**Why should machine learning be used in determining crop disease in plants?**

David Chen

Altrincham Grammar School for Boys, UK

*Abstract*— Plant diseases are defined as an abnormal growth or dysfunction within a plant. Crop diseases affect plants that are grown on a large scale which can result in many different problems for society, ranging from a drastically lower economic output, to a lack of food supply in third world countries. Current plant disease identification is often done manually, which is time consuming and costly, and prone to error due to subjectiveness caused by humans. Machine learning has emerged as an effective approach for automated plant disease detection. The aim of this report is to examine how current machine learning approaches can be used for automatically detecting crop disease. We have applied different machine learning methods to a publically available dataset in order to detect crop disease (i.e. rust, leaf spot) in this case.

Keywords—Image processing, pattern recognition, machine learning, crop disease,

# Introduction

As the population of the globe increases exponentially, projected at nearly 10 billion by 2020[[1]](#footnote-1), being able to balance converting unused land into habitable land, and farmable land becomes more difficult to manage. The number of people that suffer from food shortages is staggering. There are 34 countries that cannot provide enough food for their people[[2]](#footnote-2). Within the world, almost half the entire population lives in poverty[[3]](#footnote-3) . Out of the 3 billion people living in poverty, over 600 million live in extreme poverty[[4]](#footnote-4). This means that these people are unable to afford the food which is provided. This is also a problem due to the weighted supply and demand of food in the poorer countries. By increasing the supply of food, the prices will be on track to fall with it, allowing the people living in poverty to be able to afford the basic necessity that is food. The rising problem of needing more farmland to feed the exponentially increasing population means that it is critical to maximise the gains from each farmland for improved food production. Crop plants are the largest sources for feeding the population. The use of crops ranges from eating crops like wheat in processed products, to feeding the animals that will be eaten. The biggest loss of crops is due to crop diseases. Crop disease causes significant damages to the crops and reduces crop yield, leading to potential economic losses as well as a lack of food for the population. Therefore, it is becoming a global threat for food sustainability. To reduce this threat, there is a need for us to develop an effective way to identifying crop diseases as early as possible in order to increase food production.

A disease of a crop plant refers to an abnormal growth or dysfunction within a plant[[5]](#footnote-5). When a plant is affected by diseases, it normally shows visual symptoms or detectable changes such as in colour, shape that occur on the leaves. As disease progression, the colours, size could change.

Currently, the most common technique used to identify crop disease is by manual visual identification. The problem with identification using the naked eye is that a lot of agronomists or surveyors are needed in order to cover a large farm area effectively. This also brings in each person’s objective view. As one person may see crop disease in one plant, another may not. These inconsistencies could result in different plants being wrongly flagged. In addition, sending out people to go to the field in order to do manual surveying is very labour intensive and time consuming due to the sheer size of the fields. It is also very expensive to do this as they are often required to be trained regularly to keep update with new knowledge about diseases, as well as having to move them to different fields to do their work. This is particularly difficult to achieve, especially in rural countries, where it is hard for agronomists or surveyor to access the fields due to poor infrastructures such as roads, transport and funding issues.

To help alleviate this problem, the development of using image processing and machine learning in order to detect diseases in plants unravelled rapidly. Research was done on detecting Wheat Leaf rust using distinguished partial least squares and support vector machines[[6]](#footnote-6) . Another approach was proposed, using random forests and hyperspectral imagery [[7]](#footnote-7) to detect any weeds within a maize crop. While not the same as detecting crop disease itself, this method can be adapted to find the specific crop diseases by using the training sets for those. In itself, detecting weeds within a crop field is also very useful to farmers, as it identifies areas which are not productive and would not have any yield. This would allow the problem to be treated with quickly, to improve the entire crop field yield.

In this report, a real publically available dataset for various crops and diseases will be used to train different machine learning algorithms to draw comparisons and to show how effective machine learning is in detecting crop disease.

The rest of this report is organised as follows. Section 2 presents the related work in automatic diagnosis of crop diseases using image processing and machine learning methods; Section 3 describes the machine learning methods used for determining crop disease in this report; Section 4 conducted experimental evaluation with real datasets; Section 5 concludes the work and highlights future work.

# Related work

There have been many efforts in studying the image processing techniques to identify and classify diseases like fungal disease symptoms on agriculture crops[[8]](#footnote-8). For instance, a proposed method to detect and diagnose plant disease involved the use of convolutional neural networks[[9]](#footnote-9). Convolutional neural networks (CNN) is a type of neural network that is normally applied to analysing images. CNN itself requires very minimal pre-processing, which drastically reduces operation time. Other researchers utilised different methods, for example, using a kernel-based support vector machine (SVM)[[10]](#footnote-10) to detect diseases like rust, tikka, powdery and downy mildew, late blight and early blight in groundnut, apple, potato and tomato plants. K-Nearest neighbours (KNN) was also utilised in different situations [[11]](#footnote-11)to detect tomato plant diseases, and pest recognition. Random forests were used in detection of grape diseases using pictures from a wild environment[[12]](#footnote-12). More experiments were carried out using colour and shape patterns in 425 images of 3 types of rice diseases, which was later separated into 4 distinct classes to further increase precision in detecting what type of disease is present[[13]](#footnote-13).

It would seem that using image classification to classify plant diseases should be perfect now. However the current models are not accurate due to low sample sizes and small representation of the diseases. There are also some limitations to using machine learning in order to detect crop disease at this moment. The first of which lies in the set of images used. These images may be trained to detect a specific type of disease, like rust. However this means that the classifier will only be trained to detect these diseases. For example if we were to give the classifier a set of images of the leaf spot when the classifier was trained on detecting rust, we would not receive an accurate prediction. This would then mean that another set of images would be needed to train for another specific disease, which would include thousands more images to run through the code. Extending on from this, multiple types of images are needed for diseases that spread throughout the plant. As some plants may have diseases that start from the stem before reaching the leaves, we would also need to include pictures of this. Therefore, more efforts should be made to obtain images from different perspectives. Another problem with training an accurate classifier is the picture quality used. Many different situations within a picture could potentially change the boundaries of the class of disease. Pictures can be taken from different angles, different lighting, in different seasons and in front of an environment background. As the classifier is trained based on the actual picture itself, it becomes harder to normalise what would be an ideal model for that specific disease.

There are also over 50000 classes of crop species and diseases[[14]](#footnote-14). The practicality of training every single plant to every single disease is poor and a lot of time would be required to train every plant with each disease, even with the highest end computer specifications. With these many diseases it is also very likely several diseases have many similar features with each other, and the accuracy of the classifier to correctly predict a disease would be lowered.

# The application of The machine learning methods to crop disease detection

The purpose of this report is to examine how existing commonly used machine learning approaches can be used to detect crop diseases and the level of their accuracy. In this section, the background of machine learning and selected machine learning methods will be explored.

## Machine learning methods

Machine learning is the field of study that gives the computer ability to learn without being explicitly programmed[[15]](#footnote-15). Essentially, it is to build models that receives data and information from observations and real world interactions to learn the patterns from. It will then be able to provide accurate predictions based on the learned patterns of input data.

Broadly, machine learning can be divided into supervised learning and unsupervised learning. Supervised machine learning is what makes up most practical machine learning uses. Supervised learning utilises a function using a variable as an output dependent on the variable of the input. This is also commonly referred to

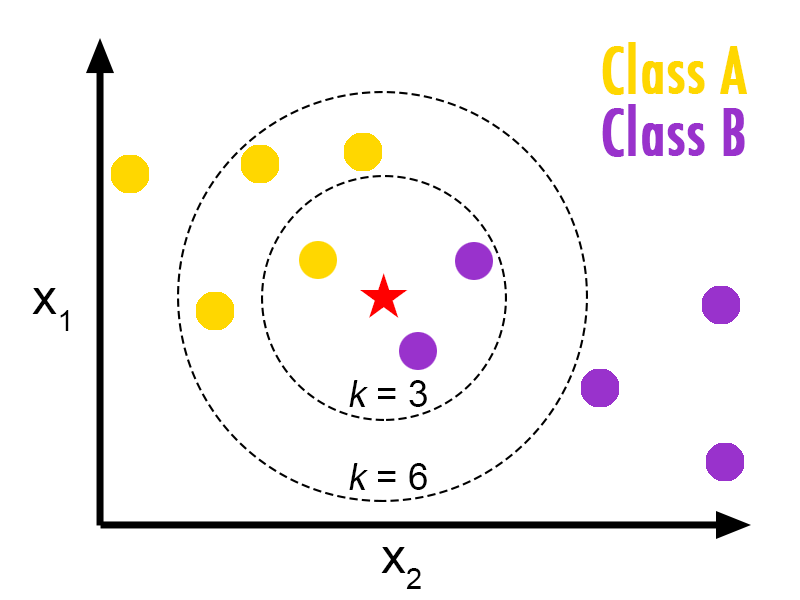
The goal is to approximate the function ***f***, so that when new input data ***x*** is provided, the output data ***y*** can be predicted for that new input variable. It is supervised because the user knows the correct answers ***y*** for the input ***x***, and the algorithm makes predictions ***y*** on training data ***x***, which the operator then corrects if it is wrong. The algorithm gradually learns from this until the predictions are acceptable. For example, the least of square error formula:

There are often two stages to consider when building supervised machine learning algorithms. These are the training, and testing stages. In the training stage, both input data and desired output are provided with feedbacks on the outcomes for building machine learning models. Once the training is completed, the trained models will be applied to the test data or new data during the testing phase. A popular example is the random forest algorithm.

In unsupervised learning, only the input data is provided without any output data. The point of this is to model the structure within the data to expose the patterns inside it. For unsupervised learning, it normally doesn’t need to provide desired output during the training stage. In this report, we only focus on supervised learning.

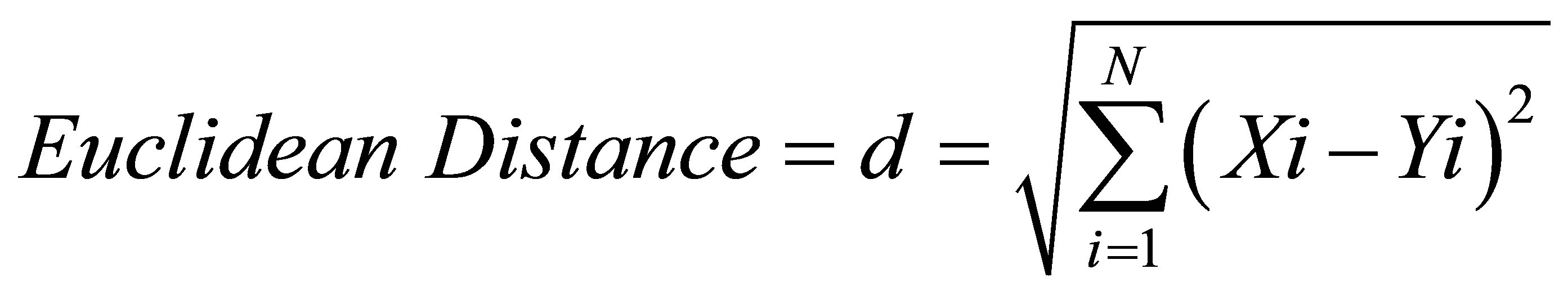
### KNN

KNN, or K-Nearest Neighbours is a popular supervised machine learning algorithm used commonly in pattern recognition. KNN utilises a graph that displays a pseudo distance between various points. Each training data point is assigned a class, which takes up a part of the graph. When a new point, or the test data, is inputted into the graph, we can set any number “k” of the nearest points. The number of each class of these nearest points, or neighbours, determines what the class of the new point is. This is visualised by the following:



***Figure 1: KNN algorithm concept[[16]](#footnote-16)***

In this case, the pseudo distance between each point is also known as the Euclidean distance between each point as the data variables are continuous. The formula for the Euclidean distance is given as



Where we wish to find the difference between points X and Y, which represent the value for various attributes, like colour and texture, in an nth dimension. With regards to the picture above, if class A was “healthy” for a plant and class B was “diseaseX”, then the test input/star picture would be classed as “diseaseX” simply because out of all the “k” values, there are more points representing class B.

### Random Forests

Random Forests is also another supervised machine learning method. It utilises the construction and merging of multiple decision trees during the training stage, which will all each output the class prediction of the test data during the testing stage. It is different from a normal decision tree on its own however, as instead of formulating a set of rules to make predictions, the random forest randomly selects observations in order to create several decision trees, and each result from each tree will be averaged.



***Figure 2: A random forest depiction, containing three separate decision trees [[17]](#footnote-17)***

For example, if the instance was a picture of a corn plant, that is yet to be classified, tree 1 could be classification based around the colours of the image. The images of the different diseases in the training sample are extracted based around these features like colour, texture and shape. The new test image would be run through each of the trees, tree 1 being colour, tree 2 being texture, and tree 3 being shape. Each individual tree classifies the image based around their own parameters, and sends the vote to the main root of the tree. The majority vote is then taken as a result from each tree.

### Logistic Regression

Logistic regression[[18]](#footnote-18) is a prediction method giving well calculated probabilities. In this context, there are multiple different classes, so a variant called Multinomial Logistic Regression will be used.

As this is a binary classifier, multiple classifiers must be trained in order to create an accurate model. When training the classifier for the disease Leaf Spot, input data is treated with this label as positive samples, and the other three diseases as negative. Similarly, for the disease rust, we treat the data with this label as positive, whilst the other three diseases are negative samples. This is done for all four categories during the training phase. During the prediction phase, the testing data is run against all four models to return their probability of whether it is that disease or not. Then, the highest probability class is chosen between the four different class predictions.

### Decision Tree classifier

Similar to Random Forests, a decision tree[[19]](#footnote-19) is a type of classifier that works by branching different nodes on a situation. Each internal node may represent a test on that situation. There three types of nodes that make up a decision tree, a decision node, chance node and end node. The decision tree breaks down a data sample into smaller subsets, to create an end product of decision and leaf nodes (node that doesn’t split). Each decision node has two or more branches. Leaf nodes represent a classification. The highest node in the tree is also named as the root node. After splitting the data, the tree is pruned so that the size of tree is reduced. This is often done when the decisions after a node is more often than not one choice, therefore that branch is usually replaced by that one choice.

Take a plant, like a tomato. The largest attribute would go at the top, like its colour for example. The questions asked would follow on from that value, like “Is the value of the plant>x”, in which it branches down. These nodes would narrow down until the end nodes are reached.

### Linear discriminant analysis (LDA)

LDA[[20]](#footnote-20) is a technique that is a preferred method when there are more than two classes, such as with the case in this report. Linear discriminant analysis (LDA) assumes that each piece of data has a distribution like a curve on a graph. LDA also assumes each attribute has the same variance, and each value vary around the mean by the same on average. Basing on these assumptions, LDA estimates mean and variance for each class. LDA converts the plant dataset from an n-dimensional space into a smaller subspace. The higher dimensional space means the more accurate we can be in determining which value belongs to which class. LDA uses the information from the nth amount of axis to create a new axis, and the data is now projected onto this axis to maximize separation between the 4(or how many diseases there are) categories. The new axis is found after finding the central point of all data, the distance between each central point of each category and dividing the sum of the distances to the central point squared by the sum of all the scatter for each category. LDA works very similar to KNN.

***Figure 3: An example chart utilizing LDA showing how a new axis would be drawn for two categories in 2 dimensions***

### GaussianNB(Naïve Bayes)

Naïve Bayes[[21]](#footnote-21) is a collection of algorithms that classifies all features independently. It uses the probability of each independent feature appearing in a data sample by matching with the training set. For example, a data sample with the categories shape and colour, matching to either an apple or banana, would predict a new piece of data as an apple if it was coloured red and had a round shape. This is due to the probability of an apple being red vs a banana being red. Similarly in crop disease, we can find the probability that a disease is rust if the shape of the outline of the colour contrast is classified as a dot.

### 7)Support vector machine (SVM)

SVM is one of the most popular machine learning methods. Given some training data samples, with labels, the algorithm builds a model that assigns new data samples to any label. SVM’s objective[[22]](#footnote-22) is to find a hyperplane (space within a dimension that has one less dimension than the dimension it is set in) in a set dimension of space, where the dimension is the same as the number of features, which distinctly classifies the data points. When dealing with multiple classes in a dataset, SVM often uses the pairwise/one-vs-one feature. In this one-vs-one feature, the idea is to develop a number of binary classifiers while being trained only on two classes. The formula to determine how many classifiers will be trained is given by

https://i.gyazo.com/fb45748b99fc1426ea8a98e57d8665ad.png

Where n is equal to the number of classes. Using the corn dataset as an example, there are four classes of diseases, so there are (4\*3)/2 = 6 classifiers. The diseases “healthy” and “leaf spot”, “healthy” and “Northern Leaf Blight”, “healthy” and “rust” etc... pair up as individual classifiers until all diseases match up with all the others. If a new test image, Y was inputted, and the disease was unknown, Y would be presented to each classifier. The classifier with the highest probability will be taken as the output.

## The proposed framwork for crop disease detection

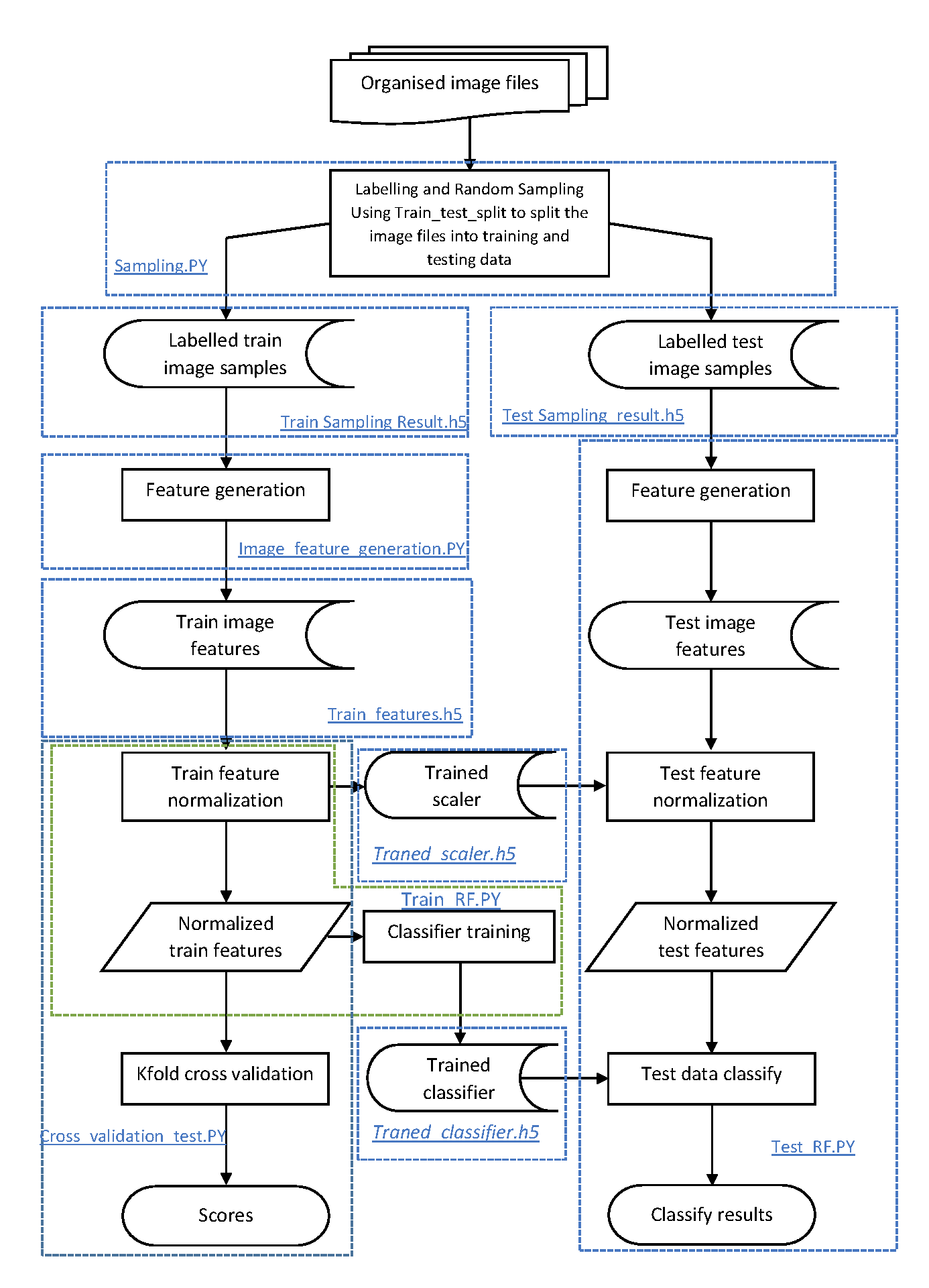
As this report mainly focuses on supervised machine learning methods, the following will describe how supervised machine learning methods work.

There will be two phases: the training stage and the testing stage.

Given the dataset, it will split into two parts: the training dataset and the testing dataset. During the training stage, the input images will be processed and then distinctive features in relation to disease patterns will be extracted from these images. Based on the extracted features, the classification models will be built.

For the testing stage, the test images are different from the training images, which were never trained on the classifier. The classifier here extracts the features from each image and applies the values for these features to the disease parameters set by the model earlier during the training stage.

The following diagram outlines the stages that results in a class prediction in more detail.



***Figure 4: The workflow of image classification***

# The Implementation and experimental evaluation

## The implementation

Python has been used for development of the machine learning methods for crop disease classification. The classification methods include KNN, SVM, LDA, LR, CART, NB, RF. In preparation for the experimental process, a free and open source distribution for Python, named Anaconda[[23]](#footnote-23) was installed. Within Anaconda, there are various options that may allow to easily install and apply different modules to solve this case including: mahotas, scikit learn, and numpy. As shown in Fig.4, there are several different components were created in order for this experiment to function.

There are 6 python programs have been created for the training testing stages.

For the training stage, the python programs include:

* 1. featuregen.py: this program pre-defines the feature generation method and data saving method. This program then will be imported into the training and testing programs.
  2. Sampling6.py: this program will firstly read all the sample file names, and then input the dataset folder (dataset/Corn) to a list, labelling each sample numerically while saving the labels to another list. By using the “train\_test\_split” function, the train and test sample files will be generated with names and labels. This program will then save those data lists in two data files (“output/Corn\_train\_samples6.h5”, and “output/Corn\_test\_samples6.h5). The test sample set size is 0.2(20%) of all total samples.
  3. Image\_feature\_generation6.py: this program reads the train samples created from the sampling result file (“output/Corn\_train\_samples6.h5”) and generates the global training features. The features are saved in an h5f file to be imported later. (“output/Corn\_train\_features6.h5”).
  4. Method\_compare6.py: this program uses a 10 fold cross validation method (kfold and cross\_val\_score function) to compare several machine learning methods in disease detection on the training features file generated in section 2.3. The features are normalised with respect to 0-1 before undergoing validation.
  5. Train\_RF61.py: this program uses the training sample features dataset generated in section 2.3 to train the scaler and various classifier models. The classifier is trained by the normalised global training features which is read from the h5f file generated earlier. The trained scaler and classifier are saved in two files: “output/Corn\_train\_scaler.h5” and “output/Corn\_train\_model.h5”, respectively.

For the testing stage, the python program is:

1) Test\_RF61.py: this program tests one machine learning method, random forests in this case, in detecting disease using the test data features. The global test features data is generated from the samples specified in the list in test sample file (“output/Corn\_test\_samples6.h5) using the predefined function “feature\_generate” in section 1. Generated global test features are normalized by the scaler which is trained by train globe feature data in section 2.5(“output/Corn\_train\_scaler.h5”). The results are used to calculate the real positive, real negative, false positive and false negative results.

The files are named “RF” to begin with due to the fact that the machine learning approach random forests will be the first one to be tested.

## The experimental evaluation

### Datasets

There are 5 different plants that will be investigated, each with its own number of diseases. There are many samples of each disease for each plant. The datasets are publicly available[[24]](#footnote-24)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Apple** | Apple scab | Cedar apple rust | Frogeye spot | Healthy |
| Number of samples | 630 | 275 | 621 | 1645 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Corn** | Leaf spot | Northern leaf blight | Rust | Healthy |
| Number of samples | 513 | 985 | 1192 | 1162 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grape** | Black rot | Esca black measles | Leaf blight/ isariopsis leaf spot | Healthy |
| Number of samples | 1180 | 1383 | 1076 | 423 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Potato** | Early blight | Late blight | Healthy |
| Number of samples | 1000 | 1000 | 152 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tomato** | Bacterial spot | Early blight | Late blight | Leaf mold | Septoria leaf spot | Spotted spider mite | Target spot | Mosaic virus | Yellow leaf curl | Healthy |
| Number of samples | 2127 | 1000 | 1909 | 952 | 1771 | 1676 | 1404 | 373 | 5357 | 1691 |

Each classifier will be trained onto each of these diseases.

### Evaluation metrics

In order to measure accuracy, the accuracy measure is based on the standard metrics as follows:

accuracy = (tp+tn)/(tp+fp+tn+fn)

Where “tp” means true positive, “tn” represents true negative, “fp” represents false positive and “fn” represents false negative.

* A true positive is a positive sample that was predicted as positive by the classifier. (e.g. if a disease was rust, and the classifier predicted it as rust)
* A true negative is a negative sample being predicted as negative by the classifier. (e.g. if a disease was not rust and the classifier predicted as not rust)
* A false positive is a negative example being classified as positive by the classifier. (e.g. if disease was not rust, but the classifier predicted it as rust)
* A false negative is a positive example being classified as negative by the classifier. (e.g. if the disease was rust but the classifier predicted it as not rust)

### Experimental setup

The experiment was run on a computer with the following specs:

Processor: Intel(R) Core(TM) i3-4130 CPU @ 3.40GHz, 3400 Mhz, 2 Core(s), 4 Logical Processor(s)  
Operating system: Windows 10 Home  
Random access memory(RAM):8GB  
Graphics card: Nvidia Geforce GTX 1060

To run the programs mentioned in section A, the sample data files are organised as such:

1. Using the class name as the subfolder name of the sample files of this class
2. Using the plant name as the plant/crops data folder
3. The folder must be under the same folder of the code written

This is done so for convenience sake. Using this modification, any plant/crop disease can be tested on without modifying the actual code. The only requirement is to input the name of the plant/crop when the program is running.

The programs are run in the following order.

1. Sampling6.py to split samples into training and testing datasets
2. Image\_feature\_generation6.py to generate the training sample features
3. Image\_feature\_generation6.py to generate testing sample features
4. Method\_compare6.py to compare the 7 machine learning methods using k-fold cross validation
5. Train\_RF61.py to train the random forest classifier and scaler(can be switched to any other ML algorithm)
6. Test\_RF62.py to test the trained classifier and scaler using the testing sample dataset.

### Experimental results

For the 5 different plants, 5 data files were generated for each plant. Table 1 shows the accuracy results for 5 different plants. As shown in Table 1, we can find that the random forests algorithm provides the very best accuracy for each plant and its diseases. KNN follows random forests with the second highest accuracy, however it struggles with corn related diseases. Decision Tree classifier is bad when classifying 9 classes, as seen in the tomato investigation. This suggests that CART is poor when dealing with multiclass classifications. GaussianNB and Support Vector Machines provide poor accuracy, hinting that these algorithms are not ideal when classifying crop diseases. SVM also requires a lot more computational power and time in comparison to the other methods, when increasing sample size. Therefore we can conclude that random forests and KNN would be the top methods for classifying crop disease.

Table 1. Accuracy results for different plants (test size=0.2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Corn** |  |  |  |
|  | Accuracy | STD | Min Accuracy | Max Accuracy |
| LR | 0.925674 | 0.011656 | 0.914018 | 0.93733 |
| LDA | 0.913341 | 0.016099 | 0.897242 | 0.92944 |
| KNN | 0.917234 | 0.011086 | 0.906148 | 0.92832 |
| CART | 0.920472 | 0.016212 | 0.90426 | 0.936684 |
| RF | 0.953582 | 0.010389 | 0.943193 | 0.963971 |
| NB | 0.893534 | 0.014619 | 0.878915 | 0.908153 |
| SVM | 0.846477 | 0.020133 | 0.826344 | 0.86661 |
|  |  |  |  |  |
|  | **Potato** |  |  |  |
|  | Accuracy | STD | Min Accuracy | Max Accuracy |
| LR | 0.972695 | 0.013259 | 0.959436 | 0.985954 |
| LDA | 0.957004 | 0.017475 | 0.939529 | 0.974479 |
| KNN | 0.976761 | 0.012188 | 0.964573 | 0.988949 |
| CART | 0.957585 | 0.01582 | 0.941765 | 0.973405 |
| RF | 0.981405 | 0.010004 | 0.971401 | 0.991409 |
| NB | 0.879698 | 0.018676 | 0.861022 | 0.898374 |
| SVM | 0.883193 | 0.027125 | 0.856068 | 0.910318 |
|  |  |  |  |  |
|  | **Tomato** |  |  |  |
|  | Accuracy | STD | Min Accuracy | Max Accuracy |
| LR | 0.898747 | 0.006125 | 0.892622 | 0.904872 |
| LDA | 0.880643 | 0.008943 | 0.8717 | 0.889586 |
| KNN | 0.936467 | 0.005799 | 0.930668 | 0.942266 |
| CART | 0.859788 | 0.008975 | 0.850813 | 0.868763 |
| RF | 0.971572 | 0.003109 | 0.968463 | 0.974681 |
| NB | 0.628512 | 0.017224 | 0.611288 | 0.645736 |
| SVM | 0.663616 | 0.009244 | 0.654372 | 0.67286 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Apple** |  |  |  |
|  | Accuracy | STD | Min Accuracy | Max Accuracy |
| LR | 0.958204 | 0.012988 | 0.945216 | 0.971192 |
| LDA | 0.950316 | 0.009198 | 0.941118 | 0.959514 |
| KNN | 0.959001 | 0.012945 | 0.946056 | 0.971946 |
| CART | 0.912857 | 0.012997 | 0.89986 | 0.925854 |
| RF | 0.981071 | 0.004932 | 0.976139 | 0.986003 |
| NB | 0.858833 | 0.021734 | 0.837099 | 0.880567 |
| SVM | 0.633315 | 0.032725 | 0.60059 | 0.66604 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Grape** |  |  |  |
|  | Accuracy | STD | Min Accuracy | Max Accuracy |
| LR | 0.949522 | 0.004612 | 0.94491 | 0.954134 |
| LDA | 0.952911 | 0.011673 | 0.941238 | 0.964584 |
| KNN | 0.959682 | 0.007067 | 0.952615 | 0.966749 |
| CART | 0.912897 | 0.020033 | 0.892864 | 0.93293 |
| RF | 0.976609 | 0.006906 | 0.969703 | 0.983515 |
| NB | 0.870433 | 0.025534 | 0.844899 | 0.895967 |
| SVM | 0.701443 | 0.028775 | 0.672668 | 0.730218 |

# Conclusion

Crop disease detection plays an important role in food security and sustainability. This report examines how machine learning mehods can be used in detection of crop diseases.

The decision made for doing this project is becuase I decided to do research on this topic due of the importance of food security, and just how much food consumption increases by each year, due to the increasing population size. Crops are a massive part of the global food supply. They could be grown to either directly feed people, or feed animals which then feed people. In 2016, there are 34 countries suffering from these food shortages. With these many countries facing food shortages, and with half the world’s population dealing with poverty, demand for food increases, but the supply cannot keep up. Habitats and wildlife are all suffering from the destruction of land to make room for farms, which also comes at an expense of land that homes can be built on. We can slow down this change, as currently, the number of crops lost to diseases is between 40-50%. We can slow this down by detecting the diseases early, and dealing with them so that they don’t affect the final crop yield. The current commonly used method to detect such crop diseases is through manual detection. While manual methods may be as much, if not more accurate than computational methods, manual analysis requires a lot more time since all recognition is done by fieldwork. The expenses to transport experts into these regions, is massively higher than the expenses required to take a picture of segments of a field and using a computer to do the analysis automatically. However, with the current pace of technological development overshadowing biological the most effective and efficient method or image detection and classification has to be selected to make a significant impact on this growth. Image based analysis has proven to have potential in automating non-invasive detection. Its execution can range from either quick snapshots or continued surveillance over a specific farmland.

I have conducted literature review studies with relations to machine learning uses in scenarios involving image detection and processing. I implemented techniques from these various papers into Python programs, which examine and compare the top seven machine learning algorithms on their accuracy. The machine learning algorithms I examined were K-Nearest Neighbours, Random Forests, Logistic Regression, Linear Discriminant Analysis, Gaussian Naïve Bayes, Decision Tree Classifier, and Support Vector Machines. Experimental results have shown that Random Forests have the highest mean accuracy, with consistently low standard deviation. This therefore means that Random Forests is the most effective algorithm to use when detecting crop diseases via image processing and analysis. Taking into account however, the nature of the study, it should be noted that this research was not completely perfect. For example, it took pictures of a single plant as data samples in the training and testing phases. This may not be fully conclusive in the applications of real world problems, as the samples from crop fields are likely to be of multiple plants within a single crop. The limitations of the machine learning methods are that it cannot clearly distinguish various shapes and individual plants when taken from a zoomed out picture. This means that in reality, when using this method, much of the samples will most likely be of small to medium sized, containing a few plants each. This will be done to find a compromise between accuracy and efficiency, as it will always be more accurate to test for diseases biologically. Another problem to consider would be the detection of visible symptoms. Not all symptoms of crop disease are visible ones, so this research cannot cover all crop diseases. On the other hand, however, many crop diseases do cover this. While it may not be possible to completely recover the lost 40% crops from crop disease, more will still be retained from early detection of this. Therefore, I can confidently state that while machine learning methods are not perfect within this application, it will save both time and money to upgrade to using these computational algorithms.

Word count : 5387

# References

(n.d.). Retrieved from Anaconda: https://www.anaconda.com/

Biswas, S., Avil, S., Sarangi, S., & Pappula, S. (2016). *Random forest based classification of diseases in grapes from images captured in uncontrolled environments.* IEEE.

Borra, S., & Pukkela, P. (2018). *Machine Learning Based Plant Leaf Disease Detection and Severity Assessment Techniques: State-of-the-Art.* ResearchGate.

Brostoff, B. (2018, January 27). *A brief look at sklearn.tree.DecisionTreeClassifier*. Retrieved from Hackernoon: https://hackernoon.com/a-brief-look-at-sklearn-tree-decisiontreeclassifier-c2ee262eab9a

Buerkle, T. (2006, October 09). *40 countries face food shortages worldwide*. Retrieved from FAONewsroom: http://www.fao.org/newsroom/en/news/2006/1000416/index.html

Cochran, Q. (2018). *Logistic Regression*. Retrieved from MedCalc: https://www.medcalc.org/manual/logistic\_regression.php

Elliot, S. (2015, April Tuesday). *Global scientists meet for integrated pest management idea sharing*. Retrieved from National Institute of Food and Agriculture: https://nifa.usda.gov/blog/global-scientists-meet-integrated-pest-management-idea-sharing

Fang, Y., & Ramasamy, R. (2015). *Current and Prospective Methods for Plant Disease Detection.* Athens: University of Georgia.

Ferentinos, K. (2018). *Deep learning models for plant disease detection and diagnosis.* Elsevier.

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). *A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests RecognitionA Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition.* US national library of Medicine .

Gale group. (2018, November 08). *Encyclopedia*. Retrieved from Plant Diseases: https://www.encyclopedia.com/literature-and-arts/biographies/french-canadian-literature-biographies/plant-diseases

Gandhi, R. (2018, June 07). *Support Vector Machine-Introduction to Machine learning Algorithms*. Retrieved from Towards Data Science: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Gao, J., Nuyttens, D., Lootens, P., He, Y., & Pieters, J. (2018). *Recognising weeds in a maize crop using a random forest machine learning algorithms and near infrared snapshot mosaic hyperspectral imagery.* Elsevier.

HelloACM. (2016, March 23). *A short Introduction to K-Nearest neighbours Algorithm*. Retrieved from Technology of Computing: https://helloacm.com/a-short-introduction-to-k-nearest-neighbors-algorithm/

Koehrsen, W. (2017, December 11). *Random Forest Explanation*. Retrieved from Medium: https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Maharjan, G., Takahashi, T., & Zhang, S. (2011). *Classification Methods Based on Pattern Discrimination models for web-based diagnosis of rice diseases.* David Publishing.

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., . . . Dandekar, A. (2014). *Advanced methods of plant disease detection.* France: INRA and Springer-Verlag .

Patil, S., Chandavale, A., & Dixit, B. (2015). *Recognition of plant disease.* Pune: MIT College of Engineering.

Pujari, J., Yakkundimath, R., & Byadgi, A. (2015). *Image Processing Based Detection of Fungal Diseases in Plants.* Elsevier.

Raschka, S. (2014, 08 03). *Linear Discriminant Analysis*. Retrieved from SebastianRaschka: https://sebastianraschka.com/Articles/2014\_python\_lda.html

Ray, S. (2017, 09 11). *6 Easy steps to learn Naive Bayes Algorithm*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Samuel, A. (1959).

Shah, A. (2013, January Monday). *Poverty Facts and Stats*. Retrieved from Global Issues: www.globalissues.org/article/26/poverty-facts-and-stat

Small, M. (2017, December). *Plant Pathology.* Retrieved from Colorado State University: cmg.colostate.edu/Gardennotes/331.pdf

spMohanty. (2018, 09 23). Dataset of diiseased plant leaf images and corresponding labels.

United Nations, D. (2015, July 29). *World population projected to reach 9..7 billion by 2050*. Retrieved from United Nations: http://www.un.org/en/development/desa/news/population/2015-report.html

VOANews. (2016, March 09). *UN: 34 countries don't have enough food for their people*. Retrieved from VOA News: https://www.voanews.com/a/united-nations-thirty-four-countries-lack-sufficient-food/3228567.html

Wang, H., Qin, F., Ruan, L., Wang, R., Liu, Q., Ma, Z., . . . Wang, H. (2016). *Identification and Severity Determination of Wheat Stripe Rust and Wheat Leaf Rust Based on Hyperspectral Data Acquired Using a Black-Paper-Based Measuring Method.* PLOS.

World Poverty. (2018, June). *World poverty clock*. Retrieved from World Poverty: https://worldpoverty.io/

1. (United Nations, 2015) [↑](#footnote-ref-1)
2. (VOANews, 2016) [↑](#footnote-ref-2)
3. (Shah, 2013) [↑](#footnote-ref-3)
4. (World Poverty, 2018) [↑](#footnote-ref-4)
5. (Small, 2017) [↑](#footnote-ref-5)
6. (Wang, et al., 2016) [↑](#footnote-ref-6)
7. (Gao, Nuyttens, Lootens, He, & Pieters, 2018) [↑](#footnote-ref-7)
8. (Pujari, Yakkundimath, & Byadgi, 2015) [↑](#footnote-ref-8)
9. (Ferentinos, 2018) [↑](#footnote-ref-9)
10. (Borra & Pukkela, 2018) [↑](#footnote-ref-10)
11. (Fuentes, Yoon, Kim, & Park, 2017) [↑](#footnote-ref-11)
12. (Biswas, Avil, Sarangi, & Pappula, 2016) [↑](#footnote-ref-12)
13. (Maharjan, Takahashi, & Zhang, 2011) [↑](#footnote-ref-13)
14. (Gale group, 2018) [↑](#footnote-ref-14)
15. (Samuel, 1959) [↑](#footnote-ref-15)
16. (HelloACM, 2016) [↑](#footnote-ref-16)
17. (Koehrsen, 2017) [↑](#footnote-ref-17)
18. (Cochran, 2018) [↑](#footnote-ref-18)
19. (Brostoff, 2018) [↑](#footnote-ref-19)
20. (Raschka, 2014) [↑](#footnote-ref-20)
21. (Ray, 2017) [↑](#footnote-ref-21)
22. (Gandhi, 2018) [↑](#footnote-ref-22)
23. (Anaconda, n.d.) [↑](#footnote-ref-23)
24. (spMohanty, 2018) [↑](#footnote-ref-24)