so there are 12,000 images in total. This newly created dataset was published on Kaggle.com for others to use.

Baseline Model

First, we created the model architecture described below and built it as a "sequential model" in TensorFlow. Next, two fully connected Dense layers with the relu activation function and an output layer with four output neurons were added at the top of the model. At this stage, we checked for overfitting and underfitting using train and validation loss. Additionally, the model's performance was evaluated using validation accuracy, a confusion matrix, and additional metrics. The images that the model predicted incorrectly were visualized to better understand the model's pitfalls.

```
Input(shape=(224,224,3))
Conv2D(32, 6, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Conv2D(32, 5, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Conv2D(32, 4, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Conv2D(32, 3, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Conv2D(32, 3, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Conv2D(32, 3, padding='same', activation='relu')
BatchNormalization()
Conv2D(32, 3, padding='same', activation='relu')
BatchNormalization()
MaxPooling2D()
Dropout(0.2)
Flatten()
Dense(512, activation='relu')
Dense(512, activation='relu')
Dense(4)
```

Figure 8. A description of the baseline CNN's architecture.

Transfer Learning

The five models tested using transfer learning are DenseNet121, EfficientNetB7, MobileNetV2, ResNet50, and VGG16.

DenseNet is a CNN with each layer connected to all other layers in a feed-forward fashion[18]. This architecture not only mitigates the vanishing gradient problem – where gradients become too small to effectively train the network during backpropagation – but also enhances feature propagation and reduces the overall number of parameters required. Among the DenseNet architectures, DenseNet121 was chosen due to its balance of depth and complexity. With 121 layers, it offered a robust feature extraction capability while still being relatively less complex and computationally demanding compared to its larger counterparts. Its dense connectivity pattern ensured effective feature propagation, which was crucial for accurate disease detection in grapes, without the computational burden of the more extensive models.

EfficientNet is a CNN that uniformly scales all dimensions (depth, width, resolution) using a compound coefficient[19]. Its main benefits include being able to train efficiently, easily adapt to a wide variety of datasets, and massively reduce the number of parameters in a model. Of the eight available model architectures, EfficientNetB7 was selected due to its enhanced capability to scale depth, width, and resolution uniformly. Despite being the largest model in its series with 813 layers, the design of EfficientNetB7 optimized performance without an excessive increase in the number of parameters. This made it adept at handling a wide variety of datasets, including complex images of grapevine diseases, thereby providing a highly effective tool for disease detection that remained efficient in terms of computational load.

MobileNet is a CNN based on an inverted residual structure; the residual block's input and output are thin bottleneck layers[20]. Its main benefits include having a small model size (only about 9MB), requiring less computation due to Depthwise Separable Convolutions, and being compatible with mobile devices since it does not require Graphics Processing Units (GPUs). Of the three available model architectures, MobileNetV2 was chosen for this paper because it struck a balance between efficiency and accuracy. Its 53 layers and inverted residual structure allowed for a smaller model size, making it ideal for deployment in mobile or edge devices that vineyard farmers might use in the field. The reduced computational requirement, without a significant sacrifice in performance, made MobileNetV2 an accessible and practical option for real-time apple leaf disease detection, even in remote locations without access to powerful computing resources.

ResNet is a CNN with a Residual Block[21]. Its main benefits include allowing deep networks to train with high accuracy, minimizing the effect of layers that negatively impact model performance, and resolving the vanishing gradient problem. Among the ResNet variants, ResNet50 was selected for its relatively lower complexity, which translated to lower computational demand. With 50 layers, it still provided sufficient depth to capture complex features in grape leaf images, making it a suitable choice for accurately detecting diseases. Its streamlined architecture ensured that apple farmers could use this model without needing extensive computational power, making it a practical choice for various operational environments.

VGG is a CNN with few layers but many parameters[22]. Its main benefit is being relatively simple to understand and explain. The two model architectures are VGG16 and VGG19. VGG16, which is simpler than VGG19 with its 16 layers, was chosen for its ease of understanding and implementation. Although it had a high number of parameters, its architectural simplicity made it a good choice for apple farmers who needed a straightforward yet effective model for disease detection.

For each transfer learning model, we imported its pre-trained weights from ImageNet. Next, a sequential model containing the transfer learning model was created. A GlobalAvgPool2D layer was then added to the sequential model. By calculating the mean of the input's width and height, this layer performed downsampling. It reduced the total number of parameters and the chance of overfitting. Finally, we added a Dense layer with a softmax activation. By doing this, raw neural network outputs were converted into probability vectors. Next, we set an EarlyStopping callback that stops training if validation accuracy does not improve after eight epochs. In addition, we implemented a ModelCheckpoint callback that saves the model with the lowest validation loss. The sequential model was compiled with the Adam optimizer. Finally, each model was trained for 25 epochs.

Max-Voting Ensemble

Max-voting involves combining multiple models and using the majority vote of their predictions to make the final decision[23]. Max-voting ensemble improves the overall accuracy by leveraging the strengths of diverse models, making it particularly effective in complex tasks like disease detection in agriculture.The most suitable CNN models for the max-voting ensemble were determined using validation accuracies, confusion matrices, and error visualizations. The HDF5 files containing these models' weights were downloaded. In our max-voting ensemble, each base model made a prediction on an image. Each prediction counted as a vote, and the disease with the most votes was the final prediction. If there was a tie, the final prediction was decided by the highest scoring base model.

Model Evaluation

All models were evaluated using accuracy score, confusion matrix, classification metrics, and error visualization. The accuracy of a model is the number of correct classifications on the testing data divided by the total number of samples in the testing data. Confusion matrices are tables that provide information regarding model errors. This enables us to get a better understanding of a model's pitfalls. Error visualizations are a tool machine learning practitioners utilize to illustrate the weaknesses of a model. We also calculated the time it takes for the max-voting ensemble to classify a single image with and without a GPU. Finally, the model's ability to classify pictures with natural backgrounds was tested using images web scraped from Google.

Website Deployment

In order to develop a website, we started by creating a Google Colab notebook. Next, we logged into an NGrok account and copied the authentication token. It was pasted into the notebook environment to connect to NGrok. Basic code was written that loaded the model and classified an input image as having a certain disease. It was saved as a Python file and run using Streamlit. The link printed in the cell output was used to access the website.

Results and Discussion

We looked at model accuracies and confusion matrices as well as conducted error visualization to analyze the findings.

Accuracy

The three most accurate models had over 99% accuracy. The ResNet50, baseline CNN, and EfficientNetB7 models had accuracies of 99.7%, 99.4%, and 99.2% respectively. They were better than the other three models, so ResNet50, baseline CNN and EfficientNetB7 were used for the max-voting ensemble. VGG16, DenseNet121, and MobileNetV2 had accuracies of 98.0%, 96.4%, and 92.7% respectively. This information was visualized in the bar graph below.

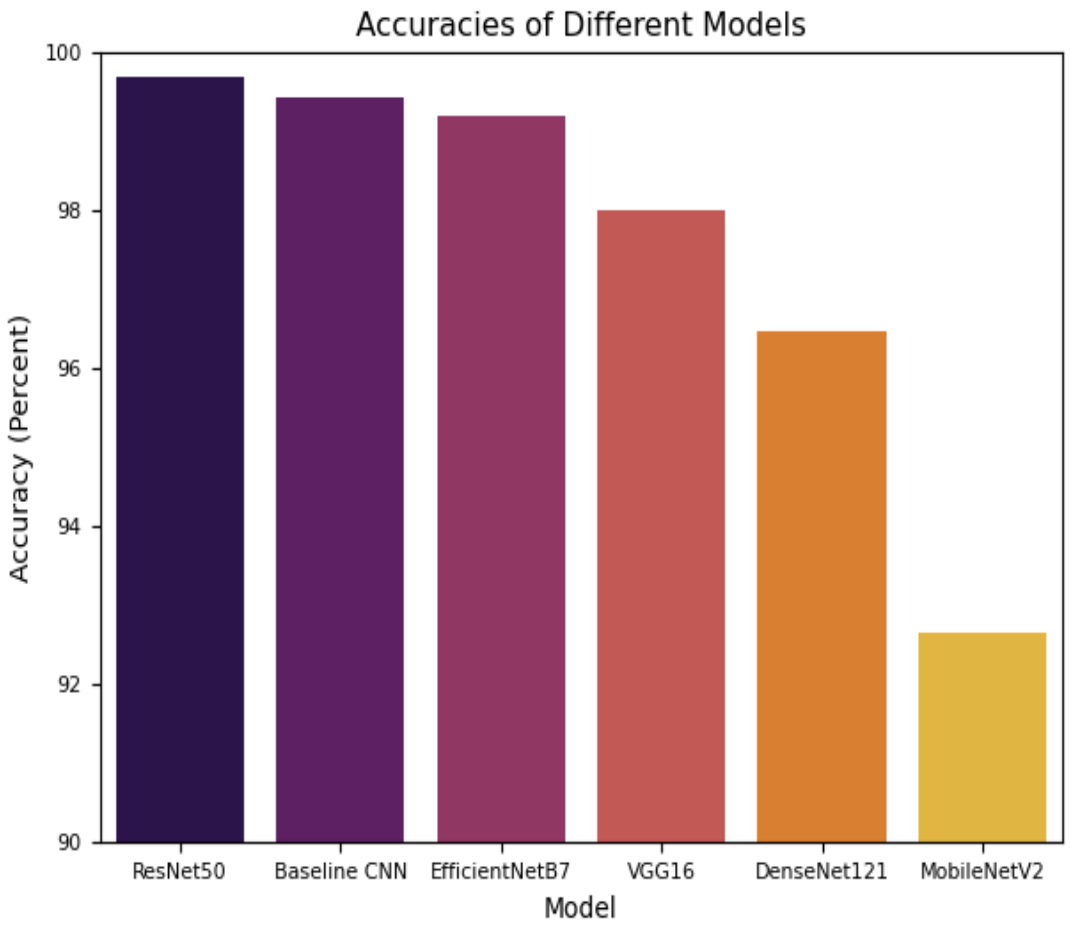

Figure 9. A bar graph depicting the accuracies on testing data of the models tested.

Confusion Matrix

Confusion matrices are tables that provide information regarding model errors. It is especially helpful when there is a drastic class imbalance. However, it always gives insights into a model's strengths and weaknesses. The confusion matrix for each model is shown below.

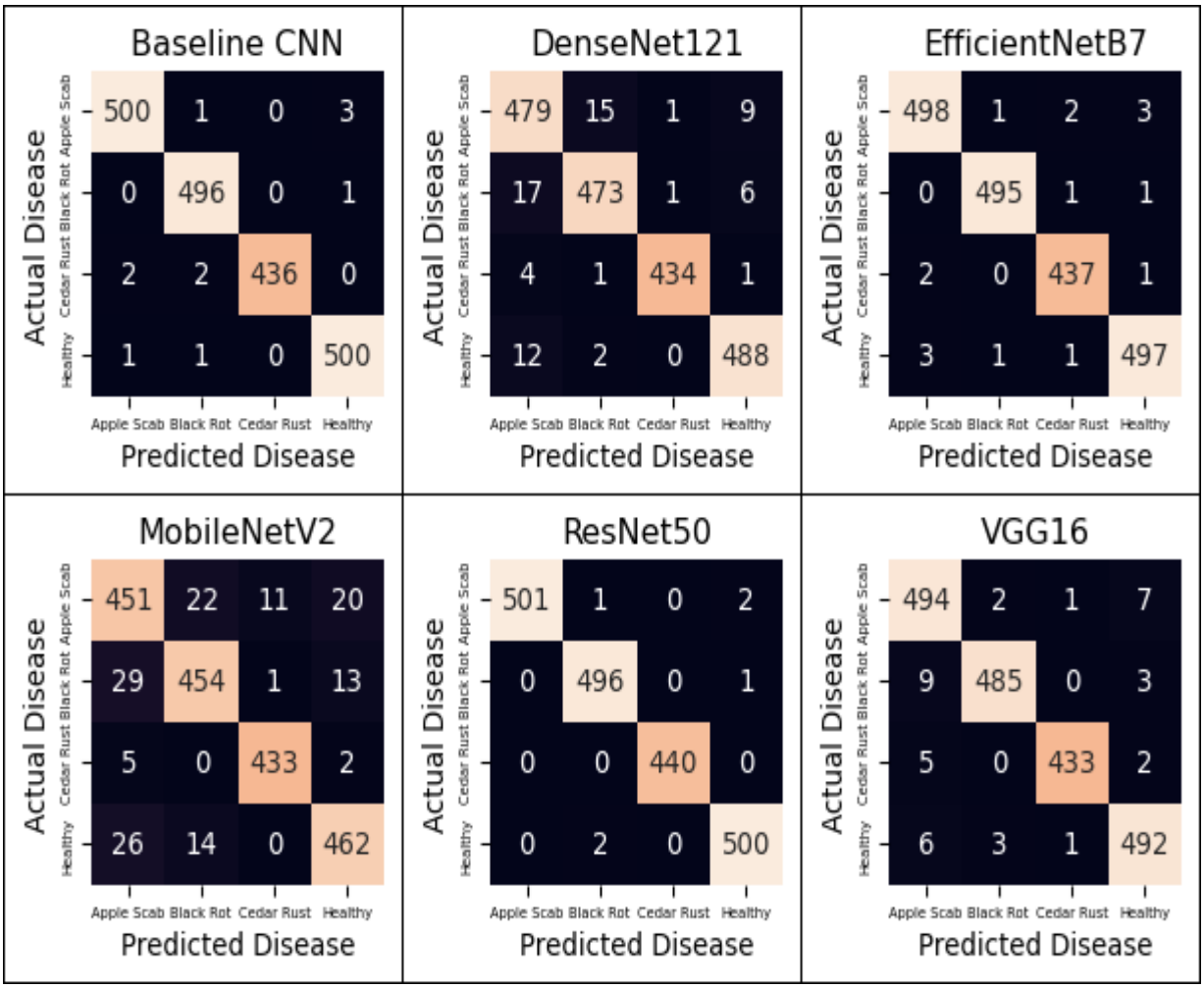

Figure 10. Six confusion matrices depicting errors of the models tested.

From these matrices, it was visually clear that there were a few mistakes in all of the classes. The models did not struggle with any particular disease. It appeared that cedar apple rust was the easiest to classify. In particular, DenseNet121, and MobileNetV2 made lots of errors, but few with cedar apple rust. ResNet50, the best-performing model, made no errors with cedar apple rust.

Error Visualization

Error visualizations are used to illustrate the weaknesses of a model. Below is the error visualization of the ResNet50 model.

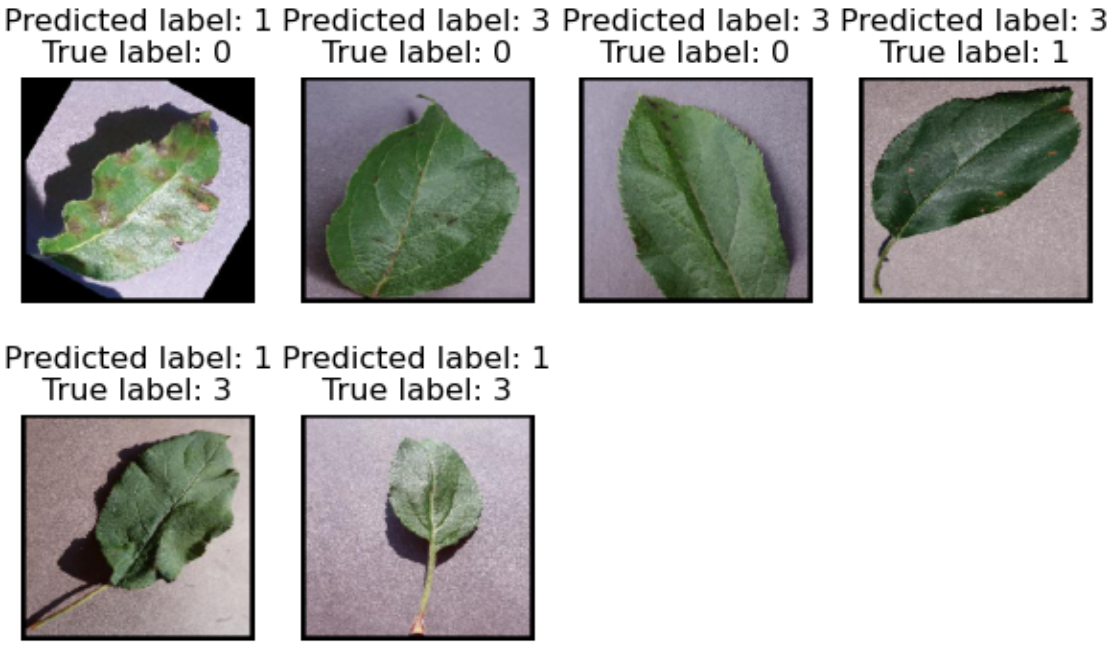

Figure 11. The error visualization of the ResNet50 model. The numbers 0, 1, 2, and 3, refer to apple scab, black rot, cedar apple rust, and healthy respectively.

These images were nearly impossible to classify into four classes. They look almost identical. To the naked eye, it appears that all but one of the images are healthy, but in reality, only two of them are healthy. It is no surprise that transfer learning models cannot classify them accurately; even a trained visual expert might be unable to categorize these images accurately.

Learning Curve

Next, we gathered insight from learning curves. The train and validation accuracy were plotted for each model.

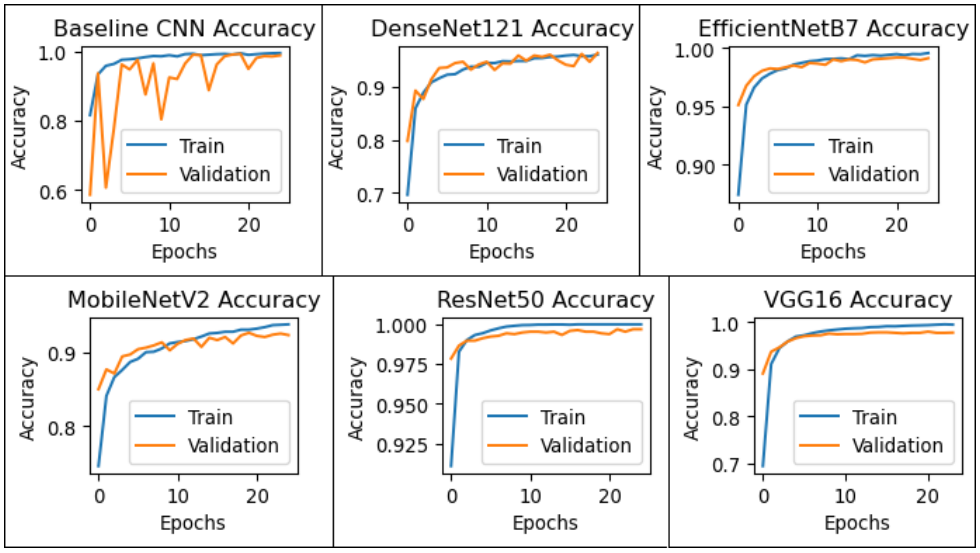

Figure 12. Six line graphs depicting the training and validation accuracies of all models tested.

First, we noticed that the training accuracy is higher than the validation accuracy. This made sense intuitively. However, we noticed that the baseline CNN's validation accuracy during training was inconsistent. This hinted that the model might be overfitting to the training data. The rest of the line graphs had no issues.

Model Size and Accuracy

An observation was also made regarding the size of file containing the model's weights and the accuracy of the model.

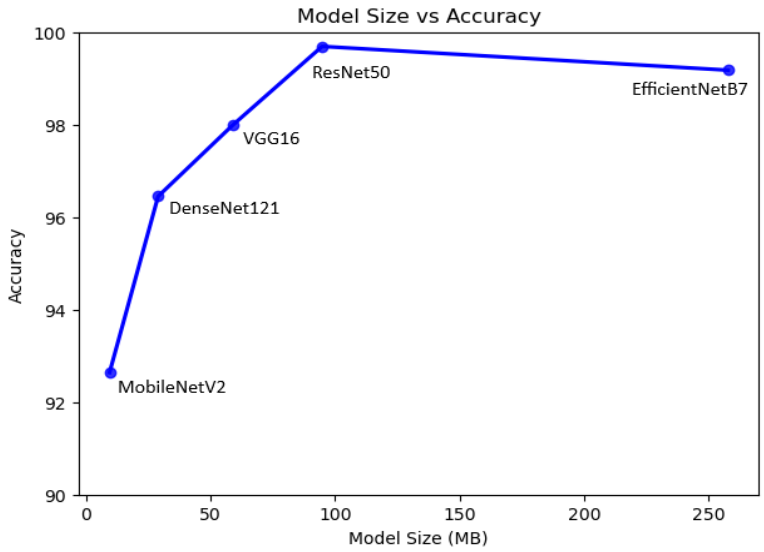

Figure 13. A line graph depicting the correlation between the size of the model's weights in megabytes and its accuracy on testing data.

After carefully inspecting the graph, we found that there is a strong correlation between the model weight's file size and accuracy. After about 150 megabytes, increases in model size will only yield a tiny improvement or may even cause some harm to the accuracy score.

Max-Voting Ensemble

The three highest performing models (ResNet50, baseline CNN, and EfficientNetB7) were used in a max-voting ensemble. Its classification metrics, confusion matrix, and error visualization are below.

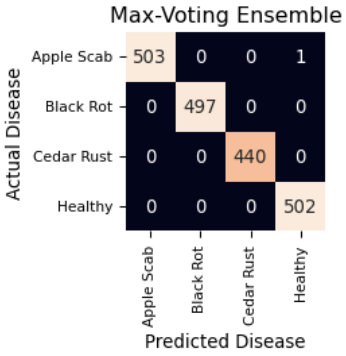

Figure 14. A confusion matrix depicting the errors made by the max-voting ensemble.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	504
1	1.00	1.00	1.00	497
2	1.00	1.00	1.00	440
3	1.00	1.00	1.00	502
accuracy			1.00	1943
macro avg	1.00	1.00	1.00	1943
weighted avg	1.00	1.00	1.00	1943

Figure 15. A table depicting classification metrics for the max-voting ensemble.

Figure 16. The error visualization for the max-voting ensemble. The numbers 0, 1, 2, and 3, refer to apple scab, black rot, cedar apple rust, and healthy respectively.

The max-voting ensemble performed better on the testing data than all other models; it had an accuracy of 99.9%. The ensemble only had one misclassification on the testing data. To determine its efficiency, the ensemble was timed while giving predictions from pictures. It took the max-voting ensemble an average of 12.59 milliseconds with a GPU and 523 milliseconds without one to classify a single image.

Conclusion

In this paper, we have learned that ResNet50 and EfficientNetB7 work well to detect diseases in apple plants. By studying the results from different CNN model architectures, it appears that a Residual Block or its inversion

help models predict apple diseases. Essentially, letting information flow directly from early to later layers allows a model to keep the information found in early layers. A simple baseline CNN model architecture can yield high accuracy but is inconsistent across just a few epochs. Additionally, we learned that ensembling methods such as max-voting enable us to achieve state-of-the-art accuracy on testing data.

An interesting positive correlation was found between the file size of a model's weights and the model's accuracy. This hints that deep models with many parameters are necessary to accurately detect plant diseases in their initial stages. Furthermore, we found that data augmentation is crucial, as it makes the final model more robust. In fact, the final model could handle images with natural backgrounds, even though all of the training data consisted only of plain backgrounds.

Future Work

Currently, it takes the max-voting ensemble 12.59 milliseconds to predict a single image with a GPU. However, most farmers lack access to GPUs. Without one, it takes the max-voting ensemble 523 milliseconds to predict a single image. For this technology to be applied effectively in rural areas with poor farmers, this time must be reduced. Furthermore, the model requires 224 by 224 pixel images or larger ones for accurate predictions. Having another model that handles extremely low-resolution images may be helpful in rural areas with low-resolution photos.

Next, we can experiment with more model architectures such as AlexNet, GoogLeNet, Inception V3, and Xception in the hopes of finding a better model architecture. In addition, hyperparameter tuning can also be conducted to further increase model performance by finding the most effective optimizer, learning rate, and activation function. This would be done to avoid an ensemble, as this would significantly reduce the prediction time per image without a GPU.

Finally, the users can only input one image at a time on the website. In the future, it is imperative that the website takes thousands of images as input simultaneously for it to be operating at the industrial level. Added functionality would solve this problem. Additionally, a mobile app would be immensely helpful because it could run locally on a farmer's smartphone.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

1. Ranganathan, J., Waite, R., Searchinger, T., & Hanson, C. (2018, December 5). How to sustainably feed 10 billion people by 2050, in 21 charts. World Resources Institute. Retrieved May 31, 2024, from https://www.wri.org/insights/how-sustainably-feed-10-billion-people-2050-21-charts

2. Food and Agriculture Organization (FAO). (2019, April 3). New standards to curb the global spread of plant pests and diseases. Retrieved May 31, 2024, from https://www.fao.org/newsroom/detail/New-standards-to-curb-the-global-spread-of-plant-pests-and-diseases/en

3. The Nutrition Source. (2021, October 5). Apples. Retrieved May 31, 2024, from https://www.hsph.harvard.edu/nutritionsource/food-features/apples/

4. Linhart, C., Niedrist, G. H., Nagler, M., Nagrani, R., Temml, V., Bardelli, T., Wilhalm, T., Riedl, A., Zaller, J. G., Clausing, P., & Hertoge, K. (2019, May 8). Pesticide contamination and associated risk factors at public playgrounds near intensively managed Apple and Wine Orchards. *Environmental Sciences Europe*. SpringerOpen. Retrieved May 31, 2024, from https://enveurope.springeropen.com/articles/10.1186/s12302-019-0206-0

5. Penn State Extension. (2017, October 17). Apple disease - apple scab. Retrieved May 31, 2024, from https://extension.psu.edu/apple-disease-apple-scab

6. OSU Extension Service - Extension and Experiment Station Communications. (2023, April 17). Apple (Malus spp.) - scab. *Pacific Northwest Pest Management Handbooks*. Retrieved May 31, 2024, from https://pnwhandbooks.org/plantdisease/host-disease/apple-malus-spp-scab

7. Division of Plant Sciences. (2020, June 19). Black rot on apples (Michele Warmund). *Missouri Environment and Garden News Article*, Integrated Pest Management, University of Missouri. Retrieved May 31, 2024, from https://ipm.missouri.edu/MEG/2020/6/blackRotApple-MW/

8. Grabowski, M. (2019). Black rot of apple. UMN Extension. Retrieved May 31, 2024, from https://extension.umn.edu/plant-diseases/black-rot-apple

9. George, H. (2023, April 20). How to identify, prevent, and control Cedar Apple rust. *Gardener's Path*. Retrieved May 31, 2024, from https://gardenerspath.com/how-to/disease-and-pests/cedar-apple-rust-control/

10. Oklahoma State University. (2017, February 1). Cedar-Apple Rust. Retrieved May 31, 2024, from https://extension.okstate.edu/fact-sheets/cedar-apple-rust.html

11. Dutot, M., Nelson, L. M., & Tyson, R. C. (2013, May 24). Predicting the spread of postharvest disease in stored fruit, with application to apples. *Postharvest Biology and Technology*. Retrieved May 31, 2024, from https://www.sciencedirect.com/science/article/abs/pii/S0925521413001087

12. Mitchell, T. M. (1997). *Machine learning*. MacGraw-Hill.

13. LeCun, Y., Bengio, Y., & Hinton, G. (2015, May 27). Deep learning. *Nature News*. Retrieved May 31, 2024, from https://www.nature.com/articles/nature14539

14. Skoneczny, H., Kubiak, K., Spiralski, M., & Kotlarz, J. (2020). Fire blight disease detection for apple trees: Hyperspectral analysis of healthy, infected and dry leaves. *Remote Sensing, 12*(13), 2101. https://doi.org/10.3390/rs12132101

15. Chakraborty, S., Paul, S., & Rahat-Uz-Zaman, M. (2021). Prediction of apple leaf diseases using multiclass support vector machine. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. https://doi.org/10.1109/icrest51555.2021.9331132

16. Baranwal, S., Khandelwal, S., & Arora, A. (2019). Deep learning convolutional neural network for apple leaves disease detection. *Social Science Research Network*. https://doi.org/10.2139/ssrn.3351641

17. Turkoglu, M., Hanbay, D., & Sengur, A. (2019). Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *Journal of Ambient Intelligence and Humanized Computing, 13*(7), 3335–3345. https://doi.org/10.1007/s12652-019-01591-w

18. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700-4708).

19. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv*. Cornell University. https://arxiv.org/abs/1905.11946

20. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*. Cornell University. https://arxiv.org/abs/1704.04861

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

22. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*. Cornell University. https://arxiv.org/abs/1409.1556

23. Sarkar, D., & Natarajan, V. (2019). Max-Voting. In *Ensemble Machine Learning Cookbook*. O'Reilly Media. Retrieved May 31, 2024, from https://www.oreilly.com/library/view/ensemble-machine-learning/9781789136609/6571afd0-0bac-4bb6-9698-c68504baebae.xhtml