Detect Foliar Disease in Apple Trees using Deep Learning

MSc Research Project
Data Analytics

Aishwarya Said
Student ID:20135670

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aishwarya Said
Student ID:	20135670
Programme:	Msc Data Analytics
Year:	2021-2022
Module:	MSc Research Project
Supervisor:	Jorge Basilio
Submission Due Date:	16/12/2021
Project Title:	Detect Foliar Disease in Apple Trees using Deep Learning
Word Count:	4176
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:	30th January 2022

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):	

Detect Foliar Disease in Apple Trees using Deep Learning

Aishwarya Said
x20135670

Abstract

Apples are one of the most important fruits in the world. We all have heard the famous saying 'an apple a day keeps the doctor away' which means that eating an apple a day helps us to maintain good health. Leaf disease is one of the major threats to the overall quality of an apple orchard. The traditional method of identifying the disease requires manual scouting of apple orchards by humans, which is time-consuming and expensive. This research focuses on exploring different deep learning approaches to identify leaf disease. In this study, we have used the standard convolution-neural network and transfer-learning based model to identify and diagnose leaf diseases in apple trees. Along with this, we would also explore different data augmentation techniques and analyse their effects on the overall accuracy of the model. All the implemented models are trained on the Plant Pathology 2021 dataset which consists of approximately 23,000 high-quality RGB images of apple leaves with 12 different foliar diseases. Out of 23,000, 18632 images are unhidden and are available to the Kaggle community to train machine learning models. However, due to computational limitations, we have used only 40% of the data such that the proportion of each disease remains the same.

1 Introduction

An apple a day keeps the doctor away. In order to keep doctors away, every human should eat apples. For every human to eat apples, apple trees need to be in good shape and yield a large quantity of apples. Apple tree first originated in central Asia. Apples are grown majorly in Asia and Europe. European colonists took apples to America and that's how America started growing apples. In 2018, 86 million tonnes of apples were grown.Wikipedia (2021).In a couple of years, apple production has been highly affected primarily due to socio-economic factors, post harvest strategy, delay in identifying plant diseases. Apple trees are susceptible majorly to fungus and bacterial disease. Three majorly named diseases are Mildew, Aphids and Apple Scab. In order to make a tree survive and have good quality apples, finding the disease is paramount.Traditionally farmers used to do manual scouting which is time consuming and expensive. Use of technology will help to identify disease much quicker. Early identification will help farmers to take precautionary measures which will halt the disease spreading into the roots. Automation has definitely helped farmers in early detection with minimal cost. Anuradha Badage(Badage; 2018) proposed a system to inform early detection of disease in the outer layer of leaves using edge detection and histogram matching.(Canny; 1986) uses edge detection to find the disease in plants. In another research,(Korkut et al.; 2018) researchers used images of plant species. Extracted the important feature, trained various machine learning models and came up with the accuracy of 94%.

Despite the advent of Machine learning algorithms, there are few limitations which need to be addressed. In this research paper, advanced deep learning algorithms are used to classify the foliar disease in apple trees. Traditionally machine learning algorithms have been used and they have provided the results but it also imposed certain limitations. Major advantage of deep learning is 'Automatic Feature engineering'. Machine itself picks the features and trains the model and provides better accuracy than the feature selection method in machine learning. Another technique of deep learning is 'Data Augmentation'. The use of data augmentation helps to build a model which is robust. Data augmentation creates a new training set of data from the existing data and trains the model. It provides data to the model which the model is likely to see in the future. It helps to diversify the data and hence the model performance better thereby giving a good accuracy. It is evident, in order to train models we need large and quality data. Providing more data to the model, model performance more efficiently and gives a good accuracy. CNN is considered to be the most powerful tool in image classification. CNN works like a human brain, input is fed to a network of neurons, it does the processing and output is provided. In this research, deep learning techniques mainly data augmentation, transfer learning and CNN are used to overcome the limitations of machine learning and to classify the foliar disease. Below figures shows the working of machine learning and deep learning

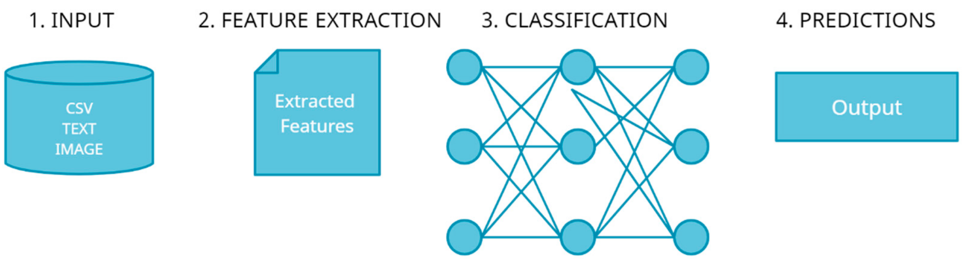

Figure 1: Working of Machine Learning

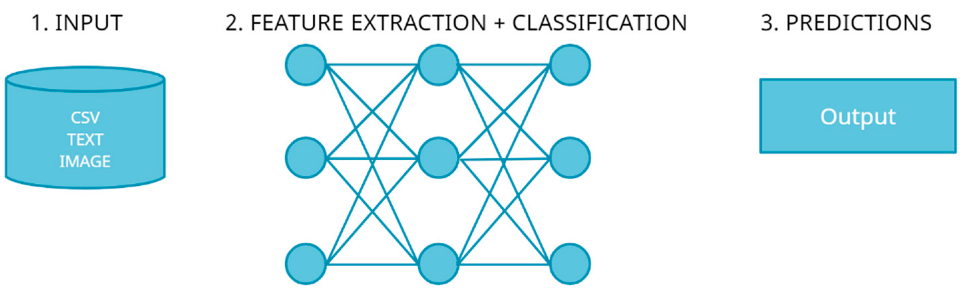

Figure 2: Working of Deep Learning

1.1 Research Question

How can we use deep learning techniques to increase the efficiency of foliar disease identification in apple trees?

1.2 Research Objective

- Implement the Standard CNN model and Transfer-Learning Model.
- Hypertune the models to achieve high accuracy.
- Analyse how data augmentation affects the accuracy of models

2 Related Work

In this section, previous work done in plant pathology and classifying diseases are highlighted. Techniques used by the researches,limitations of those techniques and how the use of deep learning algorithms can be used to overcome are discussed.

2.1 Machine Learning

Researchers have used different techniques of machine learning in the agriculture domain to classify diseases.(Korkut et al.; 2018) used image processing and machine learning methods to automatically detect plant disease using leaf dataset of different species. Researchers used different machine learning models and the proposed model gave an accuracy of 94%.(Pardede et al.; 2018) used autoencoder - an unsupervised machine learning model to classify plant disease. (Selvaraj et al.; 2019) detected disease in banana trees using pre-screen banana leaves and damaged leaves. six different models were created using 18 different classes and the accuracy range in 70 to 99% for different models. (Rumpf et al.; 2010)used spectral vegetation indices and support vector machines to differentiate disease and non disease sugar beet leaves. Proposed model achieved 97% accuracy on the validation dataset. (Sinha and Shekhawat; 2020) researched about the scope of modification by studying methodologies studied by earlier researchers.(Ramesh et al.; 2018) researched crop detection disease using the dataset created by self and extracted the feature using a histogram(HOG). (Yang and Guo; 2017)researched the importance of machine learning in plant disease and plant immunology.(Wrzesień et al.; 2019) and 3 others conducted a study in predicting apple scab.In the experiment they used sensors to understand the relation between weather and apple scab. The goal of the experiment was to replace physical sensors with virtual sensors. Use Machine learning Models to train sensors to capture the weather conditions particularly wetness of apple leaf. Use the output of this model as the input to the model to predict the apple scab. Author used 179 Machine learning models for this experiment. Random forest outperformed other classifiers with the best overall accuracy. Experiment was completely based on a hypothesis, and the virtual sensors just captured the scenario of wetness of the leaf. Model needs to be trained on the real scenarios and with all the possibilities of weather.

The limitations of these research is their inability to automate feature extraction and feature classification. Machine learning models require two additional steps that are extraction and classification. With the advent of deep learning, computer vision there are efforts to automate it and overcome these limitations.

2.2 Deep Learning

(Mohanty et al.; 2016) trained a deep learning model on a public image dataset with records more than fifty thousand of diseased and non diseased plants. The aim of this research was to identify fourteen crop species along with the different bugs. Validation accuracy on the dataset came to 99.3%. (Saleem et al.; 2019) discussed the implementation of deep learning in classifying diseases.(Goncharov et al.; 2018) introduced the Deep Siamese convolutional network for grape leaf disease. (Sun et al.; 2020)proposed a unique way to detect maize leaf blight. Approach involved a combination of preprocessing, tuning and modelling. Final model gave the highest average precision of 91.83%. Authors introduced a new technique which combined visible and infrared images from two sensors. Final model achieved more than 92% at grapevine and 87% at leaf level accuracy. (Lu et al.; 2017) introduced an approach to detect common rice diseases using CNN. Proposed CNN model achieved accuracy of 95.48%. Authors used LeNet architecture to automate the detection of banana disease under challenging conditions. (Barbedo; 2018) discussed the important factors in deep learning which need to be considered for realistic solutions. Author also implied that in detecting diseases, other parts of plants except leaves must also be considered. In the paper, the author summarised different technologies and listed their pros and cons and to what extent they can help in the detection method.(Liu et al.; 2018) developed an approach to detect four diseases of apple trees. In this method, the author reduces the model parameter compared to the standard AlexNet model and it helps to increase the accuracy by 10.83%. Overall accuracy of the model is 97.62%. This unique method enhances the robustness of the model. (Chuanlei et al.; 2017) used image and pattern recognition techniques to classify the disease. Process followed as color transformation(RGB Model) converted to HSI. Thirty eight features were extracted by removing backgrounds and only selecting the features which were essential. To get the best accuracy and to reduce the dimensionality of Genetic Algorithms(GA) and Correlation based feature selection(CFS) were used. Final model gave an accuracy of 90%. Bingze Dai, Tian Qiu et al.**?** used histogram of gradients(HOG) and trained models on 4 deep learning algorithms. Out of which GoogLeNet gave the highest accuracy of 94%. (SARDOĞAN et al.; 2020) presented Faster Region-Based Convolutional Neural Network (Faster R-CNN) with Inception v2 architecture to detect disease in apple trees. Model gave an accuracy of 84.5%. (Jiang et al.; 2019) used conventional SSD for detection it is one of the best algorithms in terms of accuracy and speed. To enhance the performance, the structure of feature maps was enhanced. Final model gave an accuracy of 78.08% with an overall speed of 23.13 frames per second.

In the previous research there are few limitations, Data Augmentation - Previous researches have limited the use of image augmentation.Image augmentation allows the model to be robust. Previous research performance was not adequate in detecting multiple diseases. Especially under challenging cases.

2.3 Image Segmentation

Image segmentation is important as it extracts only the important features of the image. It helps to concentrate on the infected part on which analysis has to be done.(Ali et al.; 2017) used Delta E segmentation, color histogram and textural features to classify disease. Their method achieved an accuracy of 99% and it also used PCA to reduce the features. Some researchers claimed ROI to be one of the best suited techniques for image segmentation.(Kao et al.; 2019) used convolutional encoder to determine ROI of the image. (Pujari et al.; 2013) used a neural network classifier for image segmentation and feature extraction. It used two phase techniques, in first it used thresholding, k-means clustering. In the second it used a k-length matrix to calculate the features. (Arsenovic et al.; 2019) and a group of researchers transformed the image into three color spaces and outperforms other research in terms of accuracy. (Hu et al.; 2017) used deep learning techniques for image segmentation. He used a two step approach to segment and classify the disease. Model proposed acquired the accuracy of 93.67%

The above techniques have shown poor performance when implemented in the real world application. Colour complexities, ROI, threshold value doesn't guarantee the image segmentation for the model which can help to correctly identify the disease

2.4 Feature Extraction

Feature extraction helps to combine two or more features or to reduce some features. In the end, it helps to have less number of features to work on with better accuracy.(Hu et al.; 2017) uses dempster shafer(DS) and multi feature fusion to recognise the disease in rape plant.(Turkoglu and Hanbay; 2019) uses LBP(Local binary pattern) which provides high performance features without the need of pre-processing. Researchers test the features using noise and then confirm the features using EML(extreme machine learning).

In the advanced deep learning models, the model automatically detects the features which are important for the model training.

3 Methodology

The findings of any research highly depend on the methodology we used to tackle the problem statement. In this study, we have used Knowledge Discovery in Database methodology. We have used this methodology to derive a lot of useful insights from the data. These derived insights will help us to understand the data better and ultimately help our deep learning models by increasing their accuracy.

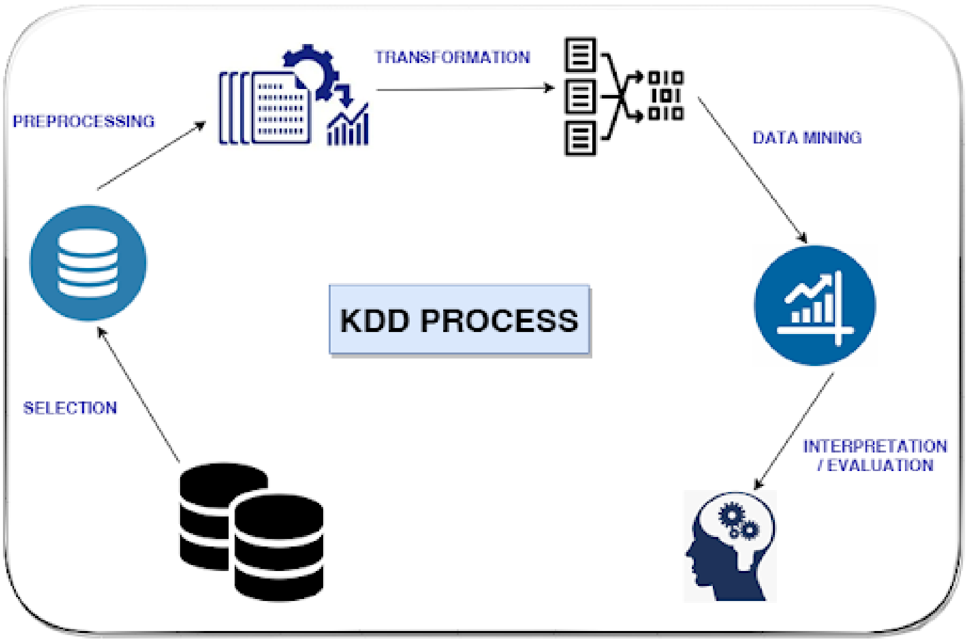

Figure 3: Methodology

4 Data Selection

The selection of the dataset depends on our research question. The right dataset is the one that will help us to answer our research question. For this research, we have selected the Plant Pathology 2021[1]dataset which is available on Kaggle. The dataset has 18632 unhidden high-quality RGB images of foliar diseases of apples. These manually captured images have real-life symptoms of multiple apple foliar diseases with variable illumination, angles, surfaces, and noise to simulate real-life scenarios. Figure 4 shows a few of the images with their labels.

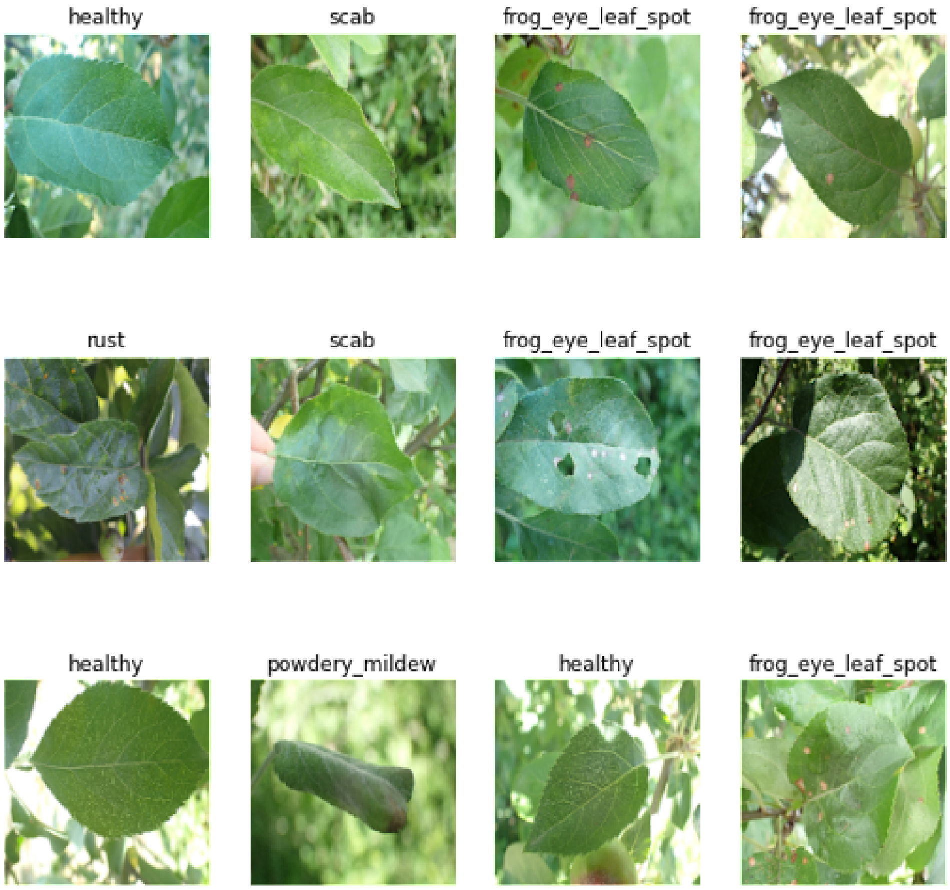

Figure 4: Few Samples of Dataset

5 Data Preprocessing

Before feeding the images to our deep learning models, all the images must be compatible with the respective model. For that, we need to rescale the size of the images such that the image size is equal to the input size of the respective model. Also, all the pixel values in our images range from 0 to 255. However, to improve the performance of our model, we will normalise the pixel value by converting their range to 0 and 1

[1]`https://www.kaggle.com/c/plant-pathology-2021-fgvc8/overview`

6 Exploratory Data Analysis

In total, we have 12 different foliar diseases. The distribution of those is shown in figure 5. From the figure, we can observe that the dataset is highly unbalanced. We can use the data augmentation technique to create more samples of images to balance our dataset.

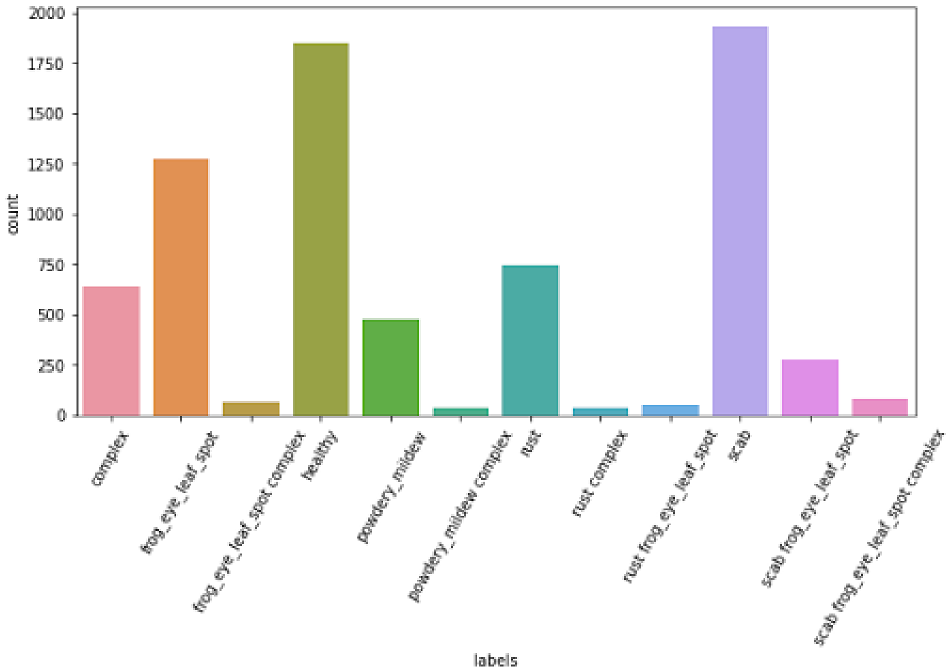

Figure 5: Count Plot of Image Labels

7 Data Augmentation

Data Augmentation is a technique that is used to increase the amount of data by slightly modifying the already existing data. Our deep learning models are as good as the data we feed to them. The variance created by data augmentation in existing data helps us to reduce overfitting. As a result, it increases the generalisation ability of the model. Data augmentation also helps to resolve the class imbalance issue in our dataset.The problem of imbalance dataset arises when data of one class is more than the other. Traditionally, upsampiling,smote and data dulpication were used to handle imbalance dataset.However, in the case of unstructered data these techniques won't be effective. Input to the model is the pixel of images, changing the pixels changes the entire information of image.This technique is called Data Augmentation. Some of the simple yet powerful data augmentation techniques which we have used in our research are flipping, rotation and contrast.Data augmentation is the artifical technique to increase the dataset.

7.1 Flipping

In flipping, we can flip the image horizontally or vertically. In vertical flipping, we exchange the order of rows whereas, in horizontal flipping, we exchange the order of columns.

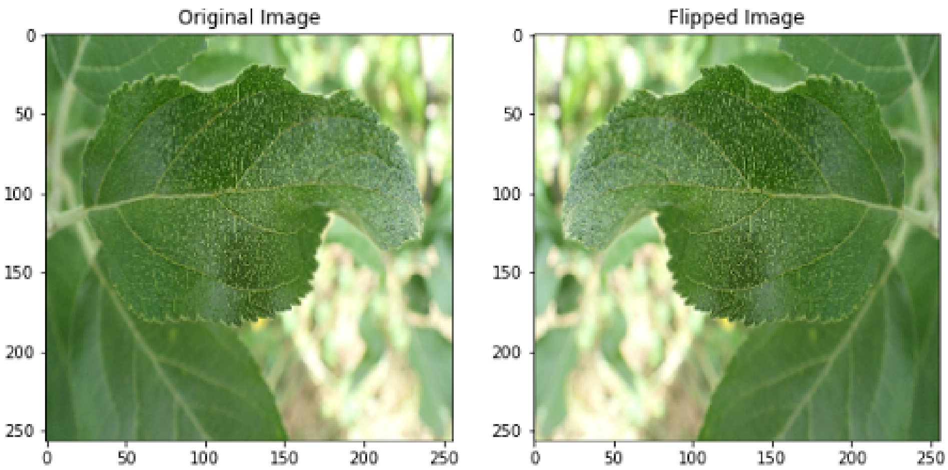

Figure 6: Flipped Image

7.2 Rotation

In rotation, we rotate the image randomly by a certain degree in the clockwise or anti-clockwise direction. Rotation randomly rotates an image from 0 to 360.

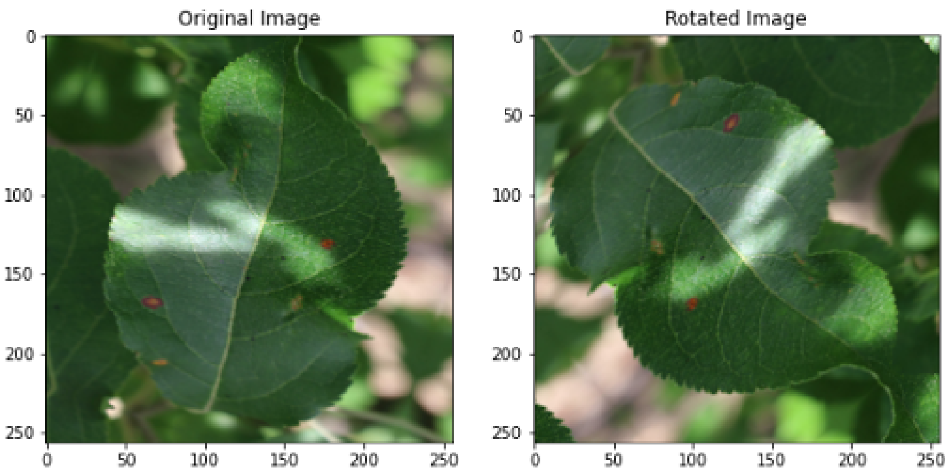

Figure 7: Rotated Image

7.3 Contrast

In this type, we change the contrast of the image by a certain degree.Contrast is the change of light and dark colour in an image. Contrast doesn't specifically means two opposite colour. There can be slight change in the tone of a colour which is also categorised as the contrast. Changing the tone of an image changes the pixel thereby giving a complete new image to the model.

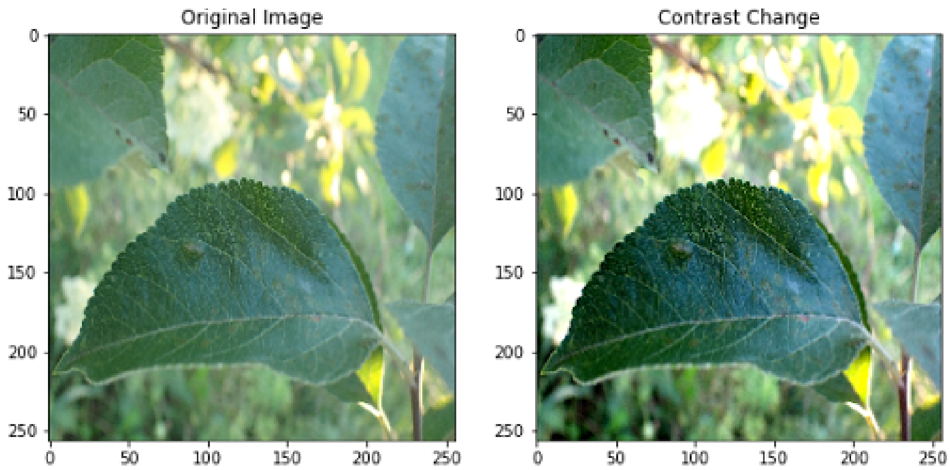

Figure 8: Contrast Image

8 Deep Learning Models

8.1 Convolutional Neural Network

Convolutional Neural Network(CNN) is a class of deep neural networks, most commonly used for image classification. A major advantage of CNN over Artificial Neural Network(ANN)(Dongare et al.; 2012), another class of deep neural networks, is its ability to reduce the number of parameters. This ability has helped researchers and developers to solve more complex tasks. One important thing we assume while solving problems using CNN is that the problem doesn't have any spatially dependent feature(Albawi et al.; 2017).

8.2 Transfer Learning

Transfer Learning is one of the deep learning approaches that focus on storing knowledge gained while solving one problem and using that gained knowledge to solve a different but related problem. The basic intuition behind image classification using transfer learning is that if a model is trained on a large and general enough dataset, the same model can be used as a base model for other image classification problems. We can take the advantage of the features learned by this model without having to start from scratch. We can fine-tune this pre-trained model as per our dataset by unfreezing a few of the top layers and jointly training the unfreeze layer of the base model and newly added classifier layers.

9 Implementation

For this study, we have used the TensorFlow library in python to create the input pipeline and implement our deep learning models. Whereas, we have used the Matplotlib library to visualise and analyse the implemented models.

9.1 Data Preparation

The TensorFlow library method 'image_dataset_from_directory' requires our images to be in a particular directory structure to infer labels automatically. The directory structure should be such that the main directory consists of different sub-directory. The name of all sub-directory should be equal to the label name and contain all the images from the same label. For example in our case, we have 12 different sub-directory with images whose label is equal to the name of the sub-directory

```
plant_images/
    --complex/ 641 images
    --frog_eye_leaf_spot/ 1272 images
    --frog_eye_leaf_spot complex/ 66 images
    --healthy/ 1850 images
    --powdery_mildew/ 474 images
    --powdery_mildew complex/ 35 images
    --rust/ 744 images
    --rust complex/ 39 images
    --rust frog_eye_leaf_spot/ 48 images
    --scab/ 1930 images
    --scab frog_eye_leaf_spot/ 274 images
    --scab frog_eye_leaf_spot complex/ 80 images
```

Figure 9: Directory of Images

9.2 Train, Validation, Test Split

We need to split our dataset into training, validation, and testing to evaluate the performance of our model. In this study, we have used 80% of the data for training, 10% data for validation, and the rest 10% data for testing.

9.3 Resize and Rescale

The size of the image must be equal to the input size of the model. Therefore, we must resize every image to the input size of the model. After resizing, the images need to be normalised by converting their pixel range to 0 and 1. Normalisation will help our deep learning model to perform better.

9.4 Data Augmentation

In data augmentation, we have augmented the image by randomly flipping, rotating and changing its contrast.

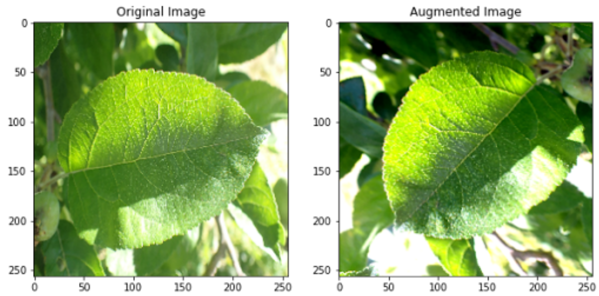

Figure 10: Data Augmentation

9.5 Model

We have used two different models to classify the images. The first model is a standard CNN model which we have built from scratch and the second model is a pre-trained MobileNet V2 model trained on ImageNet.

9.5.1 CNN Model

The summary of the model is shown in figure 11. The model has six convolution layers followed by a MaxPooling layer. After flattening, we have a dense layer of 64 neurons and then an output layer with 12 neurons as we have 12 different types of labels. In total, we have 184, 332 trainable parameters.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2D  (32, 127, 127, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 64)         0
 2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPooling  (32, 30, 30, 64)         0
 2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 64)          36928

 max_pooling2d_4 (MaxPooling  (32, 6, 6, 64)           0
 2D)

 conv2d_5 (Conv2D)           (32, 4, 4, 64)            36928

 max_pooling2d_5 (MaxPooling  (32, 2, 2, 64)           0
 2D)

 flatten (Flatten)           (32, 256)                 0

 dense (Dense)               (32, 64)                  16448

 dense_1 (Dense)             (32, 12)                  780

=================================================================
Total params: 184,332
Trainable params: 184,332
Non-trainable params: 0
_________________________________________________________________
```

Figure 11: CNN Model Summary

9.5.2 MobileNet V2

MobileNet V2 is a family of neural network architectures for efficient image classification and related tasks originally published by Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen in 2018(Sandler et al.; 2018). In this study, we have used the TF-Slim implementation of mobilenet_v2 provided by the TF-hub module which has an input size of 224x224. To fine-tune mobilenet_v2, we have unfrozen its last layer and have added an extra dense layer of 12 neurons as its output layer because we have 12 different image labels. The model summary is shown in figure 12

```
Model: "sequential_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_17 (Sequential)  (32, 224, 224, 3)         0

 sequential_18 (Sequential)  (32, 224, 224, 3)         0

 keras_layer_2 (KerasLayer)  (32, 1280)                2257984

 dense_11 (Dense)            (32, 12)                  15372

=================================================================
Total params: 2,273,356
Trainable params: 15,372
Non-trainable params: 2,257,984
_________________________________________________________________
```

Figure 12: Mobilenet V2 Model Summary

10 Evaluation

All the models are evaluated based on their accuracy on the training, validation, and test dataset. The models are trained on the training dataset till 50 epochs. After training, the models are evaluated on validation and test dataset. The best model is the one with the highest accuracy with any signs of overfitting or underfitting.

10.1 Evaluation of Models Without Data Augmentation

10.1.1 CNN

The performance of the CNN model without data augmentation on the training and validation dataset is shown in figure 13 and on the test dataset is shown in figure 14. On the training dataset, we have got an accuracy of 98.07%, on validation, the accuracy is 94.02% whereas, on the testing dataset, the accuracy is 94.01%.

10.1.2 MobileNet V2

The performance of the Mobilenet V2 without data augmentation on the training and validation dataset is shown in figure 15 and on the test dataset is shown in figure 16. On the training dataset, we have got an accuracy of 93.29%, on validation, the accuracy is 90.62% whereas, on the testing dataset, the accuracy is 88.28%.

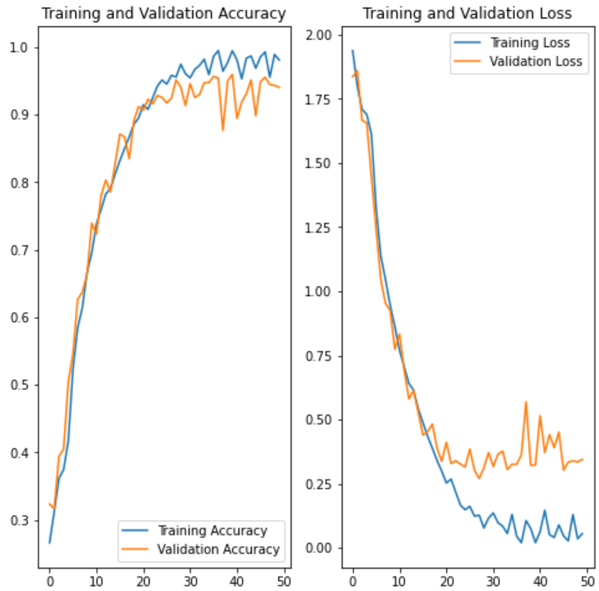

Figure 13: Model Performance - CNN without Data Augmentation

```
scores = model.evaluate(test_ds)
24/24 [==============================] - 76s 61ms/step - loss: 0.4913 - accuracy: 0.9401
```

Figure 14: Model Evaluation on Test Dataset- CNN without Data Augmentation

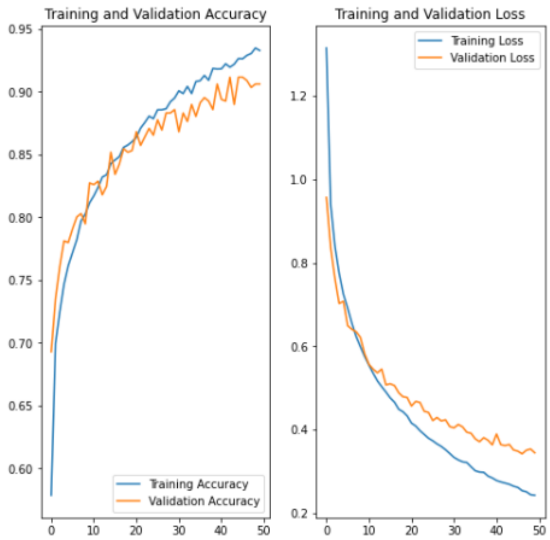

Figure 15: Model Performance - MobileNet V2 without Data Augmentation

```
scores = model.evaluate(test_ds)
24/24 [==============================] - 73s 71ms/step - loss: 0.4757 - accuracy: 0.8828
```

Figure 16: Model Evaluation on Test Dataset - MobileNet V2 without Data Augmentation

10.2 Evaluation of Models With Data Augmentation

10.2.1 CNN

Figure 17 shows the performance of CNN model with data augmentation. The training and validation accuracy is 80% and 81.66% respectively.

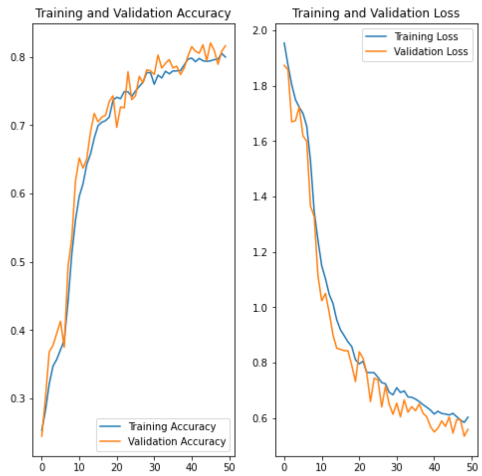

Figure 17: Model Performance - CNN with Data Augmentation

The figure 18 shows the performance of the testing dataset. The accuracy for the same is 82.16

```
scores = model.evaluate(test_ds)
24/24 [==============================] - 72s 59ms/step - loss: 0.5245 - accuracy: 0.8216
```

Figure 18: Model Evaluation on Test Dataset - CNN with Data Augmentation

10.3 MobileNet V2

The accuracy of MobileNet V2 with data augmentation on training and validation dataset is 75.76% and 77.17% respectively. Whereas on the testing dataset its accuracy is 76.30%. The performance of the same is shown in figure 19. Figure 20 shows the model performance on testing dataset.

11 Result Analysis

In Figure 21, we have two Tables. First Table shows the accuracy of CNN and MobileNet V2 without Data Augmentation whereas Second Table shows the accuracy with Data Augmentation for the same models.

From the above tables, it is evident that the CNN model performed the best without any data augmentation. It gave us an accuracy of 94.02% on the validation dataset and 94.01% on the test dataset. The model does not show any signs of overfitting as the difference between the train, validation, and test dataset is not significant. The same model

Figure 19: Model Performance - MobileNet V2 with Data Augmentation

```
scores = model.evaluate(test_ds)
24/24 [==============================] - 71s 86ms/step - loss: 0.6641 - accuracy: 0.7630
```

Figure 20: Model Evaluation on Test Dataset - MobileNet V2 with Data Augmentation

	Train	Validation	Test
CNN	98.07%	94.02%	94.01%
MobileNet V2	93.29%	90.62%	88.28%

	Train	Validation	Test
CNN	80%	81.66%	82.16%
MobileNet V2	75.76%	77.17%	76.30%

Figure 21: Model Accuracies

also gave us the second-best performance when we applied the data augmentation to the dataset. MobileNet V2 also performed well on training, validation, and test datasets without any data augmentation. However, its accuracy is still less than the CNN model which we have built from scratch. One thing we can observe from the above tables is that the performance of our models dropped significantly when we applied data augmentation. One reason for this performance drop could be that our original data itself is quite good and data augmentation is depreciating the quality of the same.

12 Error Analysis

One major hurdle we faced while conducting this study was memory limitation. The size of the dataset we have is approximately around 15GB. Our system crashed many times while computing the whole dataset because we ran out of memory. To compute this amount of data we needed more memory which was not available on our end. Thus to overcome this, we reduced the dataset to 40%. The second major hurdle was the time required to train the models. The CNN model which we used has 184,332 trainable parameters. Thus, the total time to train the model was enormous.

The TensorFlow library also came in handy to not only solve our memory problem but also our time problem. To solve all of these major hurdles, we created an input pipeline using a TensorFlow function 'image_dataset_from_directory'. This returns us a 'tf.data.Dataset' that yields batches of images with their corresponding label from the subdirectories. This helps us to handle huge datasets by streaming them from disk in batches. To take this a step further, we have used the prefetch and cache function from TensorFlow. In simple terms, prefetch fetches the next batch of data using CPU when our GPU is training the previous batch of data. As a result, when our GPU is training the previous batch of data, the CPU does not remain idle and starts fetching the next batch of data for our GPU. On the other hand, the cache saves all the preprocessed data before training in the first epoch. So from the next epoch onwards, our input pipeline does not need to do all the preprocessing which was needed to perform before feeding the dataset to our model as we have already saved that preprocessed data in memory using cache. Thus, it saves us a lot of time by not performing all of the redundancies again for each epoch. As a result, the total time to train our models dropped significantly.

13 Conclusion and Future Work

The main aim of our study was to use deep learning techniques to increase the efficiency of foliar disease identification in apple trees. For that, we use the Plant Pathology 2021 dataset which consists of 12 different foliar diseases. We performed various preprocessing to our dataset and implemented two deep learning models. Our best model was able to achieve a remarkable accuracy of 94.01% on the test dataset. Using this model, we can substantially increase the efficiency of foliar disease identification.

As we have only used 40% of the data to train the model because of limited resources. In the future, we can implement the same model on the complete dataset and analyse its result. Different data augmentation techniques can also be used other than the one which was used in this study. We can play around with the CNN model to increase its accuracy even more than one we have got. TensorFlow Hub has several pre-trained models which can be used to classify the images. A pre-trained model other than MobileNet V2 also can be used to identify the foliar disease.

14 Acknowledgment

I would like to sincerely thank professor Jorge Basilio. The professor has shared his knowledge and helped me with the queries during the research work. He has always answered to my queries via email and during the session. Professor thank you so much for your time and helping me to complete this research work.

References

Albawi, S., Mohammed, T. A. and Al-Zawi, S. (2017). Understanding of a convolutional neural network, *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6.

Ali, H., Lali, M., Nawaz, M., Sharif, M. and Saleem, B. (2017). Symptom based automated detection of citrus diseases using color histogram and textural descriptors, *Computers and Electronics in Agriculture* **138**: 92–104.
URL: *https://www.sciencedirect.com/science/article/pii/S0168169916309838*

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A. and Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection, *Symmetry* **11**(7).
URL: *https://www.mdpi.com/2073-8994/11/7/939*

Badage, A. (2018). Crop disease detection using machine learning: Indian agriculture, *Int. Res. J. Eng. Technol.(IRJET)* **5**(9): 866–869.

Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition, *Biosystems Engineering* **172**: 84–91.
URL: *https://www.sciencedirect.com/science/article/pii/S1537511018303027*

Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on pattern analysis and machine intelligence* (6): 679–698.

Chuanlei, Z., Shanwen, Z., Jucheng, Y., Yancui, S. and Jia, C. (2017). Apple leaf disease identification using genetic algorithm and correlation based feature selection method, *International Journal of Agricultural and Biological Engineering* **10**(2): 74–83.

Dongare, A., Kharde, R., Kachare, A. D. et al. (2012). Introduction to artificial neural network, *International Journal of Engineering and Innovative Technology (IJEIT)* **2**(1): 189–194.

Goncharov, P., Ososkov, G., Nechaevskiy, A., Uzhinskiy, A. and Nestsiarenia, I. (2018). Disease detection on the plant leaves by deep learning, *International Conference on Neuroinformatics*, Springer, pp. 151–159.

Hu, M., Bu, X., Sun, X., Yu, Z. and Zheng, Y. (2017). Rape plant disease recognition method of multi-feature fusion based on d-s evidence theory, *Mathematical and Computational Applications* **22**(1).
URL: *https://www.mdpi.com/2297-8747/22/1/18*

Jiang, P., Chen, Y., Liu, B., He, D. and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks, *IEEE Access* **7**: 59069–59080.

Kao, I.-H., Hsu, Y.-W., Yang, Y.-Z., Chen, Y.-L., Lai, Y.-H. and Perng, J.-W. (2019). Determination of lycopersicon maturity using convolutional autoencoders, *Scientia Horticulturae* **256**: 108538.
URL: *https://www.sciencedirect.com/science/article/pii/S0304423819304248*

Korkut, U. B., Göktürk, Ö. B. and Yildiz, O. (2018). Detection of plant diseases by machine learning, *2018 26th Signal Processing and Communications Applications Conference (SIU)*, IEEE, pp. 1–4.

Liu, B., Zhang, Y., He, D. and Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks, *Symmetry* **10**(1): 11.

Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks, *Neurocomputing* **267**: 378–384.
URL: *https://www.sciencedirect.com/science/article/pii/S0925231217311384*

Mohanty, S. P., Hughes, D. P. and Salathé, M. (2016). Using deep learning for image-based plant disease detection, *Frontiers in plant science* **7**: 1419.

Pardede, H. F., Suryawati, E., Sustika, R. and Zilvan, V. (2018). Unsupervised convolutional autoencoder-based feature learning for automatic detection of plant diseases, *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, IEEE, pp. 158–162.

Pujari, J. D., Yakkundimath, R. and Byadgi, A. S. (2013). Grading and classification of anthracnose fungal disease of fruits based on statistical texture features, *International Journal of Advanced Science and Technology* **52**(1): 121–132.

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

Rumpf, T., Mahlein, A.-K., Steiner, U., Oerke, E.-C., Dehne, H.-W. and Plümer, L. (2010). Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance, *Computers and Electronics in Agriculture* **74**(1): 91–99.
URL: *https://www.sciencedirect.com/science/article/pii/S0168169910001262*

Saleem, M. H., Potgieter, J. and Arif, K. M. (2019). Plant disease detection and classification by deep learning, *Plants* **8**(11): 468.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks, pp. 4510–4520.

SARDOĞAN, M., Yunus, Ö. and TUNCER, A. (2020). Detection of apple leaf diseases using faster r-cnn, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi* **8**(1): 1110–1117.

Selvaraj, M. G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W. and Blomme, G. (2019). Ai-powered banana diseases and pest detection, *Plant Methods* **15**(1): 1–11.

Sinha, A. and Shekhawat, R. S. (2020). Review of image processing approaches for detecting plant diseases, *IET Image Processing* **14**(8): 1427–1439.

Sun, J., Yang, Y., He, X. and Wu, X. (2020). Northern maize leaf blight detection under complex field environment based on deep learning, *IEEE Access* **8**: 33679–33688.

Turkoglu, M. and Hanbay, D. (2019). Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine, *Physica A: Statistical Mechanics and its Applications* **527**: 121297.
URL: *https://www.sciencedirect.com/science/article/pii/S0378437119307551*

Wikipedia (2021). Apple. Last accessed 21 November 2021.
URL: *https://en.wikipedia.org/wiki/Apple*

Wrzesień, M., Treder, W., Klamkowski, K. and Rudnicki, W. R. (2019). Prediction of the apple scab using machine learning and simple weather stations, *Computers and Electronics in Agriculture* **161**: 252–259. BigData and DSS in Agriculture.
URL: *https://www.sciencedirect.com/science/article/pii/S0168169917307676*

Yang, X. and Guo, T. (2017). Machine learning in plant disease research, *March* **31**: 1.