Mälardalen University
School of Innovation Design and Engineering
Eskilstuna, Sweden

Thesis for the Degree of Bachelor in Computer Science with Specialization in Intelligent Systems 15.0 credits

CLASSIFYING BASIL AFTER THEIR DIFFERENT GROWTH STAGES USING MACHINE LEARNING

Patrik Andersson
patand0224@gmail.com

Examiner: Ivan Tomasic
Mälardalen University, Västerås, Sweden

Supervisor(s): Baran Cürüklü
Mälardalen University, Västerås, Sweden

13/06/2023

Abstract

This thesis covers the topic of applying machine learning within the agriculture industry. The goal of this project was to set out with the intent of using computers in trying to classify plants after their different growth stages with the hope of making it easier for the regular person to grow the their own food. The project set out with trying 3 different machine learning algorithms to get and perspective on how they performed for this specific problem, but also to get an overview on how the algorithms performed compared to each other. The result and conclusion showed that the algorithms are currently unfeasible to use, mostly because of the lack of data. Though the result still were promising as it proved the concept and that there is potential for further developments on the work, like the tinkering on the algorithms but mostly of gathering data.

Contents

List of Figures

List of Tables

1. Introduction

The digitalization of the agriculture industry has opened up opportunity for multiple different kinds of machine learning application. Implementation such as trying to detect nutrient deficiency in plants, detect diseases and classification on multiple different types of crops are some of the areas that has been worked on and has shown success [1]. Though one area that has been rarely touched is classification of crops for their different growth stages. Still some research's has been done like a paper written by Tian [2] where they succeeded in trying to implement this kind of system, but otherwise work like this are far in between or it is not the main focus of the projects.

This is where more work has to be put into to get more knowledge within this field of study as this can allow farmers or anyone in general that grows their own food to use artificial intelligence as a tool, just as how someone uses their hand spade tool to plant crops. To explore more of this field the decided method of choice was a supervised learning method, this would allow the supervised learning method to learn from multiple different example of plants with varying growth stages [3].

Using the supervised learning method approach there were three different algorithms chosen from the category to be used, the first clear chosen algorithm was the Convolutional neural network (CNN) as it is common method to use in similar prior work and also because it is considered to be state-of-the-art when working with images. The second method of chose was a Fully connected neural network (FCNN) as it is the predecessor to the CNN method, the method was also chosen as it is a simpler method and also to get an perspective on why the CNN as become the alternative to a FCNN when working with images. The third picked algorithm was the Support vector machine (SVM) algorithm, this was picked as it was used in other similar work.

These three methods were tasked with the goal of classifying images of basil after their different growth stages. The result ended up showing some promising result when tested on different kinds of data, though the method are far from being practically used and more work has to be put into to make the idea viable. Remedies such as testing out different architectures or changing what kinds of features that could be used from the images could be considered, but most importantly collecting more data would be the top priority.

1.1 Problem Formulation

This paper explores using machine learning to predict at what growth stage a plant is currently in by purely looking at it visually. Instead of having an expert that could look at the plant and do test on it to determine how much the plant has matured, an algorithm will be the judge for determining at what stage the plant is currently in.

The idea will be to use a phone or any kinds of camera to take a photo of a plant, the photo will then be used by an algorithms to calculate and predict how old it is solely by using the image and then give an answer to the user, either it could be an absolute answer or an range of different possible answer with a confidence score.

This will allow for a regular person to use the algorithm to get a answer on how much their plant has matured without having to consult with an expert. This can also open up an opportunity for a more automated system to determine what to do with the plant and influence for the desired outcome, which has been discussed in a paper by Barus [4] where a Internet Of Things can be applied to farming. Though for this project it has limited itself to just using a machine learning method to analyze a image of a plant and determine the growth stage, mostly because the thesis is focusing more on the algorithms sides of things rather than Internet of Of Things or robotics.

2. Background

This section explains the concept about artificial neural networks, supervised learning and other terms used within machine learning.

2.1 Supervised Learning

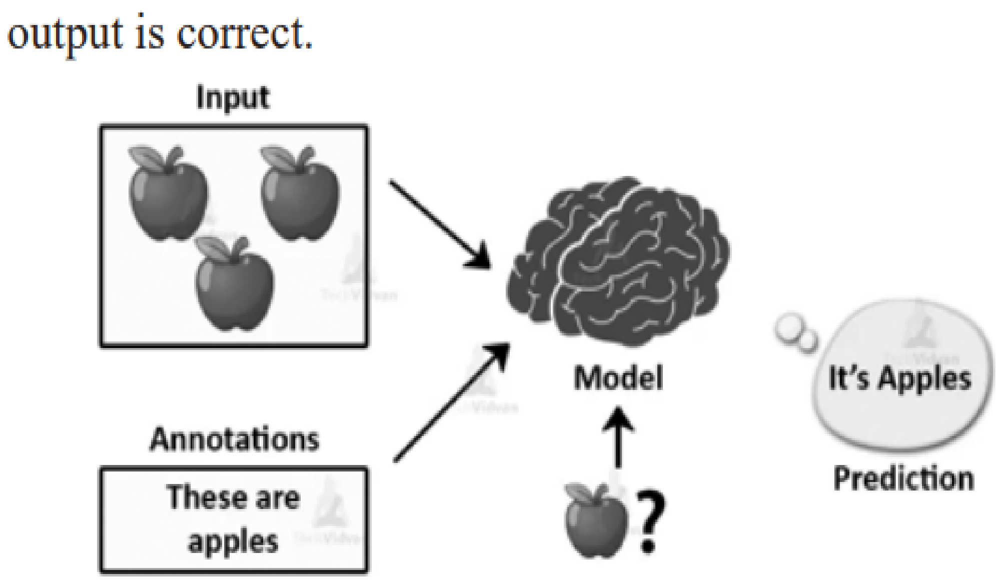

Figure 1: An image of the concept of Supervised learning. The supervise learning model receives an image of apples along with a annotations of what the image contains. The model then determines what kind object the images contains which it proceeds to predict it to be an apple [5].

Supervised learning is one of the three main techniques of machine learning, what differ from these three concept is how they learn on the data. With supervised learning the models trains itself by using data with labels on containing the true answer, this is demonstrated with an example in figure 1 where the model make a prediction of the image for it to then get a response on what the true answer is. The classification of the data can vary, but they can either be binary or multiple different classification [3], [5].

2.2 Fully connected neural networks

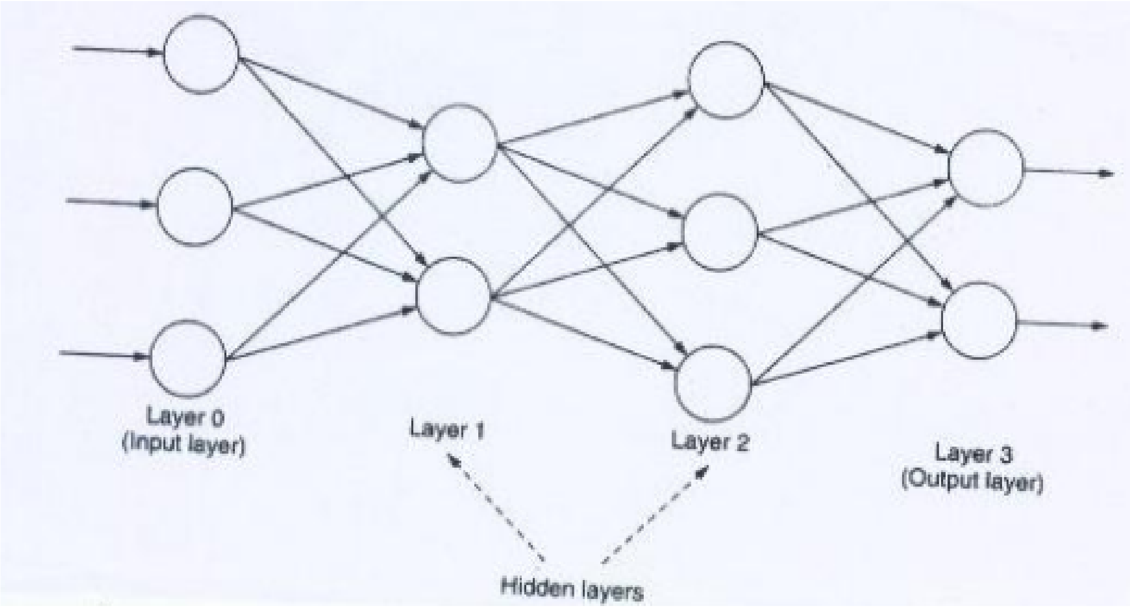

Figure 2: An image of a feed forward neural network, also can be refereed as a FCNN. The image contains the mandatory input and output layer with 3 input neurons and 2 output neurons. The neural network also contains 2 hidden layer with the first layer having 2 neurons and the second hidden layer having 3 neurons [6].

FCNN is an imitation of the theory of a biological neural network in trying make computers work the same as the brain. The brain has multiple neurons that are connected with each other, these neurons are a part of the FCNN and can be represented as nodes. The amount of neurons that are part of the FCNN can vary and are influenced by many factors, such as the amount of layers, the input data and the the output options [6], [7].

There are three types of layers in the FCNN, two of these layers are the input layer and the output layer that are a critical part of the architecture. The input layer are the neurons that receives the raw data that is going to be processed. The output layer is the chunk of the FCNN that gives information extracted from the data, these outputs can either give a simple "yes or no" answer with one output neuron or multiple different classification of the data with multiple neurons. The third layer is the hidden layer which is an optional layer and there can be multiple of these layers compared to the input and output layer, an example of how these layers can be set up is demonstrated in figure 7 [6].

In the hidden and output layer is a so called Activation function that is used to the input to each neuron to give the network more comprehensible values to work with, these activation functions comes in the form of mathematical formulas and some commonly used formulas are simgmoid, tanh and relu [7].

The outputs from each neurons are influenced by a so called weight and bias, these variables are used along with the input value to calculate the weighted sums to give the output value to be passed along in the FCNN [7]. Each neuron has a weight assigned to it, these weights are at the beginning randomized within a interval, usually from 0 to 1 or -1 to 1, same with the bias but with and exception of having one bias per layer instead of neuron. These weights and bias are then optimized over the training period for the specific task, this is done by using Back propagation [6].

Back propagation is the most common method of choice when updating the weights and bias, this is where the labeled data gives a respond to the relative error to what the true answer was to the respond the FCNN gave. This error will overtime be reduced by changing all of the weights and bias by using the gradient descent algorithm [6], [7].

2.3 Convolutional Neural Network

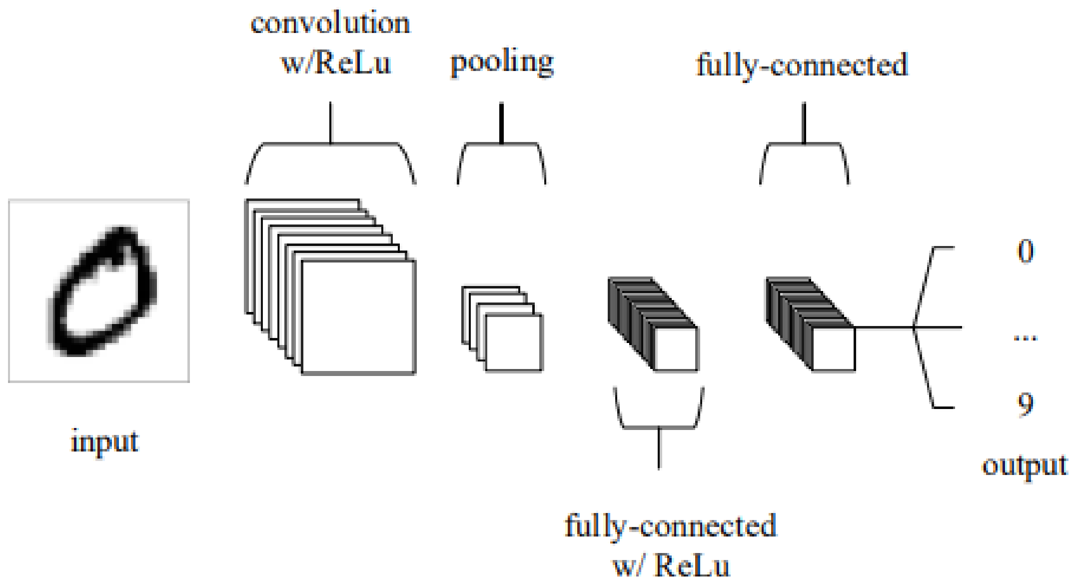

Figure 3: An image of a CNN. The CNN job is to classify a data set of whole numbers ranging from 0 to 9. In the illustration of the image the network receives a number 0, this image then goes through a convolutional layer with relu as its activation function. In the second layer the image gets reduced with a polling layer, then in the last layers the image goes through a fully connected neural network which classifies the image into one of the possible numbers [8].

CNN emerged as a alternative method to use instead of FCNN when it comes to working with images. This was mostly because of the hefty amount of neuron required to compute large resolution images, just a 1280 X 720 RGB image requires almost 3 million neurons and this can drastically increase when adding hidden layers to the FCNN or increasing the size of the image. This is where CNN shines as it can detect patterns in the images and save the most important information from the pattern extraction for the fully connected neural network [8], [9].

CNN has multiple components to the architecture, the ground pillar to the structure is the convolutional layer, this layer is used to extract the feature from the image by using kernels. These kernels comes in the form of matrix's and are smaller than the image and usually multiple of these are used in the layer. These kernels are first placed at the top left corner of the image and multiplied with each other and summed together to get the extracted value. This kernel will keep on doing this process starting from the top left of the image all the way down to the bottom right. There are multiple factors of how this kernel behaves, like the kernels size and the set Stride, which means how big of a step it takes [8], [9].

Another important part of the CNN is the poling layer which will remove unnecessary values from the matrix's, these polling layers acts similar to the convolutional layer when it comes to the size and stride. Usually these polling layers only keep the largest value from the selected values. The size of these polling layers are commonly set to 2 X 2 as this is a good middle point of keeping important information but also removing unnecessary values [8], [9].

At the end of the CNN is an fully connected neural networks which will then input the extracted features and then give a output to classify the image [8], [9]. All of these steps and components of the CNN are visually demonstrated in figure 8.

2.4 Support Vector Machines

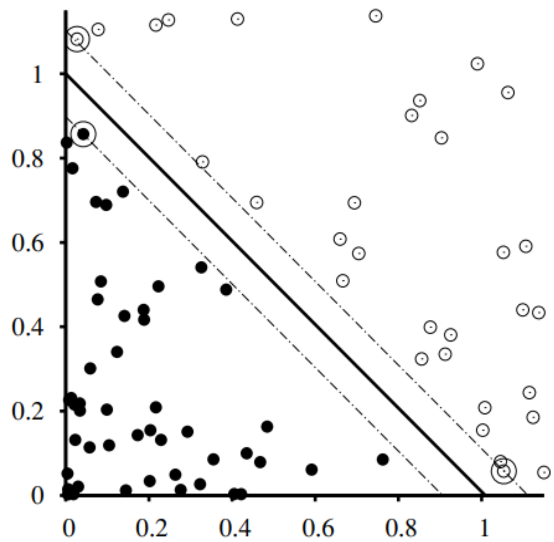

Figure 4: A figure of a support vector machine. The graphs is a two dimensional and has two different classes distinguished with white and black dots. The dimension represent two numerical features about the two classes with values ranging from 0 to 1. In the middle of the graphs contains a separator optimally place with the maximum margin placed between the two classes [3].

SVM was the more popular classification method to use within machine learning in the early 2000, but in recent years artificial neural networks and random forest has become more common of choices when working with classification, but SVM still has it valuable uses for its simplicity and being able to handle smaller quantity of data [3].

The idea behind SVM is try to group up two or more different classes after their different features, the amount of used features can depend but the point of it is to use as many features as possible to be able to distinguishes different groups of classes with taking in limitations such as computation. The features from the data are then plotted out on a graph with each dimension representing a features, these clusters are then to be grouped and separated by using one or more maximum margin separator. To find the optimal placement of the boundary the placement is calculated by using a mathematical formula to maximise the distant between the closest point of the classes [3]. In figure 4 an example is giving of a SVM trying to classify data into 2 classifying with just using 2 features.

2.5 Machine Learning Concepts

Data is one of the core aspects of machine learning and is used for the computer to analyse and learn from the data without having any specific instruction on how to interpret it [10]. Though the data is used for learning the computer to complete certain task, the data has multiple different purposes and are divided into different groups. The data is commonly divided into three groups called training, validation and test data. The training data is used for the purpose of learning the model of the underlying pattern that makes up of the data set. Validation data is there to prevent overfitting on the training data as there is the possibility the model becomes too specialised on the specific training data instead of having a more generalised interpretation on the data. The validation data is used by instead of letting the computer learn using the training data, the validation data is used to learn the computer every so often. The test data is used for evaluating the model on how it would perform on other sort of data that the computer has not been trained on, this is used to give an perspective on how good the computer would perform in practical scenarios [11].

Data augmentation is a common method to use when data is limited, the idea is to use preexisting data and make small alteration to create new data, this allows the data still to be the same kind of data but different enough for models to train on it. Types of data augmentation on images can be changing brightness, cropping the images or scaling them [3].

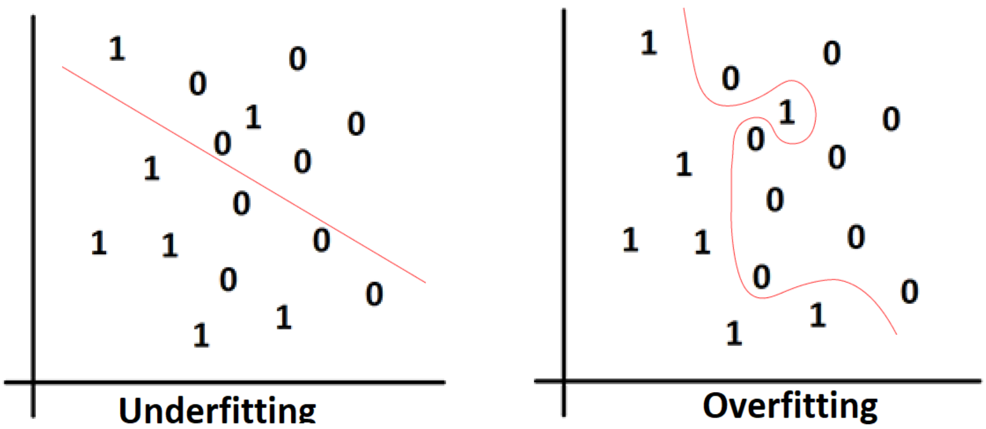

Figure 5: Two graphs with two data types plotted on them represented with ones and zeros. The example on the left has not been trained enough to have a good enough generalisation on the data while the example on the right has lost the generalisation, the class one around all the zeros is a typical example of overfitting.

Overfitting is a common problem within machine learning that causes the algorithm to perform good on the training data but not on the training data, this is opposed to underfitting as this causes the algorithm no to adapt to a generalising to the data nor unable to interpret the data at all. There are several causes to this problem, one of these is that not enough diverse data is used during the training phases which causes the algorithm to get a narrow perspective on what the data can look like, this can be solved by stopping the training phase before it over trains on the training data [12] [13]. An example of overfitting is shown in figure 5 were the algorithm has become way adapted to the data.

Normalization within machine learning is the concept of prepossessing data for it to be adapted for machine learning algorithms. As raw data can vary and have different formats it is important to reshape it to be cohesive with each other and be fitted for the algorithm. This is where normalization is usefully method of chose when reprocessing the data into the range of zero to one or minus one to one which are the common scalable ranges when normalizing the data [14]. There are multiple different method to use when normalizing data, one of these are the Min-Max method where the data is usually scaled down from 0 to 1 [15]. The formula is as follows:

$$X_{\text{norm}} = \frac{X_{\text{old}} - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

Where:
X - numerical value.
old - the original numerical value.
min - the lowest desired normalized value.
max - the highest possible normalized value.

Batch size is one of the parameters in machine learning that affects the training phase, as it influences the training time and the end result on how the algorithm perform both on the training data and the test data. Batch size is defined as the concept of determining the amount of samples used for each training cycles or epochs before updating the algorithms parameters. The most common batch sizes is set from 16, 32 and so on all the way up to 512. The smaller the batch size is the less computation is required for each training iteration but the algorithm risk not being adapted to generalise as less examples are given for each epoch [16].

Optimizers are one of the key influencer in changing artfical neural networks parameters during the learning process, these optimizers use learning rate as an factor in determine on much the algorithms parameters will change for each iteration. Common learning rates values are set to 0.01 or 0.001 or anything in between and are adjusted over the training of the algorithm as it begins to converge. This is where the algorithm can use different kinds of optimizers as a method in influencing the learning rate differently over the training period [17] [18].

Gradient descent within machine learning is an algorithm that is used to optimise algorithms, like a neural networks. As it sounds in the name the variables of the algorithm that is trying to be optimised is changed by calculating the gradients of all these variables to get a low of a loss as possible [19]:

$$\theta = \theta - \alpha \cdot \nabla J(\theta) \tag{2}$$

Where:
θ - the vector.
α - the learning rate.
∇ - the gradient calculation.

3. Related Work

This section covers some of the similar work that has been worked on with analysing plants growth stages, these works is used for reference and inspiration for this thesis.

3.1 Related 1

The research within this fields is small and has not been written a whole lot of papers on, though there still are some similar work out there like where a couple of researchers used the YOLO-V3 algorithm to not only detect apples but also to classify them after different growth stages. The paper defined three different growth stages for the apples, the first is called "young apples", the second is referred to as "expanding apples" and the final stages is called "ripe apples". The algorithm was trained on 4800 images of which among the majority was created with data augmentation. The result ended up showing that the precision and recall score was about 80% [2].

The research has a lot of similarities to this thesis work as the main focus of the paper is to use an algorithm to classify and detect crops, though the paper takes it a step further to try to also detect apples. This is why they decided to use the YOLO algorithm as this is considered to be the state-of-the-art for detection algorithms [2]. But for this thesis the YOLO algorithm is not used as it mainly focus on classification instead of detection and is beyond the scope of this project. Though YOLO uses a CNN as part of the architecture and CNN is considered to be the state-of-the-art within fruit classification [2] which is used as part of the work, but instead of fruit classification it will be growth stage classification. The paper also uses the technique of data augmentation to increase the data set, one of these techniques was rotating the images in multiple different ways, this option was considered as this thesis lacks access to quantity of data

3.2 Related 2

The second paper is about classification of different growth stages of wheat and barley using a CNN and a Support vector machine. The authors claims that they were the first to use CNN for classification of crop growth stages. In the report they made, three different models were made to test out their data set of 138000 images, the first and second model is both a CNN, but with a different level of complexity to their architecture. The third model is a Support vector machine. All of these models were supposed to classify the images after the wheat and barley different growth stages, twelve classification for wheat and eleven classification for barley. The best performing of the three models is the one with the most complex CNN architecture with a almost 100% accuracy, in the second place comes the CNN model with the more simpler architecture with a accuracy of around 90% and last comes the model using Support vector machines with a accuracy of about 60% [20].

If the claims from the paper of being the first to use CNN for classifying plants different growth stages then this paper is valuable for the thesis as it gives valuable information into how CNN were applied to this problem. One thing that the paper gives important information about is of how the architecture of a CNN could look like as it gives examples of two different models, one with a simpler model and one with a more complex one. This means that the CNN implementation of this paper definitely had a inspiration for this thesis CNN architecture. The paper also uses support vector machines as one of the models and was considered using part of this thesis, though the model might be lackluster considered how it performed compared to the other CNN models. Lastly this paper proves that algorithms can classify plants after their growth stages with near perfection which proves machine learning can be applied within this field of study.

4. Method

As there has been a lot of research done with trying to apply artificial intelligence to the farming industry, a lot of different approaches has been used to try to solve multiple kinds of problem within this field. Though to converge down to a suitable method for the specific problem a supervised learning technique is used as labeled data is available for teaching the algorithms.

To get a perspective on the result, three different methods have been selected to be compared between each other. The first chosen method is a CNN as it is a common method of choice when classifying images within this field [21]. The second method of choice is a FCNN, even if nowadays CNN are considered a preferred choose when working with images it has couple of advantages, the first is that FCNN is a simpler method to implement and the second reason is that it requires less data for it to get a good performance [22]. This could be a big advantage for the use of a FCNN as access to large quantity of data is a issue for this project and FCNN has also been used as a method of choice for classification of plants in other similar works [23]. The third method of choose is the SVM algorithm which used to be an older technique to use for classification before ANN and CNN became state-of-the-art. Though it still has its uses till this day and has been used in similar work with plant growth stage classification as has been discussed in section 3.2. This method is also picked to give a perspective on why ANN has become the standard choice of method compared to the SVM algorithm.

4.1 Methodology

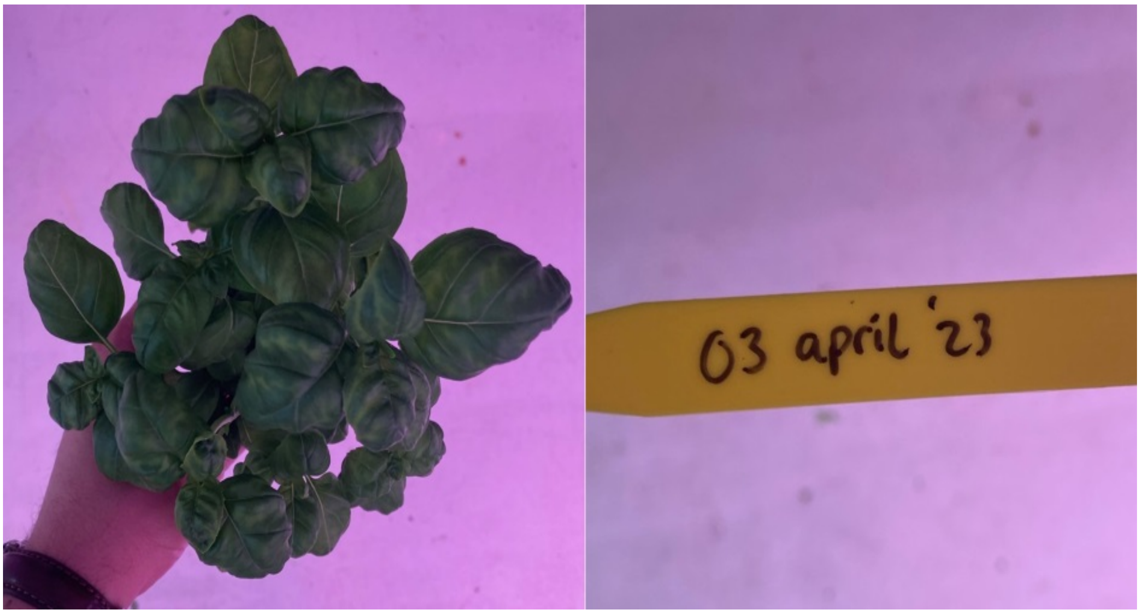

Figure 6: An example image from the data set, on the left is a image of a basil plant which is used by the algorithms to train. On the right is an image of an date, this date is directly correlated with the plant on the left as it tells certain information about the plants growth stage.

The data used for this project comes in the form of images with a resolution of 920 X 1280 and the images are in JPG format. These photos has been taken at the street of Gjörwellsgatan 30 in the city of Stockholm using an IPhone SE (2020). The images has been labeled according to what part of their growth stages they are currently in, this is demonstrated in figure 6 where their is a image of plant along with a another image of a pin representing the date label, there are four types of pins with the following dates, "03 april 2023", "10 april 2023", "31 march 2023" and the "07 april 2023". The labeling of the images is important for the work as they are used for training and testing to determine if the algorithm is right about its prediction.

The main choice of tool is Python for this project as it is a common tool to work within machine learning and includes pre-made libraries, this allows Python users to work with a lot of different tools and concepts without fully understanding them or to program everything from the grounds up. This is where the programming language is used to implement the three different methods with the use of python libraries, for the ANN and CNN the Tensorflow library is used or more specifically the Keras library as it is used to build neural networks and it belongs to the Tensorflow library collection. Beside using Tensorflow the Scikit library is used as this is another Machine learning tool and is used for building the SVM algorithm. These are the main tool of use for building the algorithms and will be used along the Python Imaging Library as this tool is used for working with images. This tool is used for the purpose of analysing the images to give an perspective on how the image is structured.

FCNN model

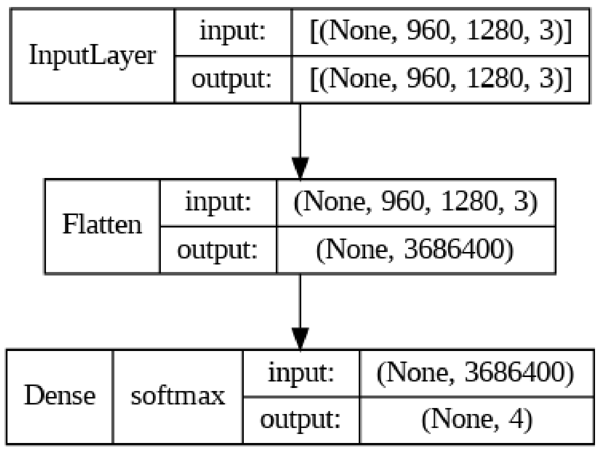

Figure 7: An image of the final version of the ANN model. At the beginning of the model the first input layer makes it so the model can only receive a 960 X 1280 X 3 matrix. In the second layer the matrix is then turned into a vector with a length of 36,846,400. This is then processed through a Dense layer that predict 4 different categories using the softmax activation function.

The final version of the FCNN is a simple model were it only uses one input layer with 36,860,400 neurons and one output layer with 4 classification neurons representing all the different types of date classification to the plant as can be seen in figure 7. Adding hidden layers and dropout layers seem to have caused a lower accuracy on the testing data which ended up with the final version having no extra layers.

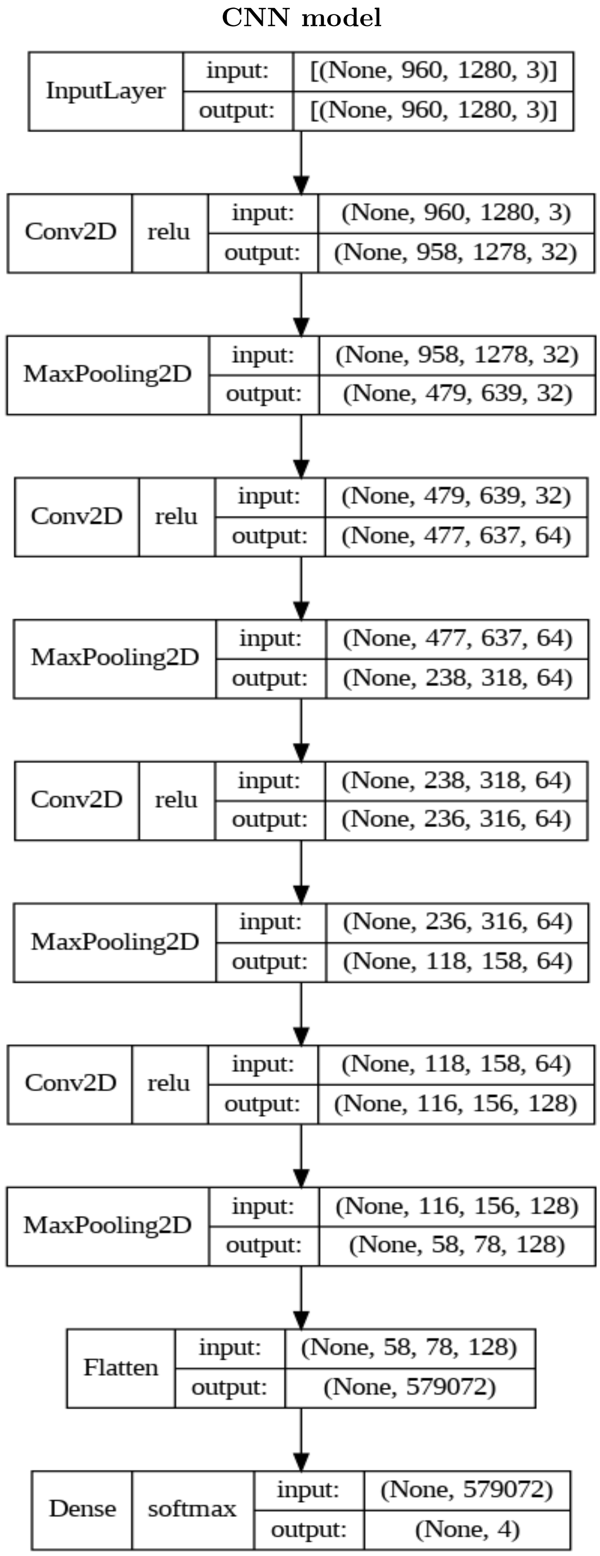

Figure 8: An image of the final CNN model. At the top of the figure is an input layer allowing the algorithm to only receive a matrix with the size of 960 X 1280 X 3. This matrix then goes through the first convolutional layer of where features are extracted with relu as its activation function and outputs a new matrix of 958 X 1278 X 32. The matrix is then reduced down using a maxpolling layer. This process goes through another 4 maxpolling and convolutional layers until it reaches the last 2 layers, the second last layer turns the matrix into a vector and then sends it into a output layers which it has 4 outputs. The output layer then classifies the image into one of the date categories.

The final CNN model has three maxpolling layers and three convolutional layers in order according to figure 8 with a kernel size set to 3 X 3 with a stride of 1 for all the convolutional layers and a kernel size of 2 X 2 with a stride of 1 for the maxpolling layers. This model is heavily based on the work by Sanaz which is discussed in section 3.2 where they worked on a similar project, this gave a good point of reference to use for the groundwork and adapt it for this specific problem. The model ended up with a similar architecture with having 32 filters in the first convolutional layer and then 64 filters in the second and third. Though the work by Sanaz they just had three convolutional layers, but this version their is one more convolutional layer added with 128 filters as this seem to ended up with the model performing better. At the end of the model there is also an output layer that classifies the plant into the 4 different classes. In the model that the CNN model is based on there is also an hidden layer with 1024 neurons, though for this version it was removed as it did not seem to improve the overall accuracy but also because of memory limitation.

The final version of the SVM did not have a whole lot of modification to it compared to the other models as there are less factors to influence when it comes to SVM, though one thing that could be changed is what kinds of features to be used. As for this project the decided upon features to be used was the pixels values as this can give more information for the SVM to work than just using like the ratio of the red, green and blue values of the images, it is an easy features to extract from the images and doesn't require that much computation.

All of these three models was build and trained in the google colab environment using the GPU V100 with having accsess to 16 GB of GPU RAM.

The evaluation of the algorithms is done with calculating the accuracy of each model by using the test data to get the result. The accuracy is one of the common choice of metrics for evaluation within machine learning and the mathematical formula is formulated as follows [24]:

$$\text{Accuracy} = \frac{\text{The amount of right predictions made}}{\text{The amount of predictions made}} \tag{3}$$

The evaluation is done with two test for each model. The first test is done by training all the model on about half the training data, with all the models using 16 images for training. The second test will be done by using all of the training data containing 28 images for training the CNN and FCNN model and 32 images for training the SVM model. The purpose of these test is to see if the models perform better if more training data is used.

5. Ethical and Societal Considerations

With the limitation of this work it is unlikely that it will have any effect on society or how the agriculture industry will function. As for the ethical regards it is important to present the result and conclusion as clear as possible so that the information can not be misunderstood and give the most important findings too other researchers within this field of study.

6. Methods implementation guide

This section explains how the work was carried out and each major step from the project.

6.1 Data collection

Before any practical implementation could be done, there had to be an understanding on what kind material would be used for the project, in this case it was the data of images containing plants with different growth stages.

As the methods used needed to be adapted to the structure of the images, the first step was to get a hold on this data to know exactly whats gonna be worked on, like important information such as how many images is gonna be used, the resolution of the images, the image type and how many classes there are of the data set. These images was then proceeded to be collected at a vertical farm in the city of Stockholm and the photos was took with a IPhone from a top down angle. The camera took photos of four types of basil which were differentiated from at what part of the vertical farm they were planted in.

From each type of basil ten photos was taken along a photo of a pin containing a date on it, see figure 6 for an example pin. This ended up with 44 images in total, 40 of which are data and 4 which are the metadata of the four types of basil classes. All these images became an important part for implementing the methods and evaluating them.

6.2 Data analysing and labeling

Before any of the images could be used, an data analyzing was done to get a perspective on how the data is structured. Secondly the data had to be categorized, also know within machine learning as labeling. The image analyzing was done with Python using a library called Python Image Library, using this gave important information like the resolution of the images which was 960 X 1280 and the amount of color channel which was red, green and blue. From this the amount of input neurons required for the ANN methods and the amount of features for the SVM could be calculated, this gave the result of 3,686,400 pixels containing the image.

With the image analyzing done the data had to be labeled, this was done by dividing up the data into four categories according to which pin they belonged to. As three different methods was used, data had to be labeled according to what type method was used, for the FCNN and CNN the images was labeled by putting them into a folder then dividing them up into four different sub folders categorized after the dates. Then the data was further split into three categories, with 28 (70%) images going into the training data, 4 (10%) images going into the validation data and 8 images (20%) going into the test data. Next the data had to be labeled for the SVM method by renaming the image file name after one of the four categories the image belonged too, for example the image file could be named "03_april_2023.jpg". After the labeling the images for the SVM was split into two categories with 32 (80%) images going into training data and 8 (20%) images becoming test data.

With a good understanding of the data and the data being processed the data became ready for usage. To see more of the data see section 10..

6.3 Applying methods

The next step of the plan was to create the models using Python as the main tool. This was done using google colab environment as it is a good tool to use to get a hold on the right amount of computation hardware. The data was loaded up from the google colab directory by using the "image_dataset_from_directory" function from the Keras library into three different variables, "training_data", "validation_data" and "test_data". Using this function the data was also normalized by scaling down the pixels values from 0 to 255 down to 0 to 1 and sorting the images into a batch size of 5, this made the data ready for the FCNN and CNN method.

For the SVM method the data was loaded from the directory from where each image was divided up into a main list, the pixels from each image was then further divided into a sub list within the main list. At the same time the pixels was also normalized down to ranges from 0 to 1. Along with putting the images into a list the label from each image was also put into a separate list with the image and label having the same index in different lists.

With the data ready for input the models was loaded up using two different libraries, for the FCNN and CNN the Keras library was used and Sklearn was used for the SVM model. As there were no real prepared model for the FCNN and CNN the models had to be build from scratch, this was mostly done with trials and error with testing different kinds of architecture by adding and removing different kinds of layers. These kinds of architectures was then trained with 50 epochs and tested to see which one performed better, the architecture that performed the best was then saved for evaluation with the best architectures usually having test score ranging from 50% all the way up to 75%. To see the final architectures see figure 7 for the FCNN model and figure 8 for the CNN model.

As for the SVM model the "SVC" function was used where it was provided the images pixel values as features and the image name as labels. See section 10. for more information about the programming.

6.4 Evaluation

Lastly after the optimisation and training, the models were ready for evaluating. To determine how each model perform the accuracy would be calculated by using the test data. For the evaluation there were two types of test they would do to evaluate their performance, one where they trained them self on about half of the training data and one where they trained on all of the training data. For the evaluation of the CNN and the FCNN model they used the "evaluate" function from the Keras library to get the accuracy and as for the SVM model the "score" function was used from the Sklearn library to get the accuracy. With the evaluation of the three methods completed the result could be finalised, see section 7. for the result.

7. Results

FCNN model training accuracy

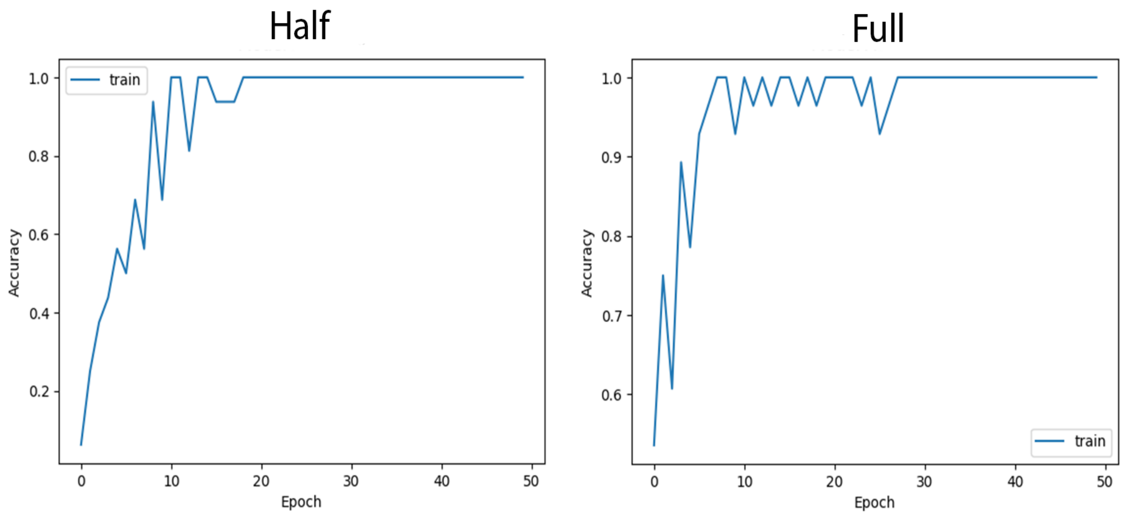

Figure 9: Two graphs of the training accuracy during the training for the FCNN model. The graph on the left represents the training when using half of the training data while the graph on the right shows when the model uses all of the training data. The two graphs represent the training accuracy over time during the training process with the Y-axis being the accuracy and the x-axis being the epochs. The greatest accuracy increase happens at the first epochs and the models hits max accuracy around 10 epochs, though the graphs accuracy on the right increases faster at the beginning compared to the one on the left.

CNN model training accuracy

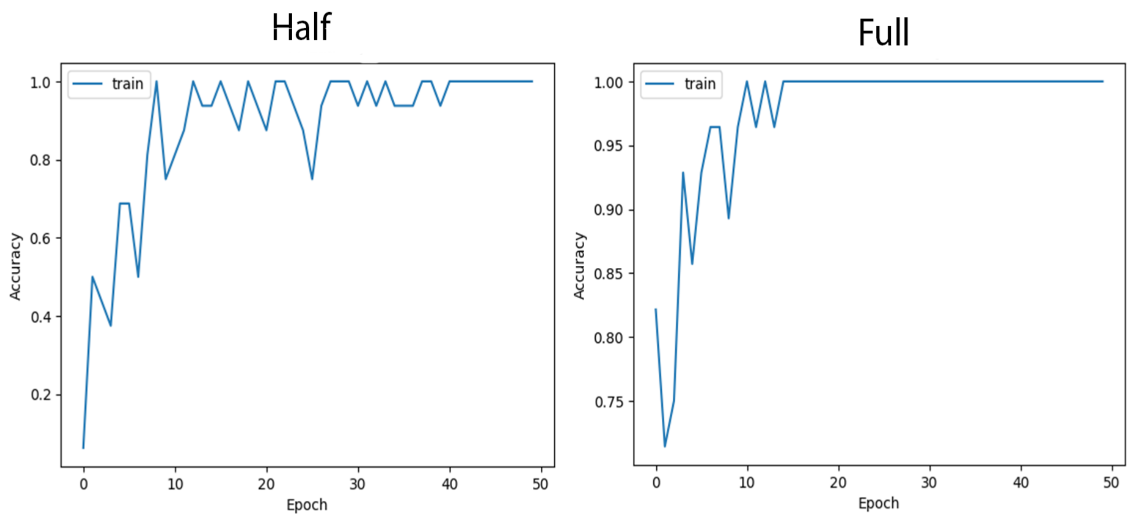

Figure 10: Two graphs of the training accuracy during the training for the CNN model. See more of the figure description in figure 9 description.

In figure 9 and 10 contains the training process for the FCNN model and the CNN model, all of which had the purpose of learning from the training data they had at hand to then apply their knowledge on classifying test data. The models learn quickly as their are not many examples to learn from as can be seen in the graphs of were the models training accuracy increases fast. The models still have some accuracy drops which can be explained with the models occasionally using the validation data instead of the training data, this causes the accuracy to drop as it trains on unfamiliar data.

	SVM	FCNN	CNN
Half	0.75	0.375	0.25
Full	0.75	0.5	0.375

Table 1: The result of the evaluation of each model. At the top row is all the three models implemented in this project, the columns to the far left is displaying how much data of the training data was used to train the models either with half the data or all the data. This table shows how much accuracy it gets from each model along with how much data the model uses. For example the SVM using half the training data got a score of 0.75 or in other words an accuracy classification of 75%.

The result is from an evaluation of three different models of its purpose is to classify images of basil into four growth stages. The three models was tested on a image data of 8 and the models could get a accuracy score ranging from 0 to 1. In table 1 the result can be viewed of the performance of each model with different amount of training data.

For the SVM model it performed the same on the different amount of training data with a 75% accuracy and is the model that performed the best out of all the models, this would mean that the model got 6 out of 8 images classification correct. The second best performance of the models is the FCNN one with a accuracy of 37,5% of half the amount of training data and a 50% accuracy on using all of the training data. The worst performing model is the CNN model with a 25% accuracy on half the data and 37.5% accuracy with all the training data, With a accuracy of 25% the model does not perform any better than compared to blind guessing or untrained models as the models on average are expected to get 25% of the guesses correct with a option of 4 choices.

In table 1 there seem to show a trend with an exception of the SVM model that the more training data the model uses the more accuracy it gets with both the FCNN and the CNN model having a higher accuracy with more training data.

8. Discussion

The results could be considered surprising as that FCNN and CNN are viewed to be more of the state-of-the-arts methods with image classification, still the SVM performed way better than compared to the ANN methods as shown in table 1. Though the result might not be as surprising bearing in mind that SVM works well when it uses high dimension data, especially when the number of samples is lower than the the amount of dimensions. Secondly, large data sets makes SVM perform generally bad and when the data set is only 40 images it makes sense why SVM performed good [25]. This is to the contrarily bad for FCNN and even more so for CNN as it is generally accepted that these algorithms need large and diverse data to perform good for their specific task. Even if the FCNN and CNN performed disappointingly which can be partially explained with the small data set, there is still a big gap between the neural networks models and the SVM models.

The final score that the FCNN and CNN got is a bit unexpected when comparing to the score of experimenting with different kinds of architecture, Some of the models even got as high of a score of almost 90% when testing the models. Compared to the result that is a big difference and is unlikely to have happened by direct chance, rather it is likely to have been caused the models being overfitted on the training data when training for the evaluation making the models lose the generalisation and being to specialised on the training data. This reasoning could be explained in figure 9 where both of the graphs do get max accuracy around half way through the training, after that the training might be doing more harm than good as the models begins to over train on the training data causing the models to perform bad on the test data. This is where multiple methods could have been used to try to limit this problem, one is to get more data but as the limitation of the project that was out of the question. The other methods could have been to use dropout layers, but as explained in section 4.1 it was not included. The last thinkable methods was to just cut the training earlier to prevent the models from over training, maybe around 25 epochs. But as mentioned before with the limitation of the project it is hard to test all kinds of factors with neural networks as there so many of them.

As these facts considered the outcome of the evaluation is to be expected with the SVM method performing the best and the CNN method doing worse out of all the methods. However this does not mean that SVM are the better method, it only proved that it performed good on a small data set but has not proven yet on how it would perform on a large and diverse data set of basil along with the other methods. As acquiring data is one of the major challenges within machine learning it is hard too test this though as a lot resources has to be invested into the collection of data. Though simpler methods could have been used to expand the data set like using data augmentation as was discussed in section 3.1 where they used data augmentation on images of apples and rotated the them.

With the discussion about the data set being mostly about lacking quantity of data, there also exist some discussion about the quality of the images. In figure 6 contains the image of the basil with a lot of important features about it, but the background of the image also takes up a lot of the space which causes it to use a lot of information that it should not use. For one example is that there is a lot of purple in the background and this is common trait in the rest of the data set. This is not optimal as the algorithms are supposed to be flexible on any kind of image with a basil in it and it is likely that the algorithm would perform even worse if it was tested on basil image with other kinds of backgrounds. There are several solutions that can be applied to this problem, one solution is just to get a larger wider range of different kinds of images with different kinds of environment. The second method is to try integrating some kind of region of interest into the algorithm so the SVM and neural networks algorithm only focus on the relevant information.

Lastly, considering the limitation of time and resources there have shown some success with the research done. The goal was not necessarily to get high accuracy from the models but rather just prove the concepts or lay the groundwork for the potential of future work as there is a lot more research that could be done with this topic. The models have shown some capability of classifying basil images and have gotten a better result when using more data. Though for using the three different methods for practical purposes is right now out of the question, for reason that the methods can only handle 4 types of growth stages and just the general inflexibility with a small data set. Until the three models have been trained and tested on more different examples, there cant be a "yes or no" answer yet on if the modules can fulfill the task of categorizing different plants after their growth stages.

9. Conclusions

This thesis set out with the goal of trying to apply artificial intelligence within the agriculture industry. The project converged down to a goal in trying to apply algorithms that could try and classify images of basil after 4 different growth stages, this was done for the purpose of democratization of the farming industry by allowing people with little to no knowledge to more easily start farming their own plants.

To approach this problem 3 different supervised learning methods was applied with the intent to get an overview on what algorithm would be suitable to use for the problem, these methods used were a fully connected neural network, a convolutional neural network and a support vector machine. The evaluation of the 3 models showed that the support vector machine performed the best with the convolutional neural network doing worst with a close in to the fully connected neural network. The reason for why the methods performed they way they did was because the amount of data used with a data set of 40 images, with Support vector machine being feasible for a small data set while neural networks needs large amount of data to train on to function optimally. The methods was determined to to be currently unsuitable for practical use, mostly because of the lack of data used. But the evaluation showed some promising result with the artificial neural network methods performing better when using more data, this showing was concluded that one of the biggest improvements the models needed was to train on more data to get a better result.

Lastly the project proved that computers have some capability of classifying plants after their different growth stages, though if it can compete with humans has yet to be proven and more research needs to be done to answer that question.

10. Appendix

The code implementation of the methods:
"https://colab.research.google.com/drive/10LJwOJAgcT6-c2bpCse2A-3dja2AGRuw?usp=sharing".

The data for the CNN and FCNN methods:
"https://drive.google.com/file/d/15EcsqPdmkjClP2ABRSnOFqUE_LL53Zx6/view?usp=sharing".

The data for the SVM model:
"https://drive.google.com/file/d/1KOglFk0QQyvHpDHC2a_y0wroOAzgYsGf/view?usp=sharing".

References

[1] J. G. A. Barbedo, 'Detection of nutrition deficiencies in plants using proximal images and machine learning: A review,' *Computers and Electronics in Agriculture*, vol. 162, pp. 482–492, 2019, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2019.04.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169918318957.

[2] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li and Z. Liang, 'Apple detection during different growth stages in orchards using the improved yolo-v3 model,' *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2019.01.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016816991831528X.

[3] S. J. (J. Russell, *Artificial intelligence : a modern approach*, eng, Fourth edition global edition. Harlow: Pearson Education Limited, 2022, ISBN: 1292401133.

[4] E. S. Barus, M. Zarlis, Z. Nasution and Sutarman, 'Forcasting plant growth using neural network time series,' in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, 2019, pp. 1–4. DOI: 10.1109/ICoSNIKOM48755.2019.9111503.

[5] V. Gupta, V. K. Mishra, P. Singhal and A. Kumar, 'An overview of supervised machine learning algorithm,' in *2022 11th International Conference on System Modeling Advancement in Research Trends (SMART)*, 2022, pp. 87–92. DOI: 10.1109/SMART55829.2022.10047618.

[6] A. Dongare, R. Kharde, A. D. Kachare *et al.*, 'Introduction to artificial neural network,' *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012.

[7] K. Shiruru, 'An introduction to artificial neural network,' *International Journal of Advance Research and Innovative Ideas in Education*, vol. 1, pp. 27–30, Sep. 2016.

[8] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: 1511.08458 [cs.NE].

[9] S. Albawi, T. A. Mohammed and S. Al-Zawi, 'Understanding of a convolutional neural network,' in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[10] C. Dictonary. 'Machine learning.' (), [Online]. Available: https://dictionary.cambridge.org/dictionary/english/machine-learning. (accessed: 02.05.2023).

[11] D. S. Wizards. 'A guide to data splitting in machine learning.' (), [Online]. Available: https://medium.com/@datasciencewizards/a-guide-to-data-splitting-in-machine-learning-49a959c95fa1. (accessed: 02.05.2023).

[12] T. Dietterich, 'Overfitting and undercomputing in machine learning,' *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[13] T. Dietterich, 'Overfitting and undercomputing in machine learning,' *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[14] P. J. M. Ali, R. H. Faraj, E. Koya, P. J. M. Ali and R. H. Faraj, 'Data normalization and standardization: A technical report,' *Mach Learn Tech Rep*, vol. 1, no. 1, pp. 1–6, 2014.

[15] J.-M. Jo, 'Effectiveness of normalization pre-processing of big data to the machine learning performance,' *The Journal of the Korea institute of electronic communication sciences*, vol. 14, no. 3, pp. 547–552, 2019.

[16] P. M. Radiuk, 'Impact of training set batch size on the performance of convolutional neural networks for diverse datasets,' *Information Technology and Management Science*, vol. 20, no. 1, pp. 20–24, 2017.

[17] R. Pramoditha. 'How to choose the optimal learning rate for neural networks.' (), [Online]. Available: https://towardsdatascience.com/how-to-choose-the-optimal-learning-rate-for-neural-networks-362111c5c783. (accessed: 08.05.2023).

[18] T. D. S. S. Ramesh. 'A guide to an efficient way to build neural network architectures- part i: Hyper-parameter selection and tuning for dense networks using hyperas on fashion-mnist.' (), [Online]. Available: https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b. (accessed: 08.05.2023).

[19] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017. arXiv: 1609.04747 [cs.LG].

[20] R. Sanaz, B. C. J, S. G. C. M, H. N. M, L. David and O. G. M. P, 'Crop growth stage estimation prior to canopy closure using deep learning algorithms,' *Neural Computing and Applications*, vol. 33, pp. 1733–1743, 2021, ISSN: 1433-3058. DOI: ttps://doi.org/10.1007/s00521-020-05064-6. [Online]. Available: https://link-springer-com.ep.bib.mdh.se/article/10.1007/s00521-020-05064-6.

[21] J. G. A. Barbedo, 'Detection of nutrition deficiencies in plants using proximal images and machine learning: A review,' *Computers and Electronics in Agriculture*, vol. 162, pp. 482–492, 2019, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2019.04.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169918318957.

[22] V. Meel. 'Ann and cnn: Analyzing differences and similarities.' (), [Online]. Available: https://viso.ai/deep-learning/ann-and-cnn-analyzing-differences-and-similarities/. (accessed: 25.04.2023).

[23] J. Garcia and A. Barbedo, 'Digital image processing techniques for detecting, quantifying and classifying plant diseases,' *SpringerPlus*, vol. 2, no. 1, pp. 1–12, 2013.

[24] A. Mishra. 'Metrics to evaluate your machine learning algorithm.' (), [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. (accessed: 01.05.2023).

[25] Editorial. 'Pros and cons of support vector machine (svm).' (), [Online]. Available: https://roboticsbiz.com/pros-and-cons-of-support-vector-machine-svm/. (accessed: 15.05.2023).

[26] [Online]. Available: https://www.youtube.com/watch?v=r7l0Rq9E8MY.