Object Recognition for Agricultural Applications Using Deep Convolutional Neural Networks

---

Dor Oppenheim

March 8, 2018
Version: My First Draft

Ben-Gurion University of the Negev
Faculty of Engineering Sciences

Department of Industrial Engineering and Management

Object Recognition for Agricultural Applications Using Deep Convolutional Neural Networks

Dor Oppenheim

Supervisors Guy Shani and Yael Edan

March 8, 2018

Dor Oppenheim

Object Recognition for Agricultural Applications Using Deep Convolutional Neural Networks

March 8, 2018

Supervisors: Guy Shani and Yael Edan

Ben-Gurion University of the Negev

Faculty of Engineering Sciences

Department of Industrial Engineering and Management

Marcus Family Campus, Ben-Gurion University of the Negev, Israel

P.O.B. 653 8410501 Be'er Sheva

Abstract

This thesis focuses on the problem of recognizing objects in agricultural environments through computer vision. This problem has been studied for many decades. However, few commercialized agricultural applications use computer vision. A major hindrance to commercialization is the relatively low levels of accuracy caused by the unstructured, highly variable and complex crop environment. Hence, in many parts of the world there is low adoption of agricultural applications using computer vision.

Promising advancements in deep neural network algorithms have shown impressive performance in solving many problems of image recognition in many fields. However, in agriculture deep learning research has been more scarce. Therefore, this thesis investigates the different aspects in implementing deep neural networks for image recognition tasks in agriculture.

Two tasks were pursued in this thesis. The first task dealt with classifying images of potato tubers into four disease classes, as a basis of a sorting application. The second task focused on detecting tomato flowers in images for a drone pollinator. Data was acquired specifically for this thesis and served as training data for the deep learning algorithms. Both image recognition algorithms for the two tasks are based on deep convolutional neural networks. These types of deep neural networks have shown major improvements in image recognition performance in general and in detection and classification tasks in particular. Adjustments were made in order to adapt the algorithm to the specific tasks.

Results show high detection and classification accuracy, slightly better or comparable to other similar tasks. An exact comparison is hard to present due to the many different methods and performance measurements used in the literature.

The main contribution of this thesis is an empirical evidence a deep neural network framework can produce good results in image recognition tasks in agriculture. With further fine-tuning of the algorithm, through more data or otherwise, this approach could potentially be used to develop applications that automate and enhance the performance of agricultural tasks.

Keywords: Deep learning, object detection in agriculture, computer vision, convolutional neural networks, disease classification in agriculture, machine learning, automation in agriculture.

Publications

This thesis is in partly based on the following publications:

Journal papers under review

1. Oppenheim Dor, Shani Guy, Tsror Leah. Identifying Potato Disease from Visual Cues using Convolutional Neural Networks (Chapter 4).

2. Oppenheim Dor, Shani Guy, Edan Yael. Tomato Flower Detection Using Deep Learning (Chapter 5).

Conference papers

1. Oppenheim, D., Shani, G. Potato Disease Classification Using Convolution Neural Networks. 11th European Conference on Precision Agriculture, Edinburgh, Scotland, 2017 (Appendix 7.1).

Contents

List of Figures

List of Tables

List of Equations

1 Introduction

1.1 Description of the problem

As humans, we see the world around us easily. We can segment an object from its background and identify it regardless of the lighting or shading of the scene. We can recognize shapes and colors effortlessly, and in pictures with other humans in it, we can relate to their feelings and even guess their emotions. However, computer vision systems and algorithms have many difficulties perceiving the world around them as we do [88].

Computer vision in agriculture (agrovision) has been studied intensively for a few decades now. Agrovision systems have been developed for many agricultural tasks. However, despite the intensive research and development, commercialized applications that use agrovision are scarce [43]. The main reasons are **a)** variable and uncertain outdoor environment, mainly due to lighting and illumination conditions; **b)** complex and variable plant color, texture and shape structure; and **c)** the required speed and robustness in an agricultural application, leading to higher costs compared to human labor [22, 23, 46, 57]. These obstacles reduce performance in a basic and crucial step - *detecting the target object and classifying it in field conditions*, and make it hard for the agrovision systems to compete against manual labor [2].

Many approaches have been studied in order to address the recognition problem. Object recognition is about attaching semantic category labels to objects and scenes in a given image [55], mainly using different types of sensors and their combination with computer vision algorithms. Sensors such as color cameras, stereo cameras, and spectral or hyper-spectral cameras are used for image acquisition. These sensors mostly differ in the amount and type of data they acquire in every image, affecting acquisition and analysis time, and their cost [23]. In this thesis all algorithms were developed with the intention to analyze simple images acquired *through simple off-the-shelf RGB cameras*.

Along with the sensors, various computer vision algorithms are used for recognizing the target object, or in other words detecting and classifying it. Recognition algorithms rely on several types of features, derived from the image. Features such as color, texture, shape and their combination. Hand engineering these features have been abundant in the literature [42, 23, 43]. Yet in recent years the use of automatically feature learning algorithms such as deep neural networks and evolution constructed features have been steadily increasing due to their impressive performance [4, 66, 59, 51].

In recent research on object recognition in agriculture, focusing on fruit targets, researchers reported accuracy of detection algorithms were from 70% to 92% for citrus and apples [23], and 85% average detection and localization success [2]. Most researchers in these projects used hand engineered features for the recognition tasks. Both pointed out that an increase of accuracy and robustness of algorithms is needed in order for them to be implemented in real world applications. Furthermore, both noted that the comparison of the performances of methods is hard due to lack of a benchmark datasets and quantitative evaluation measures definition. Therefore, a comparison of the performance between recognition algorithms in general and between algorithms using hand and automatic engineered features is complicated. In this thesis the *image recognition algorithms developed focuses on an automatic feature learning methods using deep learning*.

Deep learning is a machine learning method used to learn representations (i.e. features) from raw data automatically, for detection or classification tasks [49]. In 2012 a group of researchers from Toronto used deep learning to train a Deep Convolution Neural Network (DCNN) which won the Large Scale Visual Recognition Challenge (ILSVRC) competition by improving the classification of the ImageNet database by more than 10%. They achieved a top-5 error rate of 15.3%, while the second best achieved 26.2% error rate on classifying 1.2 million high resolution images into 1000 different categories [47]. Since then, major technology companies initiated research and development projects in the field of DCNN and have been implementing them in their products and applications [49]. In agricultural application DCNN have been increasingly used only in recent years [4, 73]. This thesis' goal is to investigate and apply *DCNNs for agricultural recognition tasks and applications.*

1.2 Research Objective

This research focuses on developing, implementing and training Deep Convolutional Neural Networks for object recognition in two specific agricultural applications, both using RGB images acquired by various off-the-shelf cameras. The first application is to detect tomato flowers in greenhouse conditions for a drone pollinator. The second, is to detect and classify potato tubers' diseases for a potato sorting application. The objective was to develop robust algorithms for the two tasks with high accuracy and while minimizing the requirements of acquisition devices and conditions.

1.3 Thesis Structure

This thesis begins with a literature review (chapter 2) of computer vision research in general (2.1), delving deeper into the research and algorithms applied to the agricultural field (2.2). Next a review of deep learning focused on deep learning for image recognition is presented (2.3), following the recent advancements of deep learning in agriculture (2.4). Methodology of the two main research

conducted as a part of this thesis is depicted in chapter 3. The potato disease detection is described in chapter 4. The tomato flower detection research is described in chapter 5. Conclusions and future research are discussed in chapter 6.

2 Literature Review

2.1 Computer Vision Overview

Computer vision is an interdisciplinary field involving artificial intelligence, neurobiology, signal processing and more, that aims to duplicate human vision by perceiving and understanding an image or a video through pixels and their relations. Despite the ease in which humans perceive the world, interpreting an image as we do is a challenging task for computers due to several reasons.

First, transforming the 3D world to a 2D picture causes inevitable loss of information. The geometric structure of the eye or camera and the projective transformation, the main process of capturing an image, results a misperception of the distance of objects. Big far-away objects and small close-by objects can appear the same size and can be interpreted the same. Humans use long gathered knowledge and reasoning to solve this problem almost instinctively. However, computer vision systems have not yet been as successful [74].

Second, while measuring the real world computer vision systems captures a huge amount of data in the form of images or videos, a large part of the data can be considered as noise, making it hard and more complex for computer vision processes to understand the image and to perform tasks in real-time. In addition, the brightness measured in an image, which in theory could have provided physical information on objects, gives us little useful information, due to the fact that brightness is composed of many variables, and the dependencies between them are hard to decipher from pixels alone [24].

Finally, a computer analyzes the image through a keyhole, pixel by pixel, which makes it hard for it to understand the whole picture. Nevertheless, there has been impressive development in computer vision capabilities since its beginning. Today computer vision is used in a wide variety of applications, such as in medical imaging, autonomous cars, face detection, machine inspection for quality assurance and many more [83].

The process of computer vision applications include the acquisition of image data, followed by the processing and analysis of that data, and ending with interpretation of the image [26].

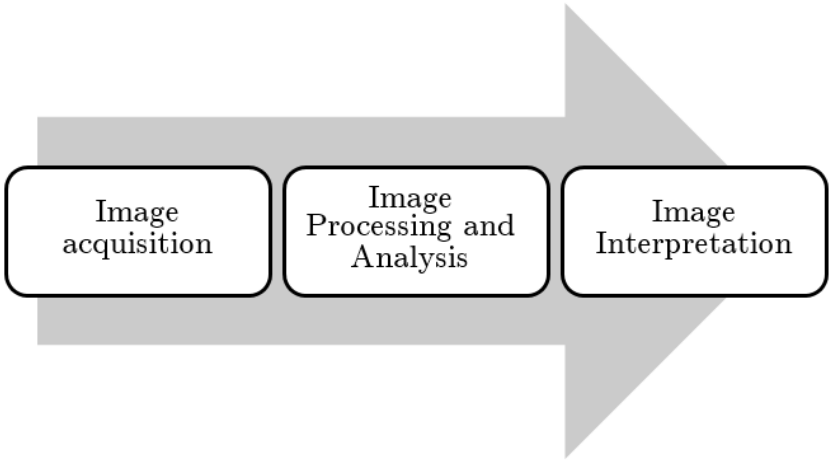

Figure. 2.1: Computer vision main process.

2.1.1 Image acquisition

Image acquisition and digitization is the process of capturing and storing the image data with a camera and a digitizing system. This process begins when the image in front of the camera is obtained and transformed into a matrix of discrete picture elements called pixels [88]. Each pixel receives a value that is proportional to the light intensity of that portion of the scene. These values are then converted to an equivalent digital value by an analog to digital converter. The digital value depends on the number of bits used in the vision system [26]. The simplest vision system is a binary vision, the light intensity of each pixel is decided using a certain threshold. The threshold divides each pixel into two values, black or white and thus a binary picture is assembled [83]. Similarly, a greyscale vision system can determine between different shades of grey according to the number of bits in every pixel [26]. More sophisticated vision systems can distinguish between greater numbers of intensities, such as 24 bits called true color. This method provides 256 shades of red, green and blue for every pixel, meaning 8 bits for each. As vision systems becomes more and more advanced, they acquire more information about the scene that can be used in a variety of applications [24], but is outside the scope of this thesis.

The basic and most usable image acquisition devices used in previous studies are color Charge-Coupled Device (CCD) cameras or complementary metal oxide semiconductor (CMOS) cameras, also called RGB cameras, and in early studies black and white cameras [42]. However, the use of thermal, multispectral, hyperspectral, depth or a combination of the cameras is also investigated for computer vision applications [23].

2.1.2 Image processing and analysis

After the image is acquired, the processing and analysis of its data begins. Several techniques have been developed to understand image data. One of the first and most studied category of techniques

is called image segmentation [88]. It is a critical and essential component for image analysis. The goal of this category of techniques is to define and separate points or regions of interest in the image [26]. Image segmentation algorithms are based on either the discontinuity principle or the similarity principle [8]. The idea behind the discontinuity principle is to extract regions that differ in properties such as intensity, color, texture, or any other image statistics. The idea behind the similarity principle is to group pixels based on a common property [83]. Thresholding is an example for segmentation that uses the discontinuity principle. A predefined threshold divides the picture into a binary value according to its intensity level. If a pixel's intensity value is greater than the threshold, it is given a binary value of white, 1 for example. Otherwise, if the pixel's intensity level is lower than the threshold, it is given a binary value of black, 0 for example [26]. The simplest approach to segment an image is based on the similarity assumption is that every pixel is compared with its neighbor for similarity check (for gray level, texture, color, shape). If the result is positive, then that particular pixel is "added" to the pixel and a region is "grown" like-wise. The growing is stopped when the similarity test fails. These procedures of reducing the image to a binary image or forming similarity regions aids in defining and identifying objects in an image [83]. Another category in the processing of the image that normally follows image segmentation is features extraction. Through these features the machine vision algorithm defines an object in the image [26]. Some of the features extracted from the segmented image are simple features such as area, width and length and others more complex ones, such as center of gravity, shape and aspect ratio [83]. The combination of a few features together usually describes the object in a satisfactory way so as to understand and interpret the image and recognize objects in it [88]. Usually the features are defined empirically, this requires expertise in the specific domain. Many techniques have been developed for feature extraction [3, 28, 53]. Modern computer vision algorithms such as deep convolutional neural networks extract features automatically from images through supervised learning (reviewed in section 2.3.1) [27].

2.1.3 Image interpretation

Based on the extracted features, image interpretation is usually the final stage and objective of the machine vision system [74]. The main goal of this stage is to recognize the objects in the image by comparing it to predefined models or standard values. The most common and simple technique is template matching. This method compares one or more features extracted from the image to a predefined model or template representing the object of interest [26]. More sophisticated techniques involve machine learning, artificial neural networks and many more methods that offer solutions to the interpretation and classification of an image [24].

2.2 Computer Vision in Agriculture

2.2.1 Overview

[80] predicted that due to the combined factors of increased international competition in the agricultural section, advances in computer technology, the low cost of new technologies and the use of intelligent and automated machines applications in agriculture was supposed to be imminent. Indeed, many tasks have been automated since, and computer vision has a major role in the automation of these tasks [91]. Tasks such as autonomous navigation and obstacles avoidance [50], precision and selective spraying [90], weed control [82], yield estimation [44], ripeness and quality evaluation [22], meat, fish and ham quality control [40, 52] and fruit detection for harvesting and counting [23, 42, 43] have been automated through agrovision or researched for automation using agrovision. Still, many problems hinder the widespread commercialization of these tasks [22, 46, 57].

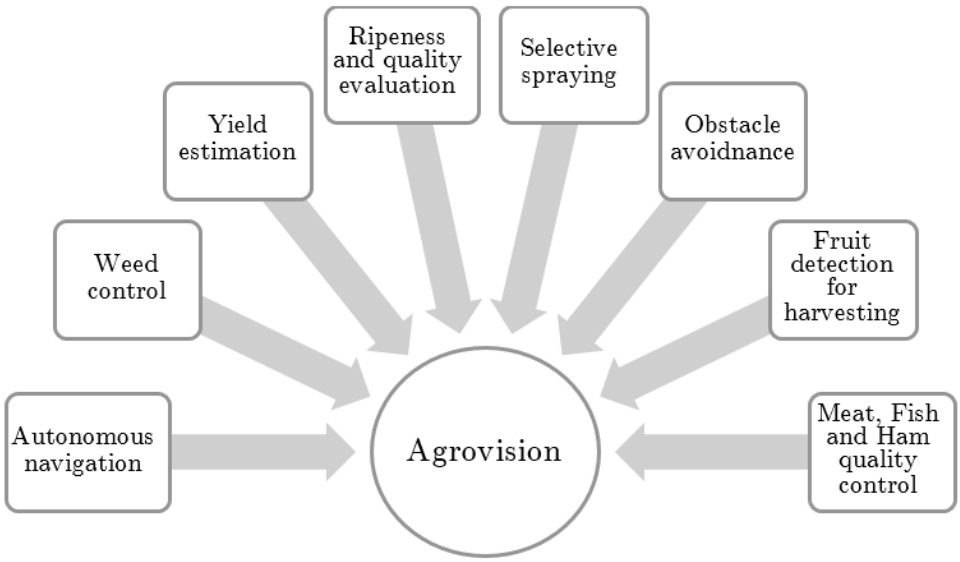

Figure. 2.2: Examples of computer vision applications in agriculture

2.2.2 Object recognition in agriculture

Detection, classification and localization of objects in an image is a crucial aspect in the development of agricultural applications. In order to harvest fruits or vegetables, navigate in the field, spray selectively etc. the objects' location in an image must be determined. According to the task, an object's 3D location has to be calculated, obstacles in the way have to be detected and object's characteristics such as ripeness and size have to be estimated. For some applications the presence and classification of diseases is sought as well [23, 22, 42]. Significant object recognition research has been conducted on a variety of fruits and vegetables. According to Jimenez et al. [42] the major steps involved with object recognition in agriculture are **a)** Image acquisition **b)** Image preprocessing, for restoration or enhancement and **c)** Image analysis to detect the target.

The major challenges with object recognition in agriculture is associated with the highly unconstrained, unstructured environment in which the target objects are situated [43, 23, 42, 2]. The main attributes of this environment are: **a)** the variability of the objects' colors, shapes, sizes, textures, and reflectance characteristics; **b)** the highly unstructured scenes with great levels of uncertainty and complexity; **c)** the changing illumination and lighting conditions; **d)** and the inevitable occlusions created by other fruit or vegetable, or by the surrounding leaves and branches [42, 23].

2.3 Deep Learning for Image Recognition

2.3.1 Deep Learning Overview

Deep Learning (DL) is a type of representation learning [25]. These kinds of learning methods allow the computer to create computational models that learn internal representations (i.e. feature vectors) of raw data so as to discover the specific representations needed for a detection or classification task. Until the introduction of such representation learning techniques, researchers and engineers carefully designed and crafted feature extractors, requiring sufficient domain knowledge to engineer good extractors that transform the raw data into useful internal representations [76]. This process was than followed by a linear or kernel learning algorithm, such as a support vector machine or regression, that used the human-crafted feature vector to produce a detector or a classifier [49]. These feature extractors were usually designed by the researcher's intuition about breaking down the problem into sub-problems and multiple levels of representation.

A typical way to extract features in object recognition problems in images begins by transforming the raw pixels into gradually more abstract representations. For example, starting by a search for edges in the image, followed by forming more complex shapes, up to identifying more abstract categories which form the objects and finally putting them all together to recognize what is needed in the image (see figure 2.3). This process is difficult not only due to the challenge of finding the appropriate representations for each level of abstractions, but also because of its task specific nature [7].

For high dimensional data inputs such as images, highly varying mathematical functions are needed in order to express the intricate statistical relationships, for successful object classification or detection. An object, such as a flower, can be resembled by many possible images which can differ from each other greatly if investigated at the pixel intensity level or through edges and shapes alone [7]. Thus, in order to capture these complex feature combinations a hierarchical processing structure can be used to form suitable representations of the data, according to the "neuron doctrine" of perception [5]. Therefore, many researchers have been trying to develop deep architectures learning algorithms for many years [76]. These architectures learn feature hierarchies automatically from the raw data. This capability is especially important for high levels of abstractions, which are

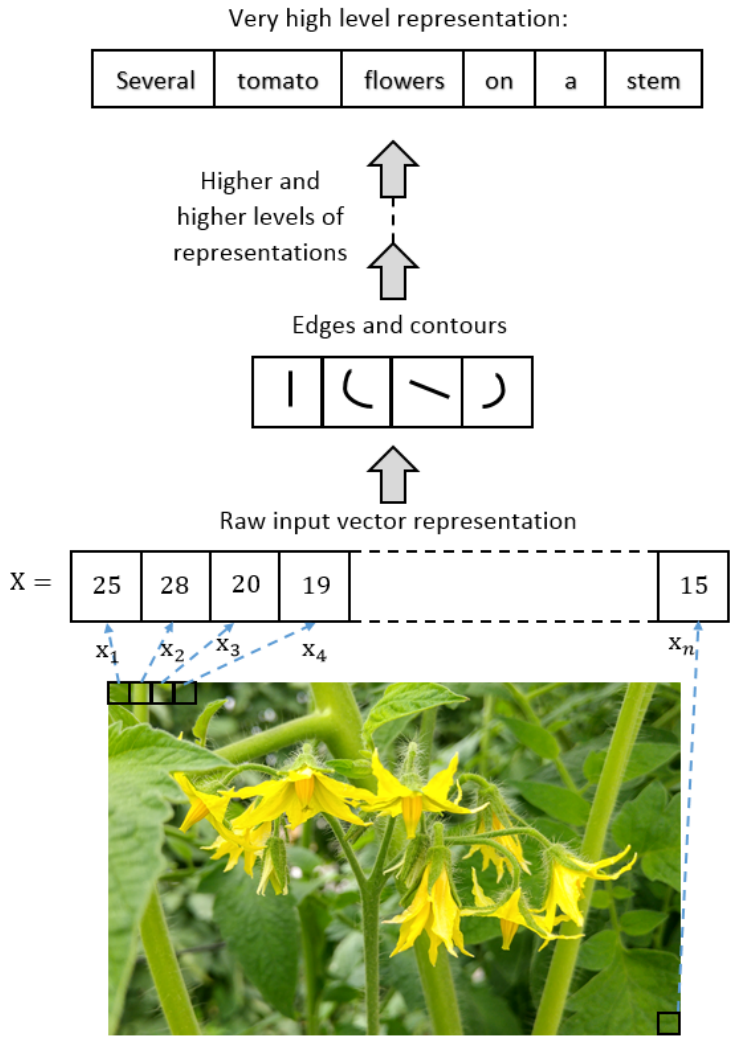

Figure. 2.3: The raw input image is being transformed into gradually higher levels of representation, representing more and more abstract functions of the raw input, e.g., edges, local shapes, object parts, etc. until a very high level of abstraction we can expect from a human. Figure inspired by [7].

harder for researchers and engineers to characterize specifically [7]. This thesis' uses concepts and approaches from the prevalent sub-field of deep learning architectures - *Deep Learning in Artificial Neural Networks* (ANNs).

2.3.2 Deep Learning in Artificial Neural Networks

ANNs are computing models biologically inspired by the study of neural computation in humans and other animals. A standard artificial neural network consists of many simple, connected, information processors called neurons [76]. These neurons operate as simple, non-linear components that transform the representation at one level of the hierarchy, starting from the raw data, into a representation at a higher slightly more abstract level. Given enough of these hierarchical transformations, very complex functions can be learned [49]. However, ANNs' architecture were not always

deep, and only until several breakthroughs both in artificial intelligence (AI) and neuroscience and the collaboration between the fields, DL in ANNs could have become implementable [29]. The introduction of the backpropagation algorithm for training, convolutional neural networks for image recognition and dropout layers are important examples [47, 76]. In addition to the algorithmic improvements, the utilization of graphical processing units (GPUs) for deep learning tasks enabled successful scaling up of deep learning algorithms. The use of GPUs allowed the implementation of bigger models which can be learned from larger sets of data [67, 11].

2.3.2.1 The Artificial Neuron

The artificial neuron was initially proposed by Rosenblatt et al. [71] and was a model to describe how the brain perceives and saves information of its surrounding. The model is loosely inspired by the biological model of a neuron (see figure 2.4). Each neuron receives input signals from its dendrites, calculates an output signal along its axon and "fires" an output through the axon terminals if a certain threshold is passed.

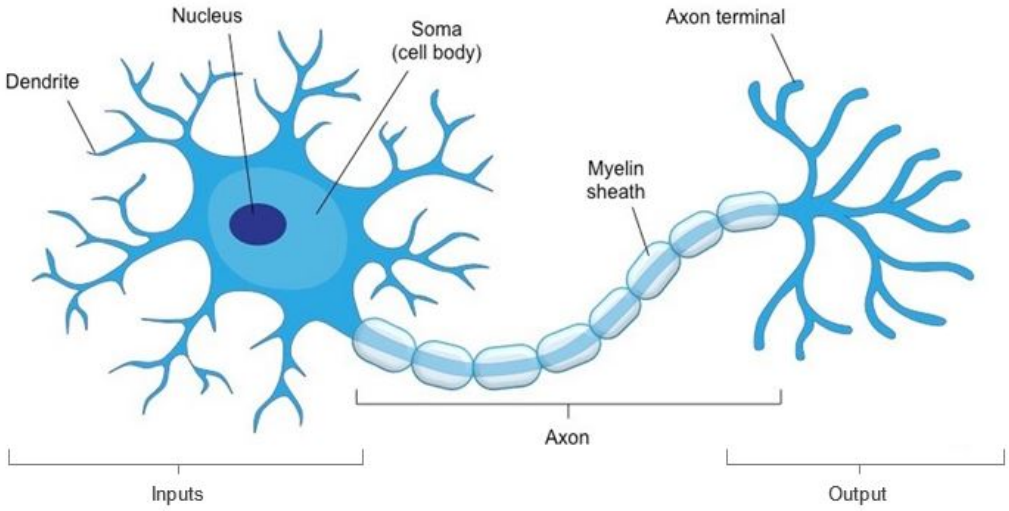

Figure. 2.4: Model of biological neuron [89]

The artificial neuron model is composed of an input vector X that is multiplied by a weights vector W, summed and than inserted into a certain activation function which determines if the neuron is activated and an output is generated (see figure 2.5). The main idea is that the synaptic strength (e.g. W) is learnable and determines the amount of influence an input has on the final output of the artificial neuron. There are many types of activation functions. Here are depicted three of them.

Sigmoid The Sigmoid non-linearity activation function has the following mathematical expression

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

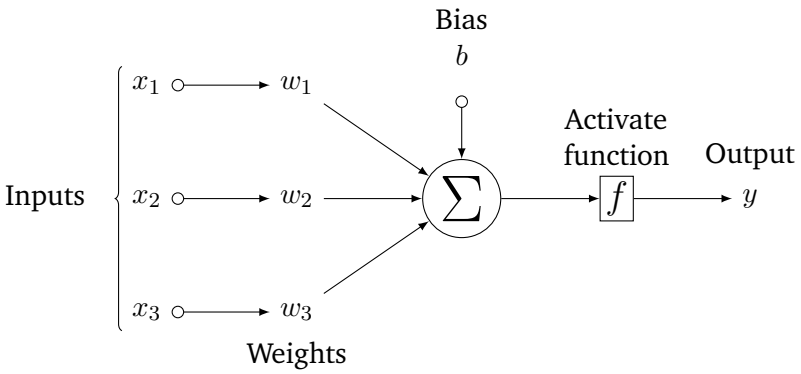

Figure. 2.5: Model of artificial neuron

The Sigmoid function takes the input value and transforms it between 0 and 1. This transformation causes the gradient in the extremes, 0 or 1, to be nearly zero (This phenomena is also called saturating non-linearity). Thus, the learning procedure fails to modify the weights parameters adequately.

Hyperbolic Tangent The Tanh non-linearity activation function has the following mathematical expression

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.2}$$

The tanh transforms input values number between -1 and 1. Thus, it suffers from the same saturating non-linearity as the Sigmoid activation function.

Rectified Linear Unit The ReLU non-linearity activation function has the following mathematical expression

$$ReLU(x) = max(o, x) \tag{2.3}$$

The ReLU has become very popular in recent years as it does not suffer from saturating non-linearity. In addition it accelerates convergence, does not suffer from vanishing or exploding gradients and the function is computed relatively cheap compared to other activation function. However, it does not allow negative information [47].

A formal mathematical equation of the computational process of an artificial neuron j with inputs X_j, weights W_j and bias b_j is displayed in equation 2.4.

$$\alpha_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{2.4}$$

The final output of the artificial neuron with a ReLU activation function would be computed as follows:

$$y_j = ReLU(\alpha_j) = max(o, \sum_{i=1}^{n} w_{ij}x_{ij} + b_j) \tag{2.5}$$

2.3.2.2 The Artificial Neural Network

Combining artificial neurons together and arranging them in layers form artificial neural networks. The motivation behind this kind of structure is depicted in 2.3.2 and is also inspired by the biological brain. Nonetheless, the human brain has an approximate of 86 billion neurons that are connected by some $10^{14} - 10^{15}$ synapses [33], so ANNs, which are comprised of much less components, are only a very simplistic model of the brain [29]. A typical ANN is constructed of at least an input layer and an output layer. Layers in between these two are usually called hidden layers. Each layer is comprised of artificial neurons which are connected to each neuron in the next layer, additionally called fully connected layers (see figure 2.6).

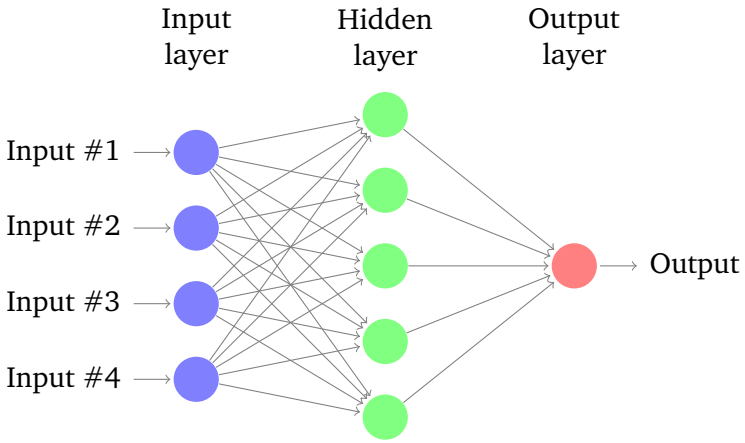

Figure. 2.6: An example of an ANN consisting of 2 fully-connected layers (one hidden layer of 5 neurons and one output layer with one neuron). In an N-layer neural network we do not count the input layer

The output of the network is computed sequentially, layer after layer. This type of sequential layers structure allows ANNs to be evaluated by using matrix operations, an easy and efficient computation process. Therefore, in practice the input layer's output is computed by a matrix multiplication of the input vector by the weights matrix, followed by an activation function. Thus, the output of each layer is the input of the consecutive layer. For example, a full forward pass of data processing of the ANN in figure 2.6 amounts to simply two matrix multiplications interwoven with the the activation functions [25].

In theory an ANN is a universal approximator. That is, given any continuous function $f(x)$ and a positive non-zero ϵ there exists a $G(x)$ ANN with one hidden layer and a non-linear activation function such that $\forall x, |G(x)\text{-}f(x)| > \epsilon$ [12]. Nevertheless, even though one hidden layer in a ANN can approximate any function in theory, the ANN may not be able to learn and generalize correctly

[7]. In addition the ANN could be infeasibly large for implementation and in practice that is the case. Many empirical studies show that shallow ANNs, with one hidden layer, perform poorly in many cases compared to deeper architectures [25]. In recent years deep ANNs have been making major advances in solving problems that have been studied in the AI community for years. Problems such as image recognition, speech recognition, natural language understanding and many more [49].

2.3.2.3 Training an Artificial Neural Network

ANNs' ability to learn representations of data is their main asset. The goal of the learning stage is to find the optimal parameters of the network so the network produces the desired output. The learned parameters are the weights of the network, W, and the biases, b. Learning is carried out by feeding the training data into the network, called the feed forward stage, and in iterative process adjusting the weights of the network to produce better results, also called the backward stage. The ANN is considered to be trained after reaching the desired performance level or after a predetermined time period.

In this thesis learning is done by training a deep artificial neural network (DNN) on labelled datasets. The datasets include images with the desired objects for the recognition tasks and their labels. This type of learning is called *supervised learning*. Learning from an unlabeled dataset is referred to as *unsupervised learning* and require different and specialized algorithms.

Training a DNN for image recognition requires to determine a few key concepts - a score function, a loss function and an optimization method.

Score function The score function maps the raw data input into a score, determining the output's success in the task. For example, in classification tasks, the score function refers to the class score of the raw input data.

Choosing the architecture for the network is actually defining the score function. As the input data will flow through the network and change according to the network's parameters and structure and output a score in each output neuron.

Loss function The loss function quantifies the agreement between the predicted scores of the network to the ground truth labeled data.

Determining the loss function is another decision to make when training an DNN. In general there are many loss functions that can be used and usually the choice depends on the task [41]. For image recognition cross-entropy or hinge loss and their variations are the most commonly used

functions. For example, if we abbreviate $f = f(W, x_i)$ as the activation function of the output layer in a DNN than the cross-entropy function loss will be a softmax function:

$$L_i = \frac{e^{f_{y_i}}}{\sum_j e_i^f} \tag{2.6}$$

followed by a cross-entropy function, together typically called a **softmax classifier**:

$$C = -\sum_{c}^{N_c} L_c log(L_i) \tag{2.7}$$

where the correct target label is L_c.

Optimization While the loss function quantifies the quality of a particular set of weights of the network, the aim of the optimization is to find the set of weights that minimizes the loss function.

The core idea of optimization is to search for the best next set of weights in an iterative and effective process. Given the data loss of the output layer it is possible to calculate the derivative for each neuron in the output layer that will minimize the loss function. Because the loss is composed of a sum of sub-functions evaluated at different sub-samples of data, optimization can be made more efficient by taking gradient steps with respect to the individual sub-functions, i.e. stochastic gradient descent (SGD). Although it has no theoretical proof of good convergence, SGD converges to local minima in practice [45]. Furthermore, based on the chain rule for derivatives it is possible to adjust the other interconnected neurons of the network through *backpropagation* [32]. The procedure relies on the fact that the derivatives (or gradients) of the loss function with respect to the input of the ANN can also be computed by working backwards from the derivative of the output of the ANN or any other preceding layer, hence *backpropagation* [49].

Over the years many methods were developed for efficient stochastic optimization. Some were utilized for DNNs as well. Other than SGD, some optimization algorithms rely on more advanced techniques such as momentum, second order approximation and adaptive learning rates [85, 45]. They are known to converge faster and their parameters are sometimes easier to fine-tune. However, each iteration of the algorithm take more processing time and use more memory.

After deciding on the DNNs architecture (i.e. score function), the loss function and the optimization method and before training can begin there are a few more crucial steps to be made - the **initialization** of the weights of the DNN and picking **regularization** methods.

Initialization The process of deciding on the initial values of the weights and biases of the DNN.

As optimization methods in DNNs are usually iterative, initialization affects strongly the success of the learning process due to local minima cnvergence. The initial weights can determine how quickly the algorithm converges, or will it converge at all [62]. Most initialization techniques today are heuristic and simple because neural network optimization is not yet well understood. Therefore, these heuristics are based on achieving good results at the beginning of learning but it is not well understood if those properties are kept throughout the whole learning process. One property that is known to be beneficial for initialization is that the initial weights of two artificial neurons with the same activation functions need to be different from one another if they are connected to the same inputs [25]. So DNNs are generally initialized with Layer-sequential unit-variance (LSUV). This method initializes each weight of the DNN as a Guassian random variable with a mean value of zero and standard deviation of $\frac{1}{\sqrt{n_{inputs}}}$, biases are initialized to zero as well [58].

Another option for weights initialization is through a method called - transfer learning. This method uses the values of weights and biases of a pretrained DNN as the initial weights for training. This method exploits what has been learned in one setting for the task in hand. The assumption is that when using transfer learning, many of the factors that explained variation in the first setting are relevant for the pursued task [25]. In image recognition problem many visual objects share low level features such as edges, shapes and other geometric features, which can be used to form higher level features for recognition [62].

Regularization A central problem in machine learning which takes place in deep learning in neural networks as well, is overfitting. In such cases the learning process adjusts the DNN to features specifically apparent in the training data, which can cause the trained DNN to generalize poorly and perform badly on unseen test data . To solve this problem many regularization techniques were developed that make modifications in the learning algorithm, intended to create DNN models which generalize better [25].

The main method in which overfitting is identified is by splitting the training data into a training and a validation set. Testing the DNN model during training on the validation set, can show error rates on data independent from the data the model is trained on. This way overfitting can be recognized. At a certain stage of training, the algorithm begins to adjust the weights to fit the training data, which can be seen after the stopping point in figure 2.7. At this point the validation error begins to increase while the training error continues to decrease. Thus one of the most popular and effective method to combat overfitting is to stop training when the validation begins to increase. This is called early stopping and is carried out by saving a copy of the model each time it is tested on the validation set, enabling to choose the model which performed the best over the validation set. In addition there are methods to penalize parameter size, augment data, use dropout layers, use ensemble of DNNs and more [25].

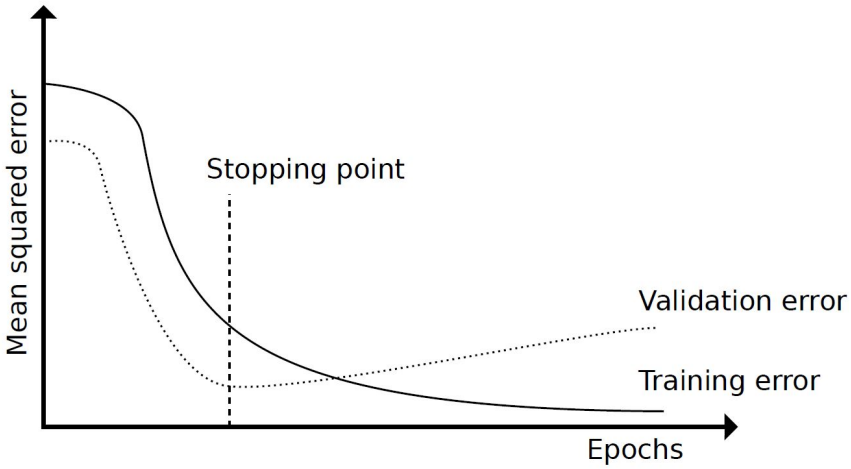

Figure. 2.7: Graph comparing training and validation errors as training progresses.

2.3.3 Deep Convolutional Neural Networks

2.3.3.1 Overview

Convolutional neural networks (CNN) are a specialized kind of neural network for processing data that has a known grid-like topology [25]. Their name is derived from the mathematical operation called discrete convolution which is a special kind of linear operation between matrices, used in CNNs. Their architecture is inspired by research of the mammalian visual cortex that revealed how visual input is filtered and pooled in a hierarchical structure [36]. [19] modeled those discoveries to an artificial neural network visual processing mechanism which is the basis of modern CNNs [29]. Therefore a typical usage of CNNs is in recognizing objects in an image. Accordingly, one of the major breakthroughs in deep convolutional neural networks (DCNN) was a DCNN called AlexNet developed for an image recognition competition called ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The ILSVRC is a benchmark in object category classification and detection on hundreds of object categories and millions of images [72]. In 2012, a group of researchers from Toronto won the competition by improving the classification of the ImageNet database by more than 10%. They achieved a top-5 error rate of 15.3% while using AlexNet, while the second best achieved 26.2% error rate [47]. This leap of classification accuracy increased interest in research and development of DCNNs and since, DCNNs have been used to solve object recognition problems in many fields [49].

2.3.3.2 Architecture of CNNs

CNNs are designed to process input data in the form of matrices or tensors[1]. For example, color images composed of three matrices i.e. a tensor, containing information of the pixel intensities

[1]Tensors are high dimensional generalizations of matrices. Tensors of order zero are simply scalars, tensors of order one are vectors and so on.

in the red, green and blue channels are common inputs for CNNs. The input is processed by propagating through the CNN's layers, which modify it according to the characteristics and the architecture of the layers. This processing creates a representation of the input data that can be used for different tasks. CNN's architecture greatly affects the processing of data and are key for improving CNNs performances. New architectures are developed rapidly. Every few weeks to months, a new best architecture for a benchmark database is announced, making it hard to specify a one best architecture [49]. Still, most of them consist of three main layer types: convolutional layers, pooling layers and fully connected layers [27]. Each layer plays a different role in the overall design of the network.

Convolutional layer The convolutional layer applies a convolution operation to the input and passes the result to the next layer. The main purpose of the convolution layer is to detect local features or a conjunction of local features by applying sets of filters to the input. The hierarchical structure of the convolutional layers allow the composition of high-level features from lower-level ones. In images, a combination of edges form shapes, shapes assemble into parts, and parts compose objects.

The convolutional layer takes the input image x and a set of filters $F = \{f_1, f_2, ...f_{N_k}\}$ and applies the convolution operation, $\otimes$ between them. This operation produces a set of N_k feature maps h:

$$h_k = f_k \otimes x \tag{2.8}$$

Each filter is a small 3D or 4D tensor (depending on the dimension of the input and output) learned during training, much smaller than the image itself. These filters are composed of artificial neurons described in section 2.3.2. A filter that correlates well with a local region in the image produces a strong response which is apparent in the feature map (see figure 2.8 for an illustration of the process). In other words, a filter searches for a certain kind of feature in small local regions in the image and if the feature is detected it produces a response that is transferred to the feature map.

The convolutional layer exploits two main characteristics typical to images. First, nearby pixels in images are often highly correlated and form distinctive local features. Second, these local features are invariant to location in the image. So the fact that the architecture of the convolution layer takes into consideration these characteristics, bring about three main advantages: **a)** weight sharing over the entire image reduces the memory requirements of the model and the number of computational operations. **b)** local connectivity learns correlations among neighboring pixels, and **c)** equivariance to translation, meaning that if the input changes, the output changes in the same way. Hence, if an object is moved in the image its representation will move the same amount in the output [27, 25].

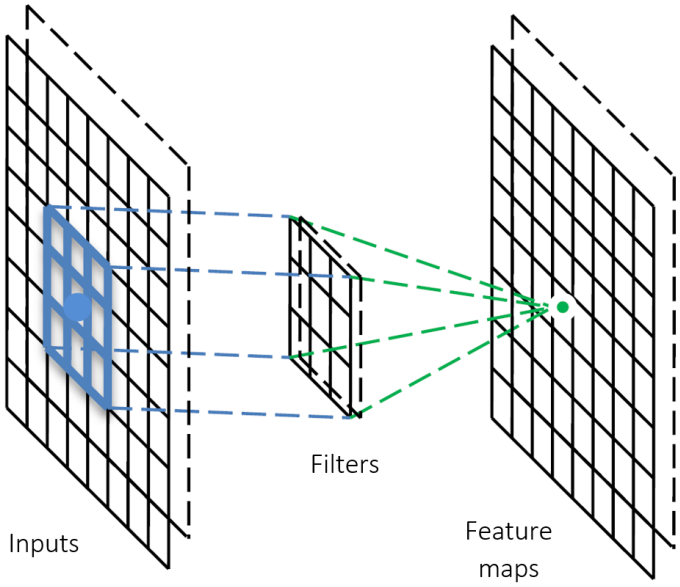

Figure. 2.8: The operation of the convolutional layer [27].

Pooling layer Although the role of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to merge semantically similar features into one [49]. This merge provides invariance to slightly different input images and to relative positions of the features.

The pooling layer replaces nearby outputs of the previous convolution layer with a summary statistic, typically average or max operations. Similar to the convolution layer, pooling is translation invariant, because their computations take neighboring pixels in feature maps into account (see figure 2.9). In addition, the summary of a response from a whole neighborhood to one statistic value reduces the dimension of the representation. [75] conducted a comparison between max and average pooling and found that max-pooling can lead to faster convergence, select superior invariant features and improve generalization. Formally, pooling executes the chosen summary operation over the feature map, in a small spatial region R:

$$p_R = \max_{i \in R} h_i \tag{2.9}$$

There are many other approaches related to the pooling layers. One approach called spatial pyramid pooling (SPP) deals with the fact that CNNs require a fixed-size input image. The SPP can extract representations in a fixed length from a previous layer, allowing for arbitrary region or image scale, size and aspect ratio.

Fully connected layer Following multiple convolutional and pooling layers, the CNN typically ends with one or more fully connected neural network layers. These layers perform like a

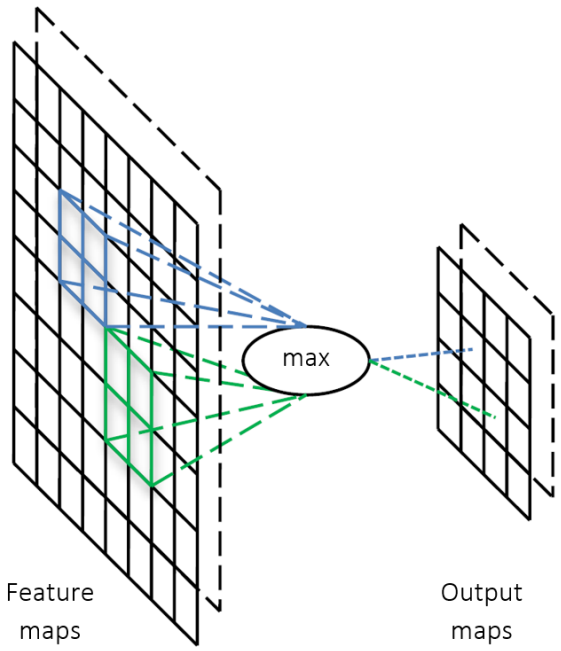

Figure. 2.9: The operation of the pooling layer [27].

regular ANN and are the last process before the final classifier. They provide the high-level reasoning of the network and are designed according to the task at hand.

Fully connected layers convert the 2D feature map from the last pooling layer to a feature vector, for further processing (as seen in figure 2.10). Although there are usually only a few fully connected layers they contain approximately 90% of the parameters [27].

Finally, the last fully connected layer is fed into a classifier which represents the target labels y as a vector, h^l where l is the last layer of the model, with N_c elements, where N_c refers to the number of classes to discriminate between. A classifier, such as the softmax classifier, is then calculated for each element in the vector h^l:

$$\hat{y}_c = softmax(h_i^l) = \frac{e^{h_i^l}}{\sum_j h_j^l} \tag{2.10}$$

When training the network, a loss is calculated. A softmax classifier uses a cross-entropy loss for optimization:

$$C = -\sum_c^{N_c} y_c \log(\hat{y}_c) \tag{2.11}$$

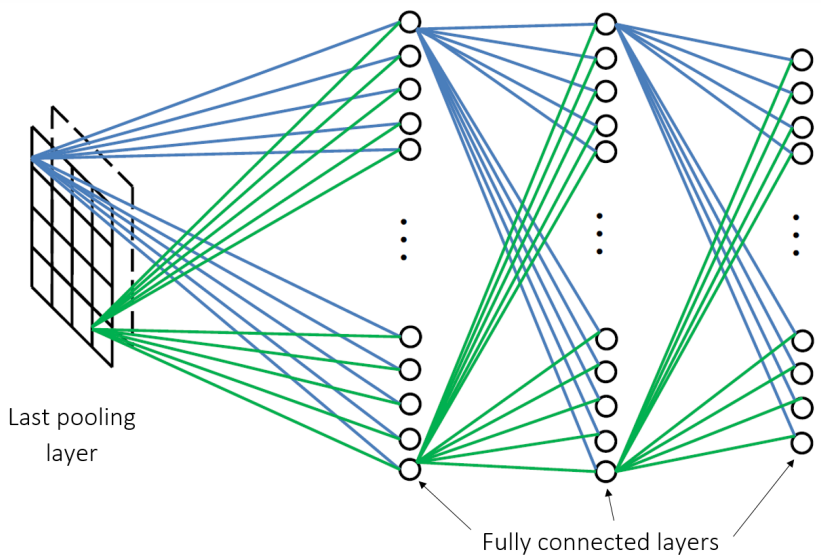

Figure. 2.10: The operation of the fully connected layers [27].

Figure 2.11 shows the general architecture of CNNs. The configuration of the various types of layers has a major effect on the classification performance. Some of the most successful CNN architectures try to replicate the hierarchical organization of mammalian cortical systems, with both convergent and divergent information flow in successive, nested processing layers [29]. Similar to the convolutional and pooling layers and the successive non-linear computations in layers.

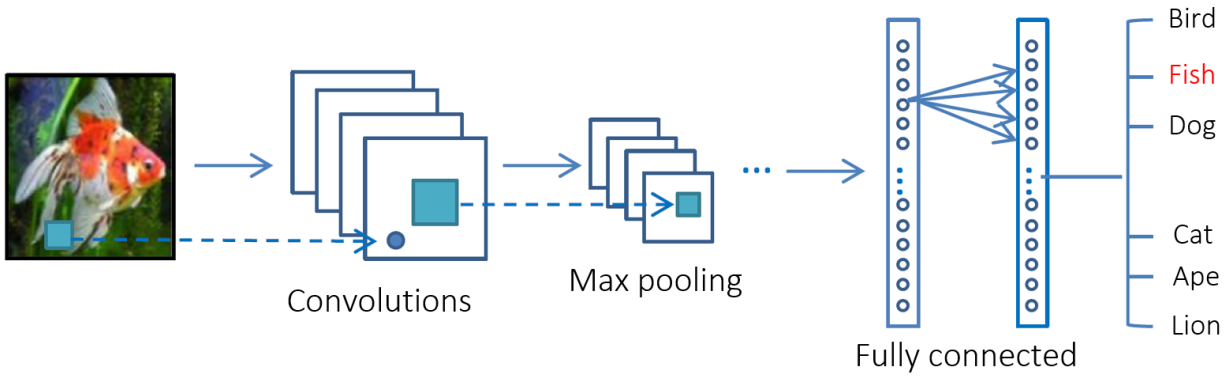

Figure. 2.11: A general architecture of layers in a CNN [27].

A few remarkable CNN architectures have advanced CNN's performances. AlexNet [47], as noted before, was an important architecture that set the tone for many following researchers. It consists of five convolutional layers with intermediate pooling layers followed by three fully connected layers and produced state-of-the-art results on the huge ILSVRC2012 database. Still, it had two major drawbacks: **a)** a fixed image size is required, and **b)** there was no clear understanding as to why it performed so well. In order to gain insight about the intermediate layers, [93] developed a visualization technique which enabled them to modify their network's configuration to outperform AlexNet and win first place in the ILSVRC in the succeeding year. Those visualizations reveals the

features learned in training are interpretable patterns, such as edges, contours and shapes as can be seen in figure 2.12.

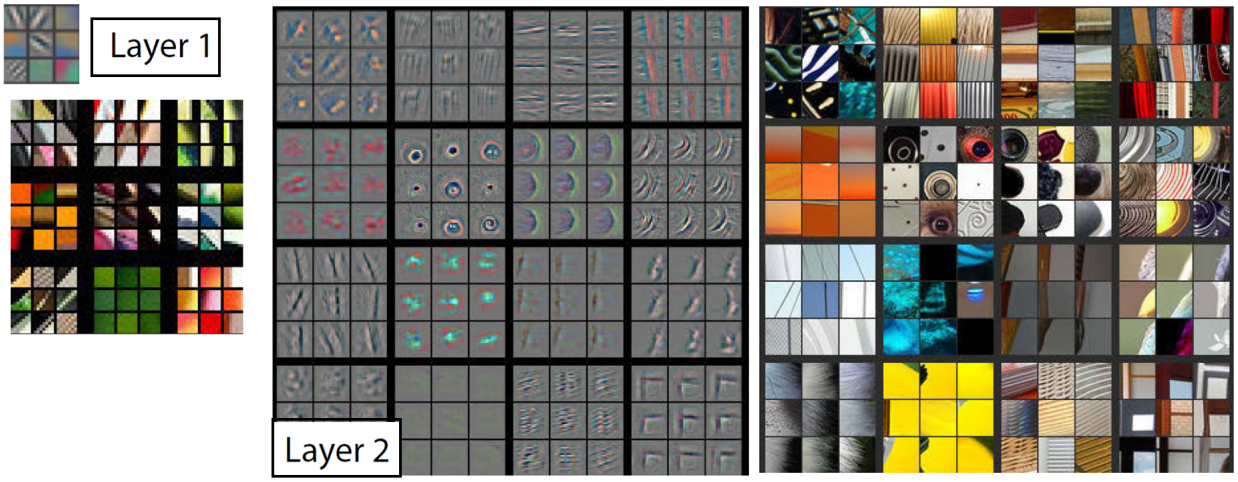

Figure. 2.12: A visualization of the first and second layers of the CNN trained and visualized by the technique developed in [93]. Reconstructing patterns from the validation set that caused high activation values in a the first and second feature maps created these visualizations. Near every feature map in the figure the corresponding image patch is shown. It is clear that the feature maps that the model learned are distinctive and intuitive.

Deeper neural networks were assumed to provide better results. Recently, they have even been proven to provide better or comparable results to shallower models [16]. Accordingly, the next breakthroughs in classification performance were deep architectures such as the VGG [78] network and the GooLeNet [86] network, which had 16-30 and 22 layers respectively. However, training these very deep networks became harder. Vanishing or exploding gradients became a problem, solved with the aid of intermediate normalization layers. Nevertheless, a new problem arised, degradation, which lead to higher training errors for deeper networks not caused by overfitting [84]. So the next breakthrough came with the introduction of the deep residual learning framework. [30] developed a deep residual neural network called ResNet. The core idea of ResNets is to create identity shortcut connections between layers that allows skipping one or more layers. Skipping unnecessary layers is learned during training time (see figure 2.13 for the main building block that was added).

Based on the deep residual network framework, a ResNet with 152 layers won the first place in the ILSVRC and the Common Objects in Context (COCO) 2015 in several computer vision detection and classification tasks [30].

CNNs did not only achieve state-of-the-art results in image classification tasks, but were part of the improvements in object segmentation and detection tasks. Further details in section 2.3.4.

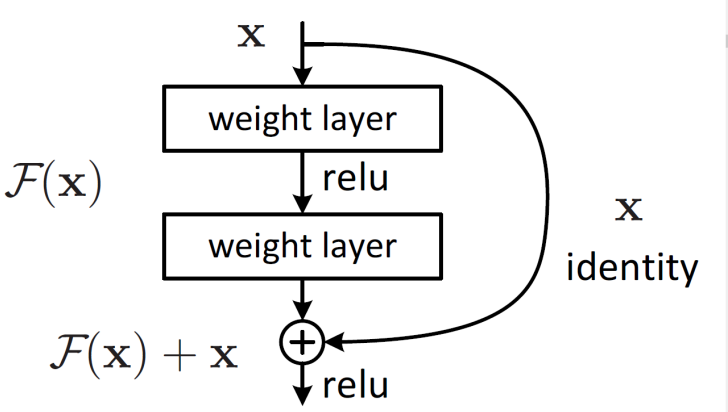

Figure. 2.13: Residual neural network building block [30].

2.3.3.3 Training CNNs

As in ANNs, CNNs are trained via a forward and backward stage. The forward stage represents the input image and maps it into a score. In classification tasks, for example, this score represents the prediction of the CNN about the input image class category. The prediction is than used to calculate the loss cost according to the ground truth class of the image and the loss function chosen for the network. Based on the loss cost the backward stage begins and gradients of each parameter of the CNN are computed, updated and prepared to the next image. Training is complete after sufficient iterations of forward and backward iterations.

In practice successfully training a deep CNN requires more than just good knowledge of what algorithms exist and the way they work. The choice of an algorithm for a particular task at hand, different methods of regularization, hyperparameters and the methods of training highly affect overfitting and results in general [25]. In image recognition problems there are several commonly used methods for successfully training CNNs and avoiding overfitting.

Weight decay Weight decay adds a term to the cost function to penalize the model's weight parameters size, preventing them to fit exactly the training data and suppress irrelevant components of the weights, improving generalization [48].

Dropout When using dropout while training a CNN, on each forward pass of a training case, artificial neurons are randomly omitted from the network with omission probability often different than 0.5. This technique prevents complex co-adaptations on the training data, meaning that the artificial neuron does not rely on its connections to other neurons [34]. Another way to view dropout is as an efficient way to implement ensemble learning, as every training case is processed through a different network. There is a variety of improvements and versions to the original dropout method, all working to decrease overfit and improve generalization.

Data augmentation Data augmentation is a technique used to generate additional training data without the need to label or acquire more data. Simple and well known techniques are cropping, rotating, and flipping input images producing more new images from the existing labeled training data. More advanced method include changing the style of the image and applying transformations corresponding to color and contrast and more.

Transfer learning Transfer learning refers to initializing the weights of the CNN with pre-trained parameters, followed by adapting and fine tuning the last output layer according to the new specific task. Many researchers use this method by initializing their CNN with the weights of networks such as AlexNet, VGG and ResNet, which were successful in the ILSVRC competitions [27].

2.3.4 Deep Learning for Object Detection

Object detection tasks are closely related to image classification, but differ in one main aspect. When classifying an image all the pixel data in the image is utilized as the input to predict the class of the object in the image. However, in object detection tasks, the position of object or objects must be estimated prior to classifying the class of the object.

Shortly after the success of AlexNet [47] in the classification task, many researchers tried applying CNNs for detection. A general approach is to generate many bounding boxes proposals as candidates for objects and classify those using CNNs. Two main problems arise from this approach. First, searching for potential objects in an image highly affects the performance of the algorithm, but proposing too many and then classifying them is very computationally intensive. Second, involves localizing the bounding boxes proposals to fit the object detected [77].

The most representative approach is the Regions with CNN features (R-CNN) [21]. It uses a selective search algorithm [92] to generate bounding boxes proposals, extract useful features using a CNN and classifying each with an SVM (see figure 2.14). Using the R-CNN scheme on the PASCAL Visual Object Classes (VOC) 2010 dataset achieved a mean average precision (mAP) of 53.7%, compared to the 33.4% of the than popular deformable part models [18]. The PASCAL VOC is a challenging benchmark dataset most widely employed for the evaluation of object detection algorithms [17]. There are 20 object classes labeled and localized in this dataset. The goal is to predict the bounding boxes and class of each object in the test images.

Following the improvements achieved by the R-CNN, two main research directions initiated. The first, was to decrease the training and testing process in various methods such in [69, 31, 68]. The second approach was to improve the accuracy of the bounding boxes proposals algorithms, such as EdgeBoxes [94] and BING [10].

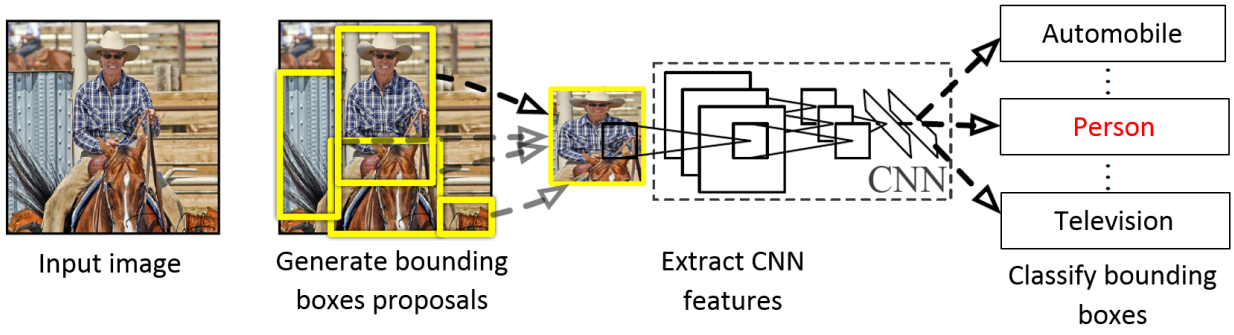

Figure. 2.14: R-CNN algorithm overview [21].

2.3.4.1 Faster R-CNN

The Faster Regions based Convolutional Neural Network (Faster R-CNN) is a third evolution of the network architecture and region proposal algorithm from [21] research from 2014. The original R-CNN used a region proposal algorithm, such as Selective Search [92] or Objectness [1] for detecting regions of interest (ROIs) in the image, as potential area containing the target object. These ROIs would subsequently be the input to a CNN for extracting features for classification by a support vector machine (SVM) see figure 2.15(a) for a visualization of the process. These methods of ROI extraction, which are external to the CNN are usually computationally intensive processes slowing the whole pipeline of the algorithm.

The second version [20] uses a different approach for classifying ROIs. Instead of applying a CNN on each of the region proposals, the image is inserted fully to the CNN and its output is inserted to a spatial pyramid pooling (SPP) layer [31]. Only ROIs obtained by a region proposal method are used in the SPP layer (see figure 2.15(b)). This process omits the need to perform a whole forward pass through the CNN for each ROI shortening execution time by a factor of 10 to 100X at test time and 3X at training. However, it still requires an external region proposal algorithm – a computational bottleneck.

The Faster R-CNN tackles this limitation. Furthermore, it creates a single pipeline for training. In Faster R-CNN a Region Proposal Network (RPN) is added as a part of the network's architecture (see figure 2.15(c)). The RPN is a small fully connected convolution network which slides over the output of the CNN's last layer (a feature map), searching for objects and their location in the feature map. The sliding network maps the feature map to a lower-dimensional vector, which in turn is fed into two fully connected layers. A box-regression layer *(reg)* - fitting bounding boxes around ROIs, and a box-classification layer *(cls)* - distinguishing between objects and non-objects in the ROIs. These proposed ROIs are than classified into categories. Finally, the network's outputs are predictions of bounding boxes and the category of the object inside the bounding box. These ROIs are created in a novel approach proposed in the paper. Each time the feature map is fed to the RPN, an $n \times n$ spatial window slides over the feature map and maps those windows to the *reg* and *cls* layers. Simultaneously k more region proposal (called anchors) in different sizes and shapes are

created from the center of the sliding window, in order to search for a greater variety of object's shapes and sizes. During training, the RPN minimizes two loss functions: **a)** a box-regression loss and **b)** a box-classification loss. Since the RPN is a part of the network, adding it came with little cost to execution time and training time.

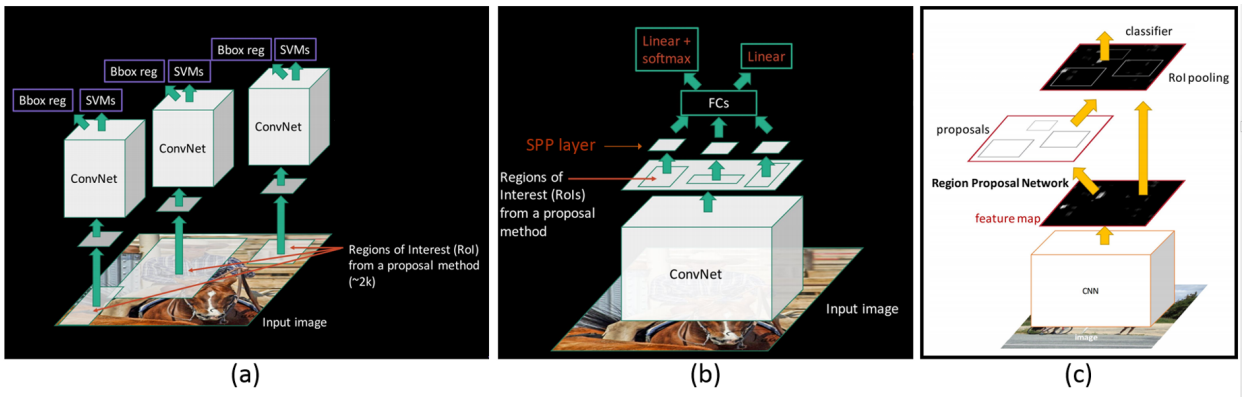

Figure. 2.15: The evolution of the Faster R-CNN algorithm: *(a)* Illustrates the feed forward pass of the first version. Notice each bounding box proposal is fed through a CNN for classification; *(b)* Illustrates the second version, where the original image is fed to one CNN and bounding boxes proposals are extracted from the last feature map of the CNN; *(c)* Is the current Faster R-CNN algorithm [21].

Faster R-CNN achieved state-of-the-art object detection accuracy on PASCAL VOC 2007 and 2012, with a mAP of 73.2% and 70.4% respectively [69]. In this thesis *an adaptation of the Faster R-CNN [69] algorithm is used in order to detect tomato flowers*.

2.4 Image Recognition in Agriculture Using Deep Learning

The fields of precision agriculture, agricultural robotics and others aimed at developing agricultural applications has seen a rise of interest in implementing deep learning methods, especially CNN based algorithms. Part of the reason is trying to replicate the successes of CNNs in other tasks such as face recognition, segmentation of biological images, traffic sign recognition and the detection of faces, text and pedestrians in natural images [49]. All these tasks had relative abundance of labeled images which is crucial for successful training of deep CNNs. In agriculture the limited amount of labeled images for specific tasks is an obstacle for using CNNs. In addition, the images acquired from the unstructured real-world agricultural conditions are usually harder to interpret using computer vision algorithms. However, some researchers invested resources and time in acquiring and labeling many images for the CNN to learn from. The main tasks researchers are using CNNs in agriculture are: fruit, vegetable or weed detection for harvesting, thinning and yield estimation and disease detection and classification for disease control and treatment. The following includes a summary of the main research for object and disease detection and classification using deep learning.

2.4.1 Disease classification research in agriculture

Produce such as fruit, vegetables and tubers can carry pathogens that can produce disease which affect processing and fresh market trade and sales [56]. Early and accurate disease detection systems can aid in avoiding such cases. Moreover, it can improve the management of the crop and can further prevent the spread of diseases [70]. Today, disease management has become even more complicated due to the global market for produce. Diseases are transferred globally more easily and new diseases occur in places where they previously were unidentified. Apparently most diseases generate some form of manifestation in the visible spectrum which can be leveraged for detection. Therefore, computer vision presents an opportunity to aid in disease detection and management [81]. A recent paper on plant stress phenotyping reviewed many researches, most of them detecting diseases in plants. However, except from one paper non used deep learning methods for the task [79]. Table 2.1 presents the main relevant disease classification research utilizing deep learning for the task.

Table. 2.1: Disease classification research in agriculture using deep learning

Ref	Produce	# Diseases	Sensors	# train images	# test images	Network architecture	Results
[81]	Pear, Apple, Peach and Grapevine leaves	13 disease classes	Images form the web	30,880	2589	AlexNet	0.963 Accuracy
[54]	Rice leaves and stems	10 diseases	Canon EOS 5D Mark III digital color camera + scanned images from a book + agricultural pest and insect pests picture database	500	-	Based on AlexNet	0.955 Accuracy
[59]	Leaves of 14 crop species	26 diseases	Plant Village dataset [37]	54,306	-	AlexNet & GooLeNet	0.993 Accuracy
[35]	Phalaenopsis seedlings (Orchid)	3 diseases	Sony XC-711 NTSC CCD Color Camera	145	144	3-Layer ANN	0.896 Accuracy

Table 2.1 shows that the state of deep learning research for plant disease classification is still at its inception, as far as this short review encompasses. The studies presented show the potential of deep learning for such a task. However, only 2 studies collected images especially for the research in conditions relevant for the task. For that reason further investigation of applying deep learning

algorithms on images from agricultural settings is important for evaluating their strength and applicability for disease classification.

2.4.2 Object detection research in agriculture

As detailed in section 2.2.2, detecting objects in agricultural environment is an initial capability for many application. Many agricultural applications such as robotic harvesters, drones for yield estimation and more are still struggling to produce sufficient detection accuracy on large and varied datasets, many performed tests on small datasets taken at a specific time of the day not resembling a common day on the farm [2, 23].

Table. 2.2: Agriculture object detection research using deep learning

Ref	Produce	Sensors	# train images	# test images	# objects	Algorithm	Results
[73]	Sweet pepper and rock melon	Multi-spectral camera, the JAI AD 130GE and Microsoft Kinect 2	209	48	-	Faster R-CNN	0.83 F1 score
[4]	Apple	PointGrey LadyBug + strobe lightning	729	112	4.5±2.9	Faster R-CNN	0.904 F1 score
	Mango	Prosilica GT3300c + strobe lightning	1154	270	5.0±3.8	Faster R-CNN	0.908 F1 score
	Almond	Handheld Canon EOS60D	385	100	7.4±5.6	Faster R-CNN	0.775 F1 score
[64]	Weeds	Multi-spectral camera, the JAI AD 130	1,600	-	-	NDVI theshold + 1 layer CNN	0.913 mAP
[65]	Wheat	consumer grade 12MP camera	415	105	4,100 ears 48,000 spikelets	Stacked hour-glass CNN [60]	0.83-0.89 for spikes 0.88-0.96 for spikelets, F1 score
[66]	Tomato	Synthetic generated images	24,000	2,400 + 100 real images	-	modified Inception-ResNet	0.91 Accuracy

Bargoti and Underwood [4] are the only researchers, as far as was found, which used a large and varied dataset, consisting of thousands of target object examples for its training and evaluation. Yet the images of apples and mangoes were acquired with sophisticated sensors and an external strobe lighting, making the detection problem less complex. Performance on almond images acquired with a simpler sensor, was accordingly significantly lower (see table 2.2).

Although deep learning research for image recognition in general and object detection in particular has shown strong results in many fields, there are some drawbacks and obstacles. The underlying theory and understanding of which architectures work better than others is not well understood. And despite progress in deep learning theory and some visualization techniques, there are many studies trying and succeeding in fooling DCNNs [61, 87]. Furthermore, because larger DCNNs trained on more data generally perform better, computational resources and the amount of labeled data have become a bottleneck for progress [27]. These drawbacks apply in the agricultural field as well, and some limitations are even more severe. For instance, acquiring enough labeled data for a specific task such as detecting fruit, can be very time and resource intensive. Moreover, the acquisition process has to be done for each new target object. Still there are some solutions to the problem, such as augmenting data through synthesizing images [6]. In conclusion, using deep learning with its limitations, has shown good results in many fields and is showing promising potential in the agricultural field as well.

3 Methodology

3.1 General

3.1.1 Problem Definition

The research goal was to implement deep Learning algorithms in object recognition tasks for agricultural applications. Two tasks were pursued, an object classification task and an object detection task, both in real-world uncontrolled lighting conditions using simple off-the-shelf RGB cameras. The classification tasks was to classify potato tubers into diseased classes according to images acquired with several RGB cameras in a warehouse. The detection task was to detect tomato flowers in images acquired in uncontrolled greenhouse conditions.

3.2 Databases

3.2.1 Image Acquisition

Images for the experiments conducted for this thesis were acquired using simple off-the-shelf RGB cameras, either smartphone cameras or simple CCD cameras (see table 3.1 for specifications). Due to their relative high quality and low cost nowadays, developing image recognition algorithms with minimal sensor hardware constraints can be beneficial for farmers worldwide [59]. Moreover, as more than 80% of the agricultural production is generated by smallholder farmers and smartphone technology is becoming prevalent worldwide, developing robust algorithms, not camera specific, for simple sensors is an important research direction [39, 38].

Table. 3.1: Sensors

Camera type	Resolution	Research used
Sony DSC-T200	3264×2448	Potato disease classification
Apple iPhone 4	960×640	Potato disease classification
Samsung Galaxy S3	720×1280	Potato disease classification
LG G4 H815	2988×5312	Tomato flower detection
Canon PowerShot 590IS	1832×3264	Tomato flower detection

Acquiring images in various illumination conditions was another principle guiding the acquisition process. Therefore, images were acquired on several different occasions and in different times of the day. Furthermore, angles and distance from target objects were not fixed or controlled.

In conclusion, the acquisition process was carried out with the goal of creating diverse and challenging datasets, so that the algorithms proposed in this thesis would generalize sufficiently well to real-world conditions.

3.2.2 Image Labeling

Labeling of ground truth instances was done using Matlab's 2014b image labeler application. Each ground truth instance was annotated using a rectangular bounding box (see figure 3.1(d)). This practice is standard in computer vision research, especially when using CNNs with square filters for the vision task. However, annotating images on the pixel-level (see figure 3.1(b)) can lead to better results in some cases. There are vision tasks where a rectangular bounding box is not a sufficient method for ground truth annotation, such as segmenting an object in an image [27]. Therefore, there is a tradeoff between the labeling time and the precision of the labeling method. Due to the big number of images used in deep learning, the vast majority of researchers use bounding box annotations, providing many more labeled examples compared to other methods. However, for some tasks a combination of methods can produce better results [63, 13].

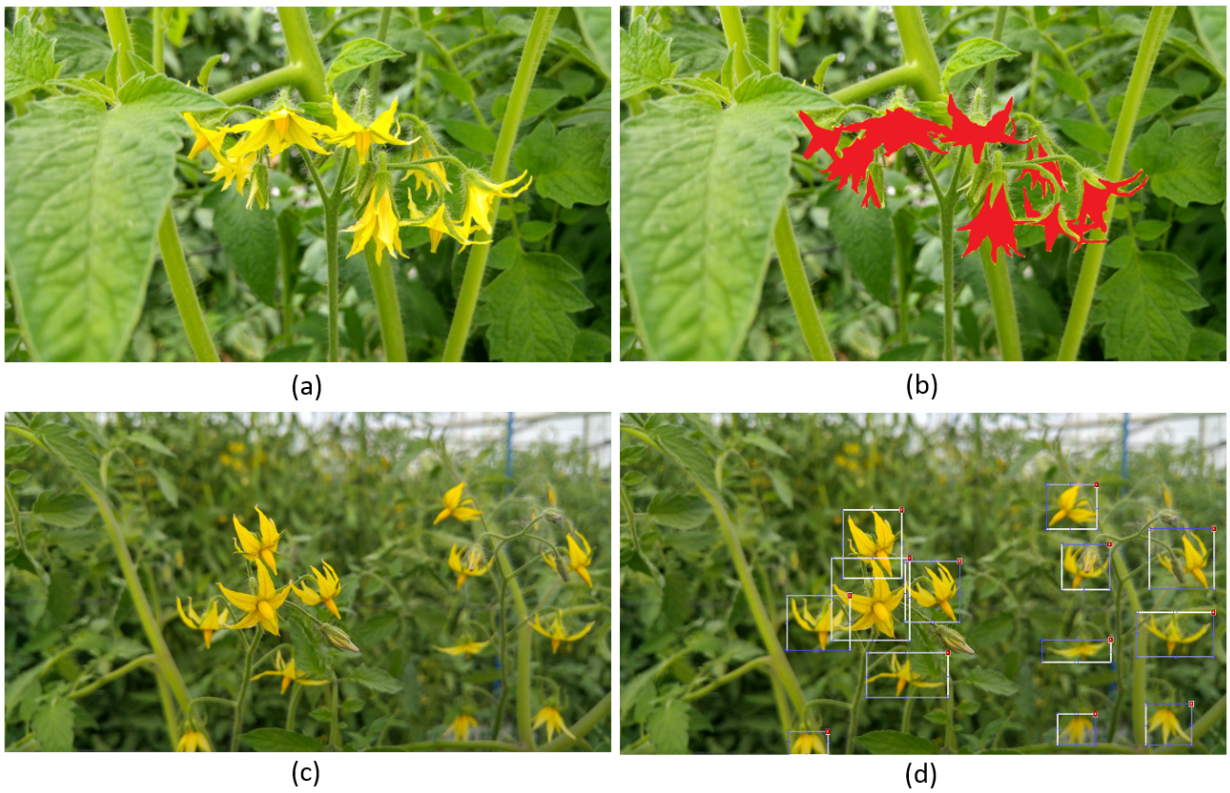

Figure. 3.1: Examples of labeling methods: *(a)* Unlabeled image; *(b)* A pixel-level labeling of image (a); *(c)* Unlabeled image; *(d)* A rectangle bounding box labeling of image (c).

3.3 Algorithm Development

3.3.1 CNN Architecture

Since training an entire CNN from scratch requires a huge amount of data and computational resources. It is common practice in the field of deep learning for object recognition to use state-of-the-art DNN models built for similar tasks, and adapt them to a specific task [25]. For example, a modern CNN can take 2-3 weeks to train across multiple GPUs on a dataset such as ImageNet. For **image classification** tasks, AlexNet is a standard CNN architecture and very influential, used as a base model in many real-world applications. However, due to implementation considerations the similar CNN-F of the Visual Geometry Group (VGG) was used for the potato disease classification task [9]. For **object detection** tasks, Faster R-CNN is a successful and influential architecture. An adaptation of the model was used for detecting tomato flowers.

Minor changes of these architectures were made in order to fit the architecture to the tasks. The last layers were changed to match the number of classes of each problem and several hyperparameters were adapted as well. Overall, the main process of fitting the model was done in the training phase.

3.3.2 Training Method

Training each model was done separately, but the method of training was similar using transfer learning. Transfer learning takes advantage of knowledge gained in previous research or problem solving and apply it to a different but related problem. In deep learning problems there are three main types of transfer learning: **a) Use a pretrained model as feature extractor** - in this method only the convolutional layers of a pretrained CNN is used. The last fully-connected layers are removed. The convolutional layers can than be used as a feature extractor for a different classifier [25]. **b) Fine-tune a pretrained model** - this method involves replacing the original classifier to a task specific classifier and retraining the network on the new dataset. It is possible to fine-tune all of the layers of the CNN or keep some layers fixed and other changeable. The idea is that lower layers learn more general features from the original dataset, so in order to fit the model to the new task only higher level features needs to be adjusted [15]. **c) Use pretrained model with no changes** - For very similar tasks, sometimes using the pretrained CNN without altering the model can be sufficient [27].

The models trained for this thesis were initially trained on the ImageNet dataset, and then fully fine-tuned for each task. ImageNet images are very diverse consisting of many categories, many of them not from the plant world [14]. Hence fine-tuning the whole network is needed.

3.4 Evaluation

The evaluation of the algorithms was done differently for each task, and is detailed in chapters 5 and 4. Yet for both the goals were similar. First, the effect of the training set size was sought, through training several models using different amount of images and testing their results on a held out test set. Evaluating the performance of each model was done according to the acceptable performance measures for each task. Accuracy and confusion matrix for the classification task and precision-recall for the detection task. Second, the best model was sought and tested similarly. See chapters 5 and 4 for specific performance measures equations and dataset divisions.

4

Identifying Potato Disease from Visual Cues using Convolutional Neural Networks

Submitted to: Computers and Electronics in Agriculture

Autors: Dor Oppenheim, Guy Shani, and Leah Tsror

Identifying Potato Disease from Visual Cues using Convolutional Neural Networks

Dor Oppenheim, Guy Shani, and Leah Tsror

***Abstract*—Many plant diseases have distinct visual symptoms which can be used to identify and classify them correctly. This paper presents a potato disease classification algorithm which leverages these distinct appearances and the recent advances in computer vision made possible by deep learning. The algorithm uses a deep convolutional neural network training it to classify the tubers into five classes, four diseases classes and a healthy potato class. The database of images used in this study, containing potatoes of different shapes, sizes and diseases, was acquired, classified, and labelled manually by experts. The models were trained over different train-test splits to better understand the amount of image data needed to apply deep learning for such classification tasks.**

Keywords*—Plant disease detection and classification, Image recognition, Convolutional neural network, Potato diseases.

I. Introduction

Potato (Solanum tuberosum) is the third most important food crop in the world, after cereals and rice. Global production exceeds 300 million metric tons and is an important nutrition and calorie provider for humanity [10]. Potato production is threatened by many diseases resulting in considerable yield losses, and/or downgrade tuber quality causing an increase of the price of potatoes [17].

Potato tubers carry many seed and soil-borne pathogens that affect quality and yield [18]. Amongst them the fungal pathogens Heminthosporium solani causal agent of silver scurf, Colletotrichum coccodes causal agent of black dot, Rhizoctonia solani causal agent of black scurf and the bacterial pathogen Streptomyces spp. the causal agent of common scab, all cause tuber blemishes and affect processing and fresh market trade and sales [8].

An early and accurate disease detection system can aid in avoiding such cases. Moreover, it can improve the management of the crop and can further prevent the spread of diseases [12]. Examination of tubers with a hand lens or microscope is required to observe the characteristic black microsclerotia of C. coccodes, or typical conidiophores and conidia of H. solani [5, 19]. However, these structures are not always present, and additional diagnostics methods must be applied (isolation on selective media, serology or molecular techniques).

D. Oppenheim is with the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel e-mail: doropp@post.bgu.ac.il.

G. Shani is with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel.

L. Tsror is with the Department of Plant Pathology, the Institute of Plant Protection, Agricultural Research Organization, Gilat Experiment Station, M.P. Negev, Israel.

Manuscript received October 29, 2017

To conclude, manually detecting and sorting potato tubers (either seed tubers or ware) is difficult, costly, and time consuming, while computerized inspection may be more efficient and cost effective.

Computer vision and machine learning techniques for disease detection have been broadly researched in the last two decades [2]. Diseases can be detected using expensive and bulky digital imaging sensors, such as spectral or near-infrared sensors. Using such sensors encumbers the widespread implementation of these methods due to its high costs and maintenance [14]. On the other hand, researchers using the visible light bandwidth, which can be captured by relatively low cost cameras, have usually focused on a single type of disease [20]. A single case identification is insufficient for real-world applications, as a single tuber can be infected by a number of diseases [4].

This paper leverages recent advances in computer vision and object recognition, for classifying multiple diseases in potatoes. In 2012 a group of researchers from Toronto won the Large Scale Visual Recognition Challenge (ILSVRC) competition by improving the classification of the ImageNet database by more than 10%. They achieved a top-5 error rate of 15.3% when using a deep Convolution Neural Network (CNN), while the second best achieved 26.2% error rate [7]. Since then, CNN methods have improved and recently the classification error dropped to 3.73% by the winning team for the same task [1]. In the field of computer vision for agricultural applications, the use of CNNs and other deep neural networks is continuously increasing [6]. A CNN was recently used for detecting and classifying seven fruits in field conditions, improving detection accuracy by 3% from previous methods [13]. CNNs used in classification tasks, such as disease classification of plant leaves or quality control of harvested fruit and vegetable, reached accuracy of more than 97% [9, 16]. In order to create successful CNNs, a large amount of training data is needed [15]. Therefore, the first aim of the current research was the collection of a sufficient dataset and classification of the displayed diseases. Results indicate a first step towards multiple disease classification for potatoes using CNNs.

II. Materials and Methods

A. Data acquisition

Photos of 400 diseased potato tubers of different cultivars, shapes, sizes and tones were acquired under normal uncontrolled illumination conditions. The tubers were manually classified by potato pathology experts as a standard procedure of monitoring the incidence and severity of various diseases in seed potato tubers prior to planting. This procedure is done

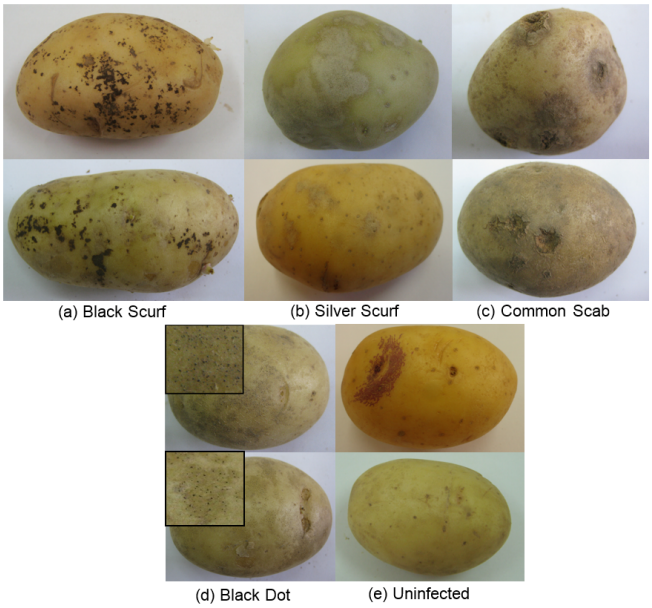

Fig. 1. Examples of visual symptoms of potato diseases: (a) Black Scurf - irregular, black, scab-like marks on the skin of the tuber. (b) Silver Scurf - circular or irregular, tan to silvery gray lesions on the tuber's skin. (c) Common Scab - circular brown rough areas, with irregular margins which can coalesce into larger areas. (d) Black Dot - tiny black dots on the skin of the tuber (hardly visible in small images). (e) Uninfected tuber.

annually independent of the current research with both seed lots imported from Europe to Israel for the spring season and domestic seed lots for the winter season [18]. The potato tubers were contaminated with several diseases simultaneously, but for the current research tubers were selected with four different diseases (separately), all with significant and typical symptoms on the tubers skin (see figure 1). The images were acquired using multiple types of standard cameras, captured from one viewpoint only. The cameras used were Sony DSC-T200, the Apple iPhone 4 camera, and the Samsung Galaxy S3 camera.

B. Data preparation

The images acquired were used to create the training and tests sets for the CNN. Every visual symptom of a disease was marked and labelled using the image labeler application in MatLab 2014b. The labelling was done with rectangular bounding boxes encompassing the visual symptom, but also much regular potato skin, as seen in figure 2. The marked areas were cropped from the original image, transformed into grayscale, and resized to a standard 224×224 pixel square. After preprocessing, a total of 2,465 patches of diseased potato skin were gathered including: 265 Black Dot patches, 469 Black Scurf patches, 686 Common Scab patches, 738 Silver Scurf patches and 307 uninfected patches.

C. Performance Measurement

The experiment was designed to evaluate the performance of the CNN's learning algorithm in classifying four diseases and uninfected potatoes. As manually labelling diseased patches of potatoes is a tedious and time costly task, an important task was to determine the minimal amount of training data that provides sufficient classification accuracy. The CNN was trained with different sizes of training sets. The smallest training set used for training was 10% of the 2,465 images, incrementally increasing by 10% to 90% of the whole dataset as detailed in table 1. In each increment the images were selected uniformly from the whole dataset. Testing of the algorithm was done on the remaining data. In total the training and testing phases was repeated 9 times over different training set sizes.

TABLE I. TRAIN AND TEST SET DIVISION

Train-Test split	Black Scurf	Common Scab	Silver Scurf	Black Dot	Uninfected
90%-10%	423/46	618/68	665/73	239/26	277/30
80%-20%	377/92	550/136	592/146	213/52	247/60
70%-30%	331/138	482/204	519/219	187/78	217/90
60%-40%	285/184	414/272	446/292	161/104	187/120
50%-50%	239/230	346/340	373/365	135/130	157/150
40%-60%	193/276	278/408	300/438	109/156	127/180
30%-70%	147/322	210/476	227/511	83/182	97/210
20%-80%	101/368	142/544	154/584	57/208	67/240
10%-90%	55/414	74/612	81/657	31/234	37/270
Total	**469**	**686**	**738**	**265**	**307**

Each training set was trained for 90 epochs, where one epoch is defined as a one full training cycle on every sample in the training set. The choice of limiting to 90 epochs was made based on empirical observations that revealed that the learning converged well within 90 epochs (as can be seen in figure 4). In order to compare between the different results over the 9 training sets, the error rate of the best scoring guess was calculated as the number of errors divided by the total number of test images in every epoch. The error was calculated both for test and train sets, in order to understand the over and under fitting of the procedure.

D. Algorithm

The algorithm chosen for the image classification task was a deep convolutional neural network (CNN). The basic architecture chosen for this problem was a CNN developed by the

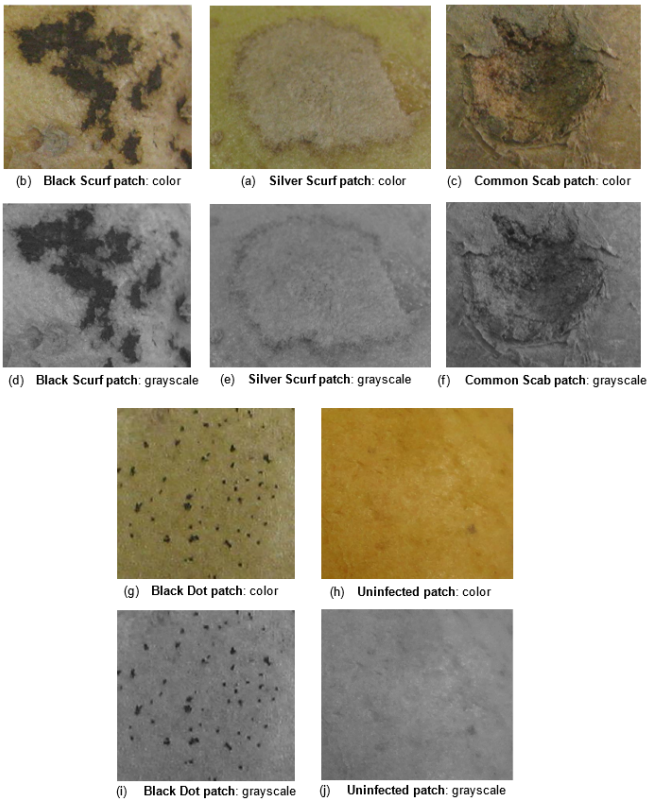

Fig. 2. Examples of diseased potato patches before and after the transformation to grayscale. (a)-(c), (g) and (h) are the original RGB images from each class. (d)-(f), (i) and (j) are the same images after the conversion to grayscale, using Matlab's rgb2gray function.

Visual Geometry Group (VGG) from the University of Oxford named CNN-F due to its faster training time [3]. Several new dropout layers were added to the VGG architecture to deal with problems of over fitting, especially due to the relatively small dataset. The required input image size for this network is a 224×224 matrix. The CNN comprises 8 learnable layers, the first 5 of which are convolutional, followed by 3 fully-connected layers and ending with a softmax layer (see figure 3). The softmax layer normalizes the input received from the final fully-connected layer (fc3) producing a distribution of values, one for each class. The sum of these values add up to 1 and they represent the probability of the input image to belong to one of the five classes. This softmax layer was also altered and adapted, reducing its size from 1,000 to 5 to fit our classification task. Training CNNs usually requires a large amount of labelled data in order to perform a good classification. Therefore, two methods were used for data augmentation; Mirroring creates additional examples by flipping the images used in training randomly. As the direction of the photos was arbitrary, mirroring the image horizontally does not change the correctness of the data; Cropping was also used, cropping the image randomly to different sizes, while keeping the cropped image minimum size to 190×190, can achieve data diversity. The use of each data augmentation method was done randomly. Before each image was inserted into the net for training it was mirrored, cropped or inserted without altering in equal distributions. Therefore, two thirds of the images trained were altered.

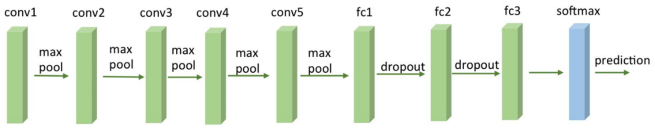

Fig. 3. A simplified model of the CNN used.

III. Results and Discussion

Results indicate, as expected, that using more data for the training phase improves the classification and reduces the error rate (see figure 4). The best trained model (trained on 90% of the dataset and tested on the remaining 10%) classified correctly 96% of the images. Results indicate that for 8 out of the 9 training sets, accuracy does not drop below 90% as the training set size decreases (Figure 4); the average difference of error rates between the best training set (90% train-10% test) and the worst training set (20% train-80% test) in these 8 sets was 5.73%. There is a significant drop in performance when the CNN was trained on 10% of the dataset and tested on 90% of it. Correct classification for this training set decreased to 83% as opposed to 90% of the classifier obtained with 20% train and 80% test. The relatively small decrease in accuracy (Table 2) for most training set sizes, is an indicator that a small amount of potato images could suffice for training a sufficiently accurate CNN.

TABLE II. Best performing model accuracy results for each train-test set

Train-Test set split	Accuracy
90%-10%	0.9585
80%-20%	0.9567
70%-30%	0.9465
60%-40%	0.9454
50%-50%	0.9069
40%-60%	0.9183
30%-70%	0.9041
20%-80%	0.9012
10%-90%	0.8321

In order to further evaluate the CNN's classification a confusion matrix was calculated. The confusion matrix's columns represent the CNN's class classification while the rows represent the actual classes. This type of representation can help evaluate the CNN's classification of each class. Figure 5 shows a confusion matrix of the best performing CNN, trained on 90% of the dataset and tested on 10%. The confusion matrix shows that the CNN classified correctly and with high accuracy infected potato tubers; 100% of the tubers which were infected with Black Dot and Black Scurf were classified correctly; over 92% of the Silver Scurf and Common Scab infected tubers were classified correctly as well. The CNN's

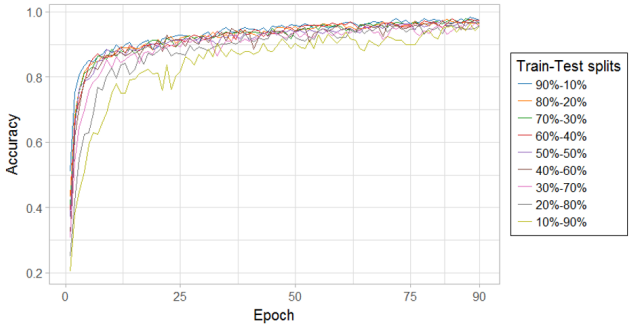

Fig. 4. Results of experiment. Plot shows the accuracy of each training set (Shown as different colors on graph) according to the epoch number.

performance dropped when classifying uninfected tubers. Most of the CNN's misclassifications occurred when classifying uninfected tubers to the disease class Silver Scurf. Silver Scurf's visual symptom are bright tan to silvery gray lesions on the tuber's skin that resemble uninfected skin. These results indicate that the trained CNN can classify correctly and accurately the four diseases presented here. However, uninfected tubers were harder to classify. The fact that the diseases were classified with high accuracy makes it suitable for a system which identification of the disease is important. Most misclassifications occurred for the non-infected class, for practical use these mistakes have less affect since planting infected tubers can spread the disease and cause considerable damage while misclassifying uninfected tubers can be solved. In this experiment only 307 images of uninfected tubers were used, increasing the amount of data of uninfected tubers can increase classification accuracy.

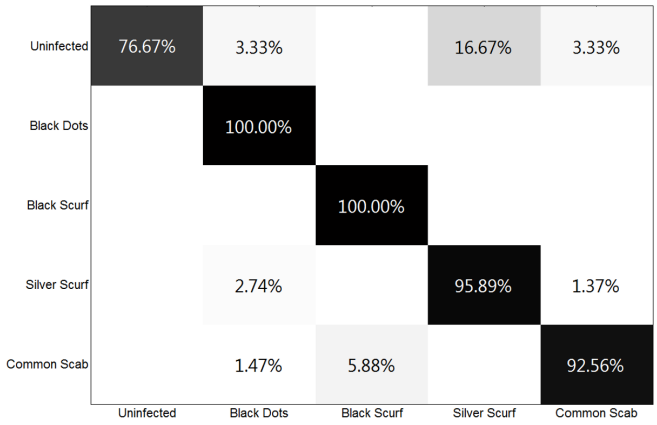

Fig. 5. - A confusion matrix of the CNN trained on 90% of the dataset and tested on the remaining 10%. Rows represent the actual classes of an image. Columns represent the CNN's class prediction. Each cell in the matrix represent the percentage of images of the row's class that were classified to the column's class.

IV. Conclusions and Future Work

The applicability of a convolution neural network in classifying image patches of diseased potato tubers into four defined disease categories and an uninfected one was examined. The 2,465 images classified by the trained CNN model varied in the acquisition device and conditions. Results indicate the robustness of the classification algorithm allowing for uncontrolled acquisition conditions. Results reveal that the correct classification of fully trained CNN models ranges from 83% for the model trained on the least amount of data, to 96%, when the model was trained on 90% of the data. To obtain classification rates higher than 90% it is sufficient to use 20% of the images (i.e., 493 images). These results further show that combining the CNN introduced here with a sliding window algorithm or an object detection network, such as faster R-CNN [11], could be utilized for classifying full images of potato tubers to different diseases with little labelling work beforehand. Ongoing research is aimed to develop a classification algorithm with an expanded number of disease classes. Acquiring data can be done easily since there are no constraints on the data acquisition.

Acknowledgment

This work is supported by the Ministry of Agriculture and partially supported by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Initiative and the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering, both at Ben-Gurion University of the Negev. We thank Professor Yael Edan for her important comments.

References

[1] Masoud Abdi and Saeid Nahavandi. "Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks". In: *arXiv preprint arXiv:1609.05672* (2016).

[2] Jayme Garcia Arnal Barbedo. "A review on the main challenges in automatic plant disease identification based on visible range images". In: *Biosystems Engineering* 144 (2016), pp. 52–60.

[3] Ken Chatfield et al. "Return of the devil in the details: Delving deep into convolutional nets". In: *arXiv preprint arXiv:1405.3531* (2014).

[4] Sergio Cubero et al. "Automated systems based on machine vision for inspecting citrus fruits from the field to postharvestffffdfffdfffda review". In: *Food and Bioprocess Technology* 9.10 (2016), pp. 1623–1639.

[5] D Errampalli, JM Saunders, and JD Holley. "Emergence of silver scurf (Helminthosporium solani) as an economically important disease of potato". In: *Plant Pathology* 50.2 (2001), pp. 141–153.

[6] A Gongal et al. "Sensors and systems for fruit detection and localization: A review". In: *Computers and Electronics in Agriculture* 116 (2015), pp. 8–19.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[8] Chakradhar Mattupalli, Ruth K Genger, and Amy O Charkowski. “Evaluating incidence of Helminthosporium solani and Colletotrichum coccodes on asymptomatic organic potatoes and screening potato lines for resistance to silver scurf”. In: *American journal of potato research* 90.4 (2013), pp. 369–377.

[9] Sharada P Mohanty, David P Hughes, and Marcel Salathé. “Using deep learning for image-based plant disease detection”. In: *Frontiers in plant science* 7 (2016).

[10] Sunil Pareek. *Postharvest Ripening Physiology of Crops*.

[11] Shaoqing Ren et al. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[12] Avery E Rich. *Potato diseases*. Academic Press, 2013.

[13] Inkyu Sa et al. “Deepfruits: A fruit detection system using deep neural networks”. In: *Sensors* 16.8 (2016), p. 1222.

[14] Sindhuja Sankaran et al. “A review of advanced techniques for detecting plant diseases”. In: *Computers and Electronics in Agriculture* 72.1 (2010), pp. 1–13.

[15] Pierre Sermanet et al. “Overfeat: Integrated recognition, localization and detection using convolutional networks”. In: *arXiv preprint arXiv:1312.6229* (2013).

[16] Wenxue Tan, Chunjiang Zhao, and Huarui Wu. “Intelligent alerting for fruit-melon lesion image based on momentum deep learning”. In: *Multimedia Tools and Applications* 75.24 (2016), pp. 16741–16761.

[17] RJ Taylor, JS Pasche, and NC Gudmestad. “Susceptibility of eight potato cultivars to tuber infection by Phytophthora erythroseptica and Pythium ultimum and its relationship to mefenoxam-mediated control of pink rot and leak”. In: *Annals of applied biology* 152.2 (2008), pp. 189–199.

[18] Leah Tsror, Orly Erlich, and Marina Hazanovsky. “Effect of Colletotrichum coccodes on potato yield, tuber quality, and stem colonization during spring and autumn”. In: *Plant disease* 83.6 (1999), pp. 561–565.

[19] Leah Tsror and Itzahk Peretz-Alon. “Control of silver scurf on potato by dusting or spraying seed tubers with fungicides before planting”. In: *American journal of potato research* 81.4 (2004), pp. 291–294.

[20] Baohua Zhang et al. “Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review”. In: *Food Research International* 62 (2014), pp. 326–343.

5

Tomato Flower Detection Using Deep Learning

Submitted to: IEEE Robotics and Automation Letters

Autors: Dor Oppenheim, Guy Shani, and Yael Edan

Tomato Flower Detection Using Deep Learning

Dor Oppenheim[1], Guy Shani[2] and Yael Edan[1]

Abstract— **Detecting objects in agricultural environments is a fundamental capability needed for automating agricultural tasks. This paper presents the adaption stages of a state-of-the-art deep learning object detection algorithm for a tomato flower detection system in a greenhouse environment. Algorithmic changes and hyper-parameters adjustments in order to succeed in the task are detailed. The algorithm used is a hybrid of a deep convolutional neural network and a region proposal network called Faster R-CNN. The diverse dataset especially acquired for this task includes 12,381 individual flowers in 771 images acquired with 2 types of cameras from different tomato cultivators and different time of day. Data augmentation techniques were used in order to increase the variety of samples the network learns from. Training and testing the model was done with datasets of various sizes aiming to reveal the effect of training set sizes on the model's performance. Results suggest that with suitable adaptations of the Faster R-CNN algorithm and sufficient image training data, a real-world detection system can be developed. The best trained detector scored an Average Precision (AP) of 0.788 which is comparable to other publically available research in agricultural object detection.**

I. INTRODUCTION

Detecting objects in an image is a crucial aspect in the development of agricultural automation applications. In order to harvest or spray selectively fruit or vegetables, determine yield, navigate in the field, objects' location in an image must be determined and objects' characteristics such as ripeness or size must be estimated [18]. Despite many years of research in agricultural oriented object detection, there are still many problems that hinder implementation of agricultural applications [10]. The highly variable and unstructured outdoor environment with changing illumination conditions, along with the complex plant structure and variable product shape and size make it hard to find a global solution to the detection of objects in the agricultural environment [18]. This paper tackles such a task, detecting tomato flowers in images taken in a greenhouse environment for a drone pollinator application. In addition to pollinating the flowers, detection of the flowers is a basic capability for many other tasks such as yield estimation [2, 6, 26], thinning [29, 42, 43] and plant phenotyping [23, 32] in general.

For years, object detection research relied mainly on hand-crafted features such as color, shape, texture or their combination [18, 17, 10]. Color, has been one of the most prominent features used in detection, despite it being affected greatly by the acquisition device, the variety of the targets color and the varying illumination conditions [10, 18]. Shape has been used mainly in detecting circular target objects such as apples [11], citrus [14] and mangoes [28], because of the relatively effective implementation of circle detection algorithms such as Circular Hough Transform. Many such studies reported that the agricultural environment characteristics were the main challenges for accurate detection and localization of the target object [10, 18, 3]. Therefore, during the years some studies tried pursuing different approaches for the detection task, such as developing adaptive thresholding algorithms to overcome the dynamic changes in illumination [40, 27] and fusing multiple sensors [40, 5, 38]. Several studies have implemented artificial neural network (ANN) algorithms [18] so as to overcome the need to define and select the features.

At first, ANN's were designed to learn to classify color features fed to the network in order to separate background pixels from target's pixels, which led to the same problems caused by the variability of the environment [38, 30, 34, 41]. Following recent advancements in computer vision algorithms accredited to deep Convolutional Neural Networks (CNNs). Performances in several classic vision tasks, such as classification [19], detection [35] and segmentation [22] have increased significantly. These networks can be fed raw data, i.e. the pixels of a flower in the image, and learn features automatically from it, avoiding the need for hand-crafted features and the challenges they bring about [20]. With enough data, good representations of the target object can be learned and a robust system can be created, able to cope with real-world vision problems which are inherently characterized by uncertainty and large amounts of variability [12]. These networks have demonstrated high performance results in the agricultural domain as well, in tasks such as disease detection and classification [21, 24, 31], fruit detection and localization [4, 36] and more. Object detection research using deep CNNs is depicted in table I.

In this paper an adaptation of the Faster R-CNN algorithm is explored for a tomato flower detection task. Algorithmic changes as detailed in this paper were implemented and hyper-parameters were adjusted in order to succeed in the task. Image data was collected and manually labeled by hand, specifically for the drone pollinator research. 12,381 flowers were tagged in 771 images. The best trained detector scored an Average Precision (AP) of 0.788 which is comparable to other publically available research in agricultural object detection.

[1]Dor Oppenheim and Yael Edan are with the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel doropp@post.bgu.ac.il, yael@bgu.ac.il

[2]Guy Shani is with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel shanigu@bgu.ac.il

TABLE I
OBJECT DETECTION RESEARCH USING DEEP LEARNING

Object detected	Detection algorithm	Results	Ref
Sweet pepper & rock melon	Faster R-CNN	0.83 F1 score	**[36]**
Apple	Faster R-CNN	0.904 F1 score	**[4]**
Mango	Faster R-CNN	0.908 F1 score	**[4]**
Almond	Faster R-CNN	0.775 F1 score	**[4]**
Weeds	NDVI theshold + 1 layer CNN	0.913 mAP	**[31]**
Wheat	Stacked hourglass CNN [25]	0.83-0.89 for spikes 0.88-0.96 for spikelets, F1 score	**[32]**
Tomato	modified Inception-ResNet	0.91 Accuracy	**[33]**

II. BACKGROUND & RELATED WORK

A. Faster R-CNN overview

The algorithm chosen for the detection task was a Faster Regions based Convolutional Neural Network (Faster R-CNN). The Faster R-CNN is a third evolution of the network architecture and region proposal algorithm [9]. The original R-CNN used a region proposal algorithm, such as Selective Search [39] or Objectness [1] for detecting regions of interest (ROIs) in the image, as potential area containing the target object. These ROIs would subsequently be the input to a CNN for extracting features for classification by a support vector machine (SVM) see Fig. 4a for a visualization of the process. These methods of ROI extraction, which are external to the CNN are usually computationally intensive processes slowing the whole algorithm pipeline. The second version [8] uses a different approach for classifying ROIs. Instead of applying a CNN on each of the region proposals, the image is inserted fully to the CNN and its output is inserted to a spatial pyramid pooling (SPP) layer proposed by [16]. Only ROIs obtained by a region proposal method are used in the SPP layer (see Fig. 1(a)). This process omits the need to perform a whole forward pass through the CNN for each ROI shortening execution time by 10 to 100X at test time and 3X at training. However, it still requires an external region proposal algorithm a computational bottleneck.

This version of the algorithm tackles this limitation. Furthermore, it creates a single pipeline for training. In Faster R-CNN a Region Proposal Network (RPN) is added as a part of the network's architecture (see Fig. 1(c)). The RPN is a small fully connected convolution network which slides over the output of the CNN's last layer (a feature map), searching for objects and their location in the feature map. The sliding network maps the feature map to a lower-dimensional vector, which in turn is fed into two fully connected layers. A box-regression layer (*reg*) - fitting bounding boxes around ROIs, and a box-classification layer (*cls*) - distinguishing between objects and non-objects in the ROIs. These proposed ROIs are than classified into categories. Finally, the network's outputs are predictions of bounding boxes and the category of the object inside the bounding box. These ROIs are created in a novel approach proposed in the paper. Each time the feature map is fed to the RPN, a $n \times n$ spatial window slides over the feature map and maps those windows to the *reg* and *cls* layers. Simultaneously k more region proposals (anchors) in different sizes and shapes are created from the center of the sliding window, in order to search for a greater variety of object's shapes and sizes.

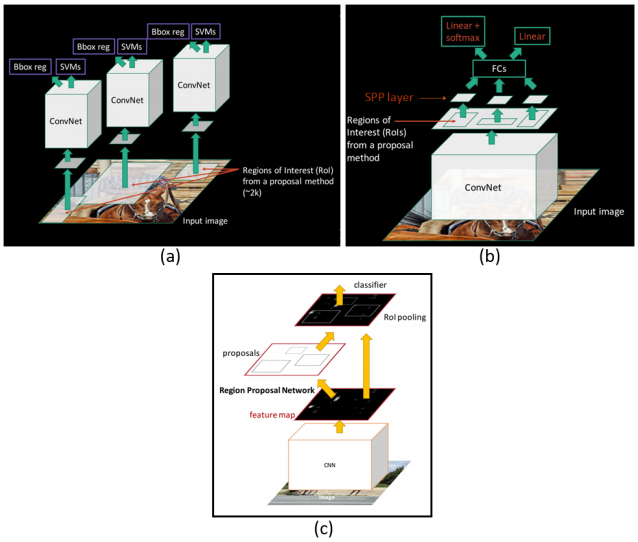

Fig. 1. The evolution of the Faster R-CNN algorithm. *(a)* Illustrates the feed forward pass of the first version. *(b)* Illustrates the second version. *(c)* Is the current Faster R-CNN algorithm.

During training, the RPN minimizes two loss functions: 1) a box-regression loss and 2) a box-classification loss. Since the RPN is a part of the network, adding it came with little cost to execution time and training time.

III. METHODOLOGY

A. Data acquisition

Image data for this research was collected in purpose to create a diverse dataset of tomato flowers, in order to create a robust detection algorithm. 771 color images of tomato flowers were acquired in a high end greenhouse in Israel on the 26th of April 2016 at three time intervals along that day, morning between 8:00AM to 9:00AM, noon between 11:30AM to 12:30AM and afternoon between 17:00PM to 18:00PM. These time intervals affect the illumination conditions in the greenhouse and were chosen in order to create diverse images. The images contained 12,183 individual

Fig. 2. Example images of tomato flowers acquired in a greenhouse in Israel. Notice the wide variety of flower sizes, hue differences and varying lighting conditions.

flowers from various cultivars of tomatoes. To minimize the effect of the acquisition device on the algorithm's performance, images were acquired using two different cameras. 466 images were acquired using a Canon PowerShot 590IS camera, and saved in a .jpg format with a resolution of 1832×3264. The remaining 305 images were acquired using the camera of a LG-G4 smartphone, and saved in a .jpg format as well in resolution of 2988×5312. Images were taken at a random distance from between 0.2m and 0.8m, within the plant's rows, and from various angles. The varying distances, time of acquisition, acquisition device, tomato flower's cultivars, hue and background created a diversified dataset of tomato flowers which differed in size, hue, shape and illumination conditions (see Fig. 2 for representative examples). The variation of the dataset intended to ensure the modelling of a robust and accurate detection algorithm that would generalize well in the complex and uncertain environmental conditions.

B. Data preparation

The images acquired were used to create the training and test sets for the Faster R-CNN algorithm. Every tomato flower in the images was marked using the image labeler application in MatLab 2017a. Very small flowers, usually from a different plant row were not labeled (see Fig. 3). Flowers were labeled with rectangular bounding boxes encompassing the flower. 12,183 flowers were manually labeled in total by three human labelers, each labeling a third of the database serving as positive instances, while the negative instances were extracted from the non-marked area of the images. After marking all of the flowers in the images the marked dataset was divided randomly into a training set, a validation set and a test set. The training set consisted of 80% of the images and the test set consisted of the remaining 20%. These division ratios are standard practice in machine learning research [13].

Initial experimentation with training the Faster R-CNN resulted in very low accuracy and average precision, due to the ratio between the size of images acquired and the flowers in the image. Images were large and the flowers very small (132×135 on average). The implementation of the Faster

Fig. 3. Examples of the labeling of flowers with Matlab's image labeler.

R-CNN presented by [35] resized images so that the shorter side of the image would be 600 pixels (the longer side was resized so that the image maintained the same aspect ratio), because of hardware constraints. This caused the flowers in the resized image to be too small for the algorithm to extract meaningful features. Therefore, images were cropped into 3×3 parts, so resizing won't affect flowers in the image as much and images will be similar to those in the PASCAL-VOC dataset (the dataset on which the Faster R-CNN was designed and trained). Cropping was done with padding in order to prevent losing labeled flowers.

C. Evaluation

The experiment was designed to evaluate the performance of the Faster R-CNN's learning algorithm in detecting tomato flowers in an unstructured and uncontrolled greenhouse environment. As manually labeling tomato flowers is a tedious and time costly task, determining the minimal amount of training data that provides sufficient detection accuracy is important. Therefore, the Faster R-CNN was trained with increasing sizes of training sets (see Table II for details). Training duration was determined by a validation set which was obtained from the training set by holding out an additional 20% of randomly selected images. Training was considered done when a convergence was detected in the validation set results or after 50,000 iterations. An iteration was considered as a full run of the learning algorithm over one image. Each trained model was then tested on the same held out test set for model evaluation. Accounting for variance was done by repeating training 5 times over the smallest dataset due to the long training times. Fully training a model requires approximately 15 hours on a PC equipped with an Intel core i7-6700, 64-bit quad-core 3.4GHz CPU, an NVIDIA GeForce GTX TITAN X and 32GM memory running on Microsoft Windows 10 system.

D. Performance Measurement

The algorithm's detection performance was evaluated using the recall and precision indicators. Precision indicates the fraction of the algorithm's predictions that are flowers (1). Recall is the fraction of flowers in the images that were detected by the algorithm (2). TP in the equations refer to true positive, which specify if a flower in the image was

TABLE II
EXPERIMENT DATASETS SIZES FOR BENCHMARKING EVALUATION

Set #	% training set	# train	# validation
1	20	1616	400
2	40	3227	806
3	60	4840	1209
4	80	6453	1613
5	100	8067	2016

detected by the algorithm. FP refers to false positive, which specifies an algorithm's mistake of predicting background as a flower. Lastly, FN refers to false negative, which specify flowers not detected by the algorithm. Precision and recall are commonly visualized using a Precision-Recall curve.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

From these indicators average precision (AP) is calculated (3). AP is a summary of the precision-recall curve and is calculated as a weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$AP = \sum_{n} (recall_n - recall_{n-1}) precision_n \tag{3}$$

This scalar score is a conventional measurement for detection tasks as it provides information about the accuracy of the algorithm in addition to a consideration to misclassifications or missed predictions [46]. The AP measurement has been used in the Pascal Visual Object Classes (VOC) challenge which is a benchmark in visual object category recognition and detection [7] and other benchmark competitions as well.

IV. ALGORITHM

A. Faster R-CNN implemetation details

The implementation of the Faster R-CNN for this research followed the guidelines in the original paper [35]. However, some algorithmic changes had to be made so the Faster R-CNN design would be suitable for the tomato flower detection task. In the original paper two networks are used and compared, the VGG16 network [37] and the ZF network [45]. The GPU used in this research did not have enough memory for implementing the VGG16, which is a very large network, so we used a residual neural network called ResNet50. This network has 50 layers, deeper than the VGG16's 16 layers, however due to the network's innovative design, it still requires less parameters and is less complex. The core idea of ResNets is to create identity shortcut connections between layers that allows skipping one or more layers [15]. Based on the deep residual network framework, a similar network won the first place in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and the Common Objects in Context (COCO) 2015 in several computer vision detection and classification tasks [15].

Data augmentation techniques were used in order to increase the variety of samples the network learns from. This technique has been proven useful in improving neural network performance and generalization (Krizhevsky et al. 2012). Two simple methods were used 20% of the time: 1) images were flipped horizontally 2) images were randomly rotated left or right in 90. More adaptations were made to deal with the difference in the average object sizes between the PASCAL-VOC and the tomato flowers database [7]. The anchor parameters provided initial guesses of the object's sizes in the images. For the PASCAL-VOC initial anchor area sizes were 128^2, 256^2 and 512^2 and aspect ratios were 1:1, 1:2 and 2:1, however these appeared unsuitable for tomato flowers, since they tend to appear smaller and in bigger numbers in our images. For these reasons we chose the initial anchor area sizes to be 64^2, 128^2 and 256^2 and aspect ratios of 1:1, 1:1.5 and 1.5:1. In addition, the number of bounding boxes proposed were increased from 300 to 2000 during training. This generates many bounding boxes proposals, some of them overlap. To reduce redundancy, a non-maximum suppression (NMS) algorithm is adopted on the proposals based on their *cls* layer output score. The threshold for the NMS is chosen to be 0.7. During testing the number of proposals was reduced to 500 so as to speed up execution time.

An important parameter is Intersection over Union (IoU). The IoU is a metric for deciding if an ROI proposal is an object. It is calculated as the overlapping ratio between a proposed bounding box to the ground truth bounding box. IoU scores above 0.6 were considered as a detection of flowers, while smaller IoU was considered as background. In general IoU above 0.5 is considered a fairly good detection.

V. RESULTS ANS DISCUSSION

Results indicate that detecting tomato flowers in a greenhouse environment through the Faster R-CNN deep learning framework can lead to reasonable performance, with the right adaptations. Best AP was achieved on the dataset with 80% of image data (see table III).

TABLE III
EXPERIMENT DATASETS RESULTS

Set #	% training set	AP
1	20	0.773
2	40	0.747
3	60	0.786
4	80	0.788
5	100	0.784

However, the differences in AP between all of the datasets used in the study were not significant, see precision-recall

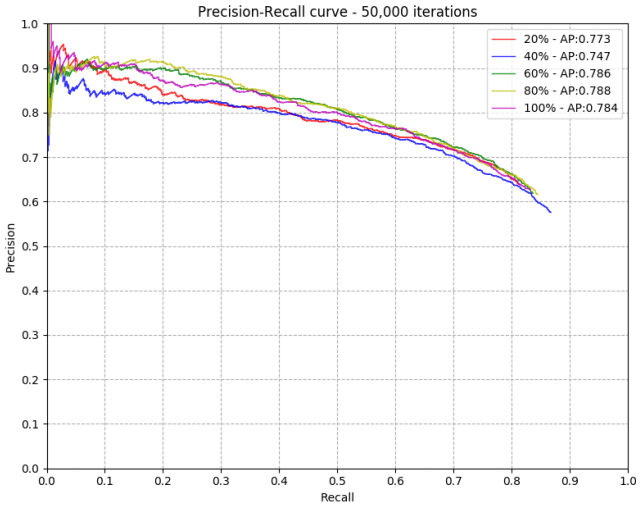

Fig. 4. Precision-Recall curve from the experiment's results.

curve in Fig. 4 for details. These results are derived from the test set of 424 test images containing 2137 tomato flowers. More images for training was expected to yield better results in test time. Here we see that it was not necessarily the case. Best results were achieved with 80% of the images, second best results were 60% and when using 100% of the images third best results were achieved. A possible explanation could be that there were enough examples for the model to generalize well with the data augmentation techniques. Even in the smallest dataset consisting of 2000 images achieved good and comparable results to the bigger datasets. Cross-validation on the best performing model (trained on 80% of the data) was repeated 5 times to account for variance in the training data, where each time 6,453 random images were sampled from the training set without replacement.

Studying test images results provides insight about the algorithm's mistakes and of the training set data labelling process. The labelling task is tedious and prone to errors. Unintentionally missing positive examples or labelling erroneous examples occur and can affect the learning phase and test results greatly. In some cases tomato flower examples were missed and on others very small tomato flowers from another plant row were labeled. Occluded and overlapping flowers were especially challenging, due to each of the human labeler's interpretation of when an occluded flower should be labeled and the challenge of labeling highly overlapping flowers (see Fig. 5 for examples). There are several known methods to reduce labeling errors, such as labeling each image by a number of human labelers, incorporating a consensus voting scheme. In addition, more rigorous and accurate labeling would have helped. However, these methods are usually expensive and time consuming.

More mistakes can be a consequence of the tomato flower's size. Flowers that were too large or too small weren't detected. A probable cause for these mistakes is the anchor size and aspect ratios hyper-parameters. These anchors were set in three different sizes and aspect ratios, which are supposed to cover most of the flower's sizes in the image. However, some outlier flowers are missed. Expanding the number of anchors and aspect ratios can increase accuracy, with the cost of computational speed.

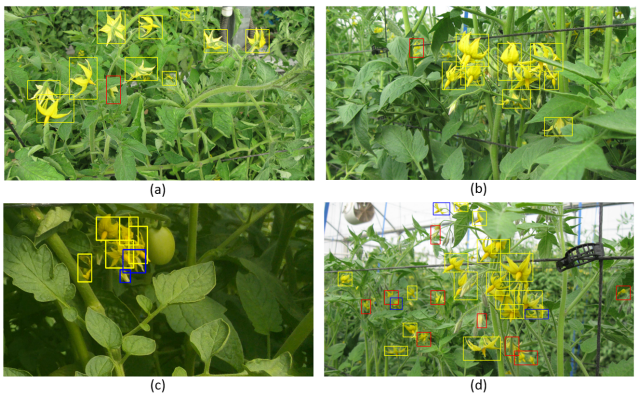

Fig. 5. Examples of Faster R-CNN algorithm's prediction output. Yellow bounding boxes represents a correct prediction (True positives). Red bounding boxes represent erroneous predictions of the algorithm (False positives). Lastly, blue bounding boxes represent misses of the algorithm, ground truth flowers that were not detected by the algorithm (False negatives). Image *(a)* shows an example of a flower that wasn't labeled properly. Alternatively, the human labeler did not consider the occluded flower as a flower. *(b)* is an example of a flower or a couple of flowers from a different row mistakenly predicted as flower by the algorithm. *(c)* is an example of highly overlapping flowers, the algorithm missed one flower, the top blue box. The bottom blue box is an example of an ambivalent case, is a flower bud a flower? *(d)* an example of an image with many such mistakes similar to *(a)*, *(b)* and *(c)*.

VI. Conclusions and Future Work

This paper presents a tomato flower detection system for images acquired in greenhouse settings using the Faster R-CNN deep learning framework. Results indicate that detecting tomato flowers in the challenging agricultural environment can be done with reasonable accuracy. Through careful hyper-parameter settings a detector can be trained to cope with the large variation, uncertainty and lack of structure of the agricultural environment. Using this technique image acquisition can be done with simple cameras in uncontrolled conditions. Furthermore, there was no need to predefine or select best-fit features. Average Precision of the best model trained was 0.788, comparable to other benchmark datasets.

Transfer learning enabled the Faster R-CNN to learn from a relatively small amount of image examples to detect tomato flowers. The model trained on the smallest dataset in this paper got similar results to the model trained on the largest dataset, which suggests that good results can be acquired using fairly small datasets, with careful image acquisition and data augmentation techniques.

The fact that the Faster R-CNN detector operates as a black box after training is one of the causes of criticism of deep neural network research and applications. However, careful investigation of the results and new tools such as visualizing the activations and features produced by each layer [44] can help acquire more insight into the algorithms classification and detection process and can help improve and fine-tune it.

Future work should adjust the algorithm to real-time execution speed and test it in a greenhouse environment. In addition developing an algorithm that can deal with objects in a larger variety of sizes would be very helpful for fast implementation and better performance detectors.

ACKNOWLEDGMENT

This work was supported by by Ben-Gurion University of the Negev through the Helmsley Charitable Trust, the Agricultural, Biological and Cognitive Robotics Initiative, the Marcus Endowment Fund, and the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering. We thank Liron Dadon and Tomer Wizman for their help in the labeling process.

REFERENCES

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. "Measuring the objectness of image windows". In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2189–2202.

[2] Arturo Aquino, Borja Millan, Salvador Gutiérrez, and Javier Tardáguila. "Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis". In: *Computers and Electronics in Agriculture* 119 (2015), pp. 92–104.

[3] C Wouter Bac, Eldert J Henten, Jochen Hemming, and Yael Edan. "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead". In: *Journal of Field Robotics* 31.6 (2014), pp. 888–911.

[4] Suchet Bargoti and James Underwood. "Deep fruit detection in orchards". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 3626–3633.

[5] DM Bulanon, TF Burks, and V Alchanatis. "Image fusion of visible and thermal images for fruit detection". In: *Biosystems Engineering* 103.1 (2009), pp. 12–22.

[6] Maria P Diago, Andres Sanz-Garcia, Borja Millan, Jose Blasco, and Javier Tardaguila. "Assessment of flower number per inflorescence in grapevine by image analysis under field conditions". In: *Journal of the Science of Food and Agriculture* 94.10 (2014), pp. 1981–1987.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[8] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[10] A Gongal, S Amatya, Manoj Karkee, Q Zhang, and K Lewis. "Sensors and systems for fruit detection and localization: A review". In: *Computers and Electronics in Agriculture* 116 (2015), pp. 8–19.

[11] A Gongal, Abhisesh Silwal, S Amatya, Manoj Karkee, Q Zhang, and K Lewis. "Apple crop-load estimation with over-the-row machine vision system". In: *Computers and Electronics in Agriculture* 120 (2016), pp. 26–35.

[12] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. "Deep learning for visual understanding: A review". In: *Neurocomputing* 187 (2016), pp. 27–48.

[13] Isabelle Guyon. "A scaling law for the validation-set training-set size ratio". In: *AT&T Bell Laboratories* (1997), pp. 1–11.

[14] MW Hannan, TF Burks, and Duke M Bulanon. "A machine vision algorithm combining adaptive segmentation and shape analysis for orange fruit detection". In: *Agricultural Engineering International: CIGR Journal* (2010).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *European Conference on Computer Vision*. Springer. 2014, pp. 346–361.

[17] AR Jimenez, R Ceres, and JL Pons. "A survey of computer vision methods for locating fruit on trees". In: *Transactions of the ASAE* 43.6 (2000), p. 1911.

[18] Keren Kapach, Ehud Barnea, Rotem Mairon, Yael Edan, and Ohad Ben-Shahar. "Computer vision for fruit harvesting robots–state of the art and challenges ahead". In: *International Journal of Computational Vision and Robotics* 3.1-2 (2012), pp. 4–34.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[21] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. "Deep-plant: Plant identification with convolutional neural networks". In: *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 452–456.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[23] Jennifer Mack, Christian Lenz, Johannes Teutrine, and Volker Steinhage. "High-precision 3D detection and

reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning". In: *Computers and Electronics in Agriculture* 135 (2017), pp. 300–311.

[24] Sharada P Mohanty, David P Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection". In: *Frontiers in plant science* 7 (2016).

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *European Conference on Computer Vision*. Springer. 2016, pp. 483–499.

[26] Stephen Nuske, Kyle Wilshusen, Supreeth Achar, Luke Yoder, Srinivasa Narasimhan, and Sanjiv Singh. "Automated visual yield estimation in vineyards". In: *Journal of Field Robotics* 31.5 (2014), pp. 837–860.

[27] Ahmad Ostovar, Ola Ringdahl, and Thomas Hellström. "Adaptive Image Thresholding of Yellow Peppers for a Harvesting Robot". In: *Robotics* 7.1 (2018), p. 11.

[28] A Payne, K Walsh, Phul Subedi, and Dennis Jarvis. "Estimating mango crop yield using image analysis using fruit at 'stone hardening' stage and night time imaging". In: *Computers and Electronics in Agriculture* 100 (2014), pp. 160–167.

[29] Alison B Payne, Kerry B Walsh, PP Subedi, and Dennis Jarvis. "Estimation of mango crop yield using image analysis–segmentation method". In: *Computers and electronics in agriculture* 91 (2013), pp. 57–64.

[30] Alessio Plebe and Giorgio Grasso. "Localization of spherical fruits for robotic harvesting". In: *Machine Vision and Applications* 13.2 (2001), pp. 70–79.

[31] Ciro Potena, Daniele Nardi, and Alberto Pretto. "Fast and accurate crop and weed identification with summarized train sets for precision agriculture". In: *International Conference on Intelligent Autonomous Systems*. Springer. 2016, pp. 105–121.

[32] Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. "Deep learning for multi-task plant phenotyping". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2055–2063.

[33] Maryam Rahnemoonfar and Clay Sheppard. "Real-time yield estimation based on deep learning". In: *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II*. Vol. 10218. International Society for Optics and Photonics. 2017, p. 1021809.

[34] Murali Regunathan and Won Suk Lee. "Citrus fruit identification and size determination using machine vision and ultrasonic sensors". In: *2005 ASAE Annual Meeting*. American Society of Agricultural and Biological Engineers. 2005, p. 1.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[36] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. "Deepfruits: A fruit detection system using deep neural networks". In: *Sensors* 16.8 (2016), p. 1222.

[37] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[38] Yongting Tao and Jun Zhou. "Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking". In: *Computers and Electronics in Agriculture* 142 (2017), pp. 388–396.

[39] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective search for object recognition". In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[40] Efi Vitzrabin and Yael Edan. "Adaptive thresholding with fusion using a RGBD sensor for red sweet-pepper detection". In: *Biosystems Engineering* 146 (2016), pp. 45–56.

[41] Juan P Wachs, H I Stern Stern, T Burks, and V Alchanatis. "Low and high-level visual feature-based apple detection from multi-modal images". In: *Precision Agriculture* 11.6 (2010), pp. 717–735.

[42] Zhenglin Wang, Brijesh Verma, Kerry B Walsh, Phul Subedi, and Anand Koirala. "Automated mango flowering assessment via refinement segmentation". In: *Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on*. IEEE. 2016, pp. 1–6.

[43] Niels Wouters, Bart De Ketelaere, Tom Deckers, Josse De Baerdemaeker, and Wouter Saeys. "Multispectral detection of floral buds for automated thinning of pear". In: *Computers and Electronics in Agriculture* 113 (2015), pp. 93–103.

[44] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579* (2015).

[45] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[46] Mu Zhu. "Recall, precision and average precision". In: *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo* 2 (2004), p. 30.

6 Conclusions and Future Research

6.1 Summary and Conclusions

This research investigated the development of deep learning algorithms for image recognition problems in the agricultural field. The unstructured and complex agricultural environment and products, makes developing computer vision applications for automation of agricultural tasks or enhancement of farm management a challenge. These conditions, amongst others, have hindered integration of many technological advancements in the agricultural sector.

Success of deep learning in solving problems in other fields has been abundant in recent years. However, due to certain factors, deep learning methods have hardly been utilized in agriculture. Therefore, this research explored the procedure of adapting the deep learning framework for two agricultural tasks. Both consists of image recognition problems solved by utilizing deep convolutional neural network architectures, with adequate adaptations to suit each task.

Training data was acquired and labeled for this research from different agricultural settings using various off-the-shelf cameras, with the intention of creating diverse datasets for the learning algorithm and for testing the generalization of them. Contrary to many researches which use high-end sensors for image acquisition, using off-the-shelf was meant to examine if the strength of deep learning algorithm could compensate the quality of the acquisition devices. This could enable a more widespread use of the algorithms developed, due to the lack of dependence on costly hardware, and relying on more sophisticated software.

The first task was classifying potato tuber diseases from RGB images for a sorting application. A convolutional neural network was used for classifying each image into one of four classes of diseases or to a healthy potato tuber class. Due to the complex geometry and colors of the diseases' visual appearances as well as the uneven lighting conditions, classifying the 2,465 images through hand-engineered features would have been an impractical effort. Utilizing CNN's ability of learning high level features through their hierarchical structure yielded high levels of classification accuracy, 95.85% of the images were classified correctly into the five classes, while training on the largest dataset consisting of 2172 images. Moreover, not a single image with a disease category was classified as a healthy potato tuber, an important component of a disease sorting application. Another aspect of the research was to determine how and to what extent, the amount of training data affected the performance of the model. As expected the more data used for training, the better

the results were, but training on 20% of the data already achieved reasonable performance of above 90% accuracy.

The second task was to detect tomato flowers in a greenhouse environment for a drone pollinator. A modified CNN was used for this task called Faster R-CNN designed by Ren et al. [69]. This network is an adapted CNN designed to detect and classify objects in images. It is a combination of a CNN and a region proposal network forming together a single framework for object detection. Detecting objects in images presents an additional challenge to classification, since first objects have to be localized in the image, prior to classifying them. Several different studies tackled this problem using deep learning algorithms, yet as far as I know, most used sophisticated sensors such as multispectral and depth cameras providing additional information on the environment, or artificial illumination producing images with more even lighting conditions. In this research simple off-the-shelf cameras were used for the task, investigating the feasibility of such systems. The database created consists of 771 images containing 12,183 annotated single tomato flowers. The effect of the amount of training data was tested in this study as well, but no apparent impact on the results was found, probably due to the test dataset division. The smallest dataset included 2,000 tomato flowers seemingly consisting of enough data for the learning algorithm to create a good representation of the tomato flower. Mean average precision of the best trained model was 0.788, slightly better than other benchmark datasets for detection. Additional insight about the annotation process was discovered while examining test's results. As can be anticipated, the precision of labeling flowers in addition to a coordinated agreement of the human labelers as to what is considered a flower, is essential for improving detection accuracy.

The deep learning frameworks performed well in both cases exceeding results in similar studies. Despite there not being any other studies with the same objectives for the same crops using deep learning. These results provide a proof of concept for implementing deep learning algorithms for detection and classification tasks in agricultural settings.

The main conclusions from this study are that utilizing deep learning in agriculture can have good results, when provided with sufficient and varied amount of data. This is quite obvious, however for both tasks 20% of the collected data was sufficient to provide reasonable results. Meaning, that with a few hours of acquiring images and labeling the ground truth instances a satisfactory sized dataset can be established. Furthermore, the fact that the databases for this thesis were acquired with simple sensors hint that while using deep learning algorithms there isn't hard constraints on acquisition devices. I believe that this work succeeded in providing sufficient evidence of the potential of deep learning algorithms to solve computer vision problems to accelerate technological advancements in the agricultural sector, providing another tool to help in automation and improved management of farms and agricultural land.

6.2 Future Research

Some research areas remain open for future expansion of this work.

On-line implementation Although images were acquired in the field, analysis and performance measurements were done off-line in the lab. In order to fully appreciate the capabilities of these algorithms, it is required to adapt them so they can perform on-line with the constraints and demands of agricultural conditions.

Training data size effect This research presented only empirical results of how training data affected performance of the algorithms. Nevertheless, for practical use, it would be beneficial to know before acquiring images, how much should be acquired. This can be done by repeating some of the experiments with different training set divisions and further statistical analysis.

Labeling quality effect While examining the predictions in images from the test set of the tomato flower dataset. It was noticeable that inconsistencies in some of the labeling of ground truth instances worsened results. For example, in some images, small flowers from a different row of plants were labeled as flowers and in some they did not, causing the model to sometimes detect flowers in the image which were not labeled. Another example, is inconsistencies about when is a partially occluded flower is considered a flower and when not. This raises a question about the trade-off between quality and speed. Determining an optimal point could be an important factor for applications that need an active acquisition and labeling of data.

Bibliography

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. “Measuring the objectness of image windows”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2189–2202 (cit. on p. 24).

[2] C Wouter Bac, Eldert J Henten, Jochen Hemming, and Yael Edan. “Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead”. In: *Journal of Field Robotics* 31.6 (2014), pp. 888–911 (cit. on pp. 1, 2, 8, 27).

[3] Dana H Ballard. “Generalizing the Hough transform to detect arbitrary shapes”. In: *Readings in computer vision*. Elsevier, 1987, pp. 714–725 (cit. on p. 6).

[4] Suchet Bargoti and James Underwood. “Deep fruit detection in orchards”. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 3626–3633 (cit. on pp. 1, 2, 27, 28).

[5] Horace B Barlow. “Single units and sensation: a neuron doctrine for perceptual psychology?” In: *Perception* 1.4 (1972), pp. 371–394 (cit. on p. 8).

[6] R Barth, J IJsselmuiden, J Hemming, and EJ Van Henten. “Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset”. In: *Computers and Electronics in Agriculture* 144 (2018), pp. 284–296 (cit. on p. 28).

[7] Yoshua Bengio. “Learning deep architectures for AI”. In: *Foundations and trends® in Machine Learning* 2.1 (2009), pp. 1–127 (cit. on pp. 8, 9, 13).

[8] Jonathan M. Blackledge. *Digital Image Processing: Mathematical and Computational Methods*. Woodhead Publishing Series in Electronic and Optical Materials. Elsevier Science, 2005 (cit. on p. 6).

[9] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Return of the devil in the details: Delving deep into convolutional nets”. In: *arXiv preprint arXiv:1405.3531* (2014) (cit. on p. 31).

[10] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. “BING: Binarized normed gradients for objectness estimation at 300fps”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 3286–3293 (cit. on p. 23).

[11] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. “cuDNN: Efficient Primitives for Deep Learning”. In: *CoRR* abs/1410.0759 (2014). arXiv: 1410.0759 (cit. on p. 10).

[12] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of Control, Signals, and Systems (MCSS)* 2.4 (1989), pp. 303–314 (cit. on p. 12).

[13] Jifeng Dai, Kaiming He, and Jian Sun. “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1635–1643 (cit. on p. 30).

[14]Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255 (cit. on p. 31).

[15]Li Deng and Dong Yu. “Deep learning: methods and applications”. In: *Foundations and Trends® in Signal Processing* 7.3–4 (2014), pp. 197–387 (cit. on p. 31).

[16]Ronen Eldan and Ohad Shamir. “The power of depth for feedforward neural networks”. In: *Conference on Learning Theory*. 2016, pp. 907–940 (cit. on p. 21).

[17]Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes (voc) challenge”. In: *International journal of computer vision* 88.2 (2010), pp. 303–338 (cit. on p. 23).

[18]Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645 (cit. on p. 23).

[19]Kunihiko Fukushima and Sei Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285 (cit. on p. 16).

[20]Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448 (cit. on p. 24).

[21]Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (cit. on pp. 23–25).

[22]Juliana Freitas Santos Gomes and Fabiana Rodrigues Leta. “Applications of computer vision techniques in the agriculture and food industry: a review”. In: *European Food Research and Technology* 235.6 (2012), pp. 989–1000 (cit. on pp. 1, 7).

[23]A Gongal, S Amatya, Manoj Karkee, Q Zhang, and K Lewis. “Sensors and systems for fruit detection and localization: A review”. In: *Computers and Electronics in Agriculture* 116 (2015), pp. 8–19 (cit. on pp. 1, 2, 5, 7, 8, 27).

[24]Rafael C Gonzalez, Richard E Woods, and Steven L Eddins. “Digital Image Processing Using MATLAB: AND Mathworks, MATLAB Sim SV 07”. In: (2007) (cit. on pp. 4–6).

[25]Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016 (cit. on pp. 8, 12, 13, 15–17, 22, 31).

[26]Mikell P Groover. “Automation, Production Systems, and Computer Integrated Manufacturing”. In: (1990) (cit. on pp. 4–6).

[27]Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. “Deep learning for visual understanding: A review”. In: *Neurocomputing* 187 (2016), pp. 27–48 (cit. on pp. 6, 17–20, 23, 28, 30, 31).

[28]Chris Harris and Mike Stephens. “A combined corner and edge detector.” In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244 (cit. on p. 6).

[29]Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. “Neuroscience-inspired artificial intelligence”. In: *Neuron* 95.2 (2017), pp. 245–258 (cit. on pp. 10, 12, 16, 20).

[30]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 21, 22).

[31]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *European Conference on Computer Vision*. Springer. 2014, pp. 346–361 (cit. on pp. 23, 24).

[32]Robert Hecht-Nielsen et al. “Theory of the backpropagation neural network.” In: *Neural Networks* 1.Supplement-1 (1988), pp. 445–448 (cit. on p. 14).

[33]Suzana Herculano-Houzel. “The human brain in numbers: a linearly scaled-up primate brain”. In: *Frontiers in human neuroscience* 3 (2009) (cit. on p. 12).

[34]Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012) (cit. on p. 22).

[35]Kuo-Yi Huang. “Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features”. In: *Computers and Electronics in agriculture* 57.1 (2007), pp. 3–11 (cit. on p. 26).

[36]D. H. Hubel and T. N. Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. In: *The Journal of Physiology* 148.3 (1959), pp. 574–591 (cit. on p. 16).

[37]David Hughes, Marcel Salathé, et al. “An open access repository of images on plant health to enable the development of mobile disease diagnostics”. In: *arXiv preprint arXiv:1511.08060* (2015) (cit. on p. 26).

[38]ICT. “ICT Facts and Figures–the World in 2017.” In: *Geneva: International Telecommunication Union* (2017) (cit. on p. 29).

[39]UNEP IFAD. “Smallholders, food security and the environment”. In: *Rome: International Fund for Agricultural Development* (2013) (cit. on p. 29).

[40]Patrick Jackman, Da-Wen Sun, Cheng-Jin Du, and Paul Allen. “Prediction of beef eating qualities from colour, marbling and wavelet surface texture features using homogenous carcass treatment”. In: *Pattern Recognition* 42.5 (2009), pp. 751–763 (cit. on p. 7).

[41]Katarzyna Janocha and Wojciech Marian Czarnecki. “On Loss Functions for Deep Neural Networks in Classification”. In: *CoRR* abs/1702.05659 (2017). arXiv: 1702.05659 (cit. on p. 13).

[42]AR Jimenez, R Ceres, and JL Pons. “A survey of computer vision methods for locating fruit on trees”. In: *Transactions of the ASAE* 43.6 (2000), p. 1911 (cit. on pp. 1, 5, 7, 8).

[43]Keren Kapach, Ehud Barnea, Rotem Mairon, Yael Edan, and Ohad Ben-Shahar. “Computer vision for fruit harvesting robots–state of the art and challenges ahead”. In: *International Journal of Computational Vision and Robotics* 3.1-2 (2012), pp. 4–34 (cit. on pp. 1, 7, 8).

[44]Rupinder Kaur and Shrusti Porwal. “An Optimized Computer Vision Approach to Precise Well-Bloomed Flower Yielding Prediction using Image Segmentation”. In: *International Journal of Computer Applications* 119.23 (2015) (cit. on p. 7).

[45]Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980 (cit. on p. 14).

[46]Jyoti A Kodagali and S Balaji. “Computer vision and image analysis based techniques for automatic characterization of fruits-a review”. In: *International Journal of Computer Applications* 50.6 (2012) (cit. on pp. 1, 7).

[47]Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. on pp. 2, 10, 11, 16, 20, 23).

[48]Anders Krogh and John A Hertz. “A simple weight decay can improve generalization”. In: *Advances in neural information processing systems*. 1992, pp. 950–957 (cit. on p. 22).

[49]Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444 (cit. on pp. 2, 8, 9, 13, 14, 16–18, 25).

[50]Ming Li, Kenji Imou, Katsuhiro Wakabayashi, and Shinya Yokoyama. “Review of research on agricultural vehicle autonomous guidance”. In: *International Journal of Agricultural and Biological Engineering* 2.3 (2009), pp. 1–16 (cit. on p. 7).

[51]Kirt Lillywhite, Dah-Jye Lee, Beau Tippetts, and James Archibald. “A feature construction method for general object recognition”. In: *Pattern Recognition* 46.12 (2013), pp. 3300 –3314 (cit. on p. 1).

[52]Nicolas Louka, Frédéric Juhel, Victurnien Fazilleau, and Pierre Loonis. “A novel colorimetry analysis used to compare different drying fish processes”. In: *Food Control* 15.5 (2004), pp. 327–334 (cit. on p. 7).

[53]David G Lowe. “Object recognition from local scale-invariant features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157 (cit. on p. 6).

[54]Yang Lu, Shujuan Yi, Nianyin Zeng, Yurong Liu, and Yong Zhang. “Identification of rice diseases using deep convolutional neural networks”. In: *Neurocomputing* 267 (2017), pp. 378–384 (cit. on p. 26).

[55]Jitendra Malik, Pablo Arbeláez, João Carreira, Katerina Fragkiadaki, Ross Girshick, Georgia Gkioxari, Saurabh Gupta, Bharath Hariharan, Abhishek Kar, and Shubham Tulsiani. “The three R’s of computer vision: Recognition, reconstruction and reorganization”. In: *Pattern Recognition Letters* 72.Supplement C (2016). Special Issue on ICPR 2014 Awarded Papers, pp. 4 –14 (cit. on p. 1).

[56]Chakradhar Mattupalli, Ruth K Genger, and Amy O Charkowski. “Evaluating incidence of Helminthosporium solani and Colletotrichum coccodes on asymptomatic organic potatoes and screening potato lines for resistance to silver scurf”. In: *American journal of potato research* 90.4 (2013), pp. 369–377 (cit. on p. 26).

[57]Cheryl L McCarthy, Nigel H Hancock, and Steven R Raine. “Applied machine vision of plants: a review with implications for field deployment in automated farming operations”. In: *Intelligent Service Robotics* 3.4 (2010), pp. 209–217 (cit. on pp. 1, 7).

[58]Dmytro Mishkin and Jiri Matas. “All you need is a good init”. In: *CoRR* abs/1511.06422 (2015). arXiv: 1511.06422 (cit. on p. 15).

[59]Sharada P Mohanty, David P Hughes, and Marcel Salathé. “Using deep learning for image-based plant disease detection”. In: *Frontiers in plant science* 7 (2016) (cit. on pp. 1, 26, 29).

[60]Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 483–499 (cit. on p. 27).

[61]Anh Nguyen, Jason Yosinski, and Jeff Clune. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 427–436 (cit. on p. 28).

[62]Michael A Nielsen. *Neural networks and deep learning*. 2015 (cit. on p. 15).

[63]George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. “Weakly-and semi-supervised learning of a DCNN for semantic image segmentation”. In: *arXiv preprint arXiv:1502.02734* (2015) (cit. on p. 30).

[64]Ciro Potena, Daniele Nardi, and Alberto Pretto. “Fast and accurate crop and weed identification with summarized train sets for precision agriculture”. In: *International Conference on Intelligent Autonomous Systems*. Springer. 2016, pp. 105–121 (cit. on p. 27).

[65]Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. “Deep learning for multi-task plant phenotyping”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2055–2063 (cit. on p. 27).

[66]Maryam Rahnemoonfar and Clay Sheppard. “Real-time yield estimation based on deep learning”. In: *SPIE Commercial+ Scientific Sensing and Imaging*. International Society for Optics and Photonics. 2017, pp. 1021809–1021809 (cit. on pp. 1, 27).

[67]Rajat Raina, Anand Madhavan, and Andrew Y Ng. “Large-scale deep unsupervised learning using graphics processors”. In: *Proceedings of the 26th annual international conference on machine learning*. ACM. 2009, pp. 873–880 (cit. on p. 10).

[68]Joseph Redmon and Anelia Angelova. “Real-time grasp detection using convolutional neural networks”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1316–1322 (cit. on p. 23).

[69]Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99 (cit. on pp. 23, 25, 48).

[70]A.E. Rich. *Potato Diseases*. Elsevier Science, 2013 (cit. on p. 26).

[71]Frank Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386 (cit. on p. 10).

[72]Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. “Imagenet large scale visual recognition challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252 (cit. on p. 16).

[73]Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. “Deepfruits: A fruit detection system using deep neural networks”. In: *Sensors* 16.8 (2016), p. 1222 (cit. on pp. 2, 27).

[74]Robert J Schalkoff. *Digital image processing and computer vision*. Vol. 286. Wiley New York, 1989 (cit. on pp. 4, 6).

[75]Dominik Scherer, Andreas Müller, and Sven Behnke. “Evaluation of pooling operations in convolutional architectures for object recognition”. In: *Artificial Neural Networks–ICANN 2010* (2010), pp. 92–101 (cit. on p. 18).

[76]Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85 –117 (cit. on pp. 8–10).

[77]Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. “Overfeat: Integrated recognition, localization and detection using convolutional networks”. In: *arXiv preprint arXiv:1312.6229* (2013) (cit. on p. 23).

[78]Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 21).

[79]Arti Singh, Baskar Ganapathysubramanian, Asheesh Kumar Singh, and Soumik Sarkar. “Machine learning for high-throughput stress phenotyping in plants”. In: *Trends in plant science* 21.2 (2016), pp. 110–124 (cit. on p. 26).

[80]F Sistler. “Robotics and intelligent machines in agriculture”. In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 3–6 (cit. on p. 7).

[81]Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. “Deep neural networks based recognition of plant diseases by leaf image classification”. In: *Computational intelligence and neuroscience* 2016 (2016) (cit. on p. 26).

[82]DC Slaughter, DK Giles, and D Downey. “Autonomous robotic weed control systems: A review”. In: *Computers and electronics in agriculture* 61.1 (2008), pp. 63–78 (cit. on p. 7).

[83]Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014 (cit. on pp. 4–6).

[84]Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. “Highway networks”. In: *arXiv preprint arXiv:1505.00387* (2015) (cit. on p. 21).

[85]Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. 2013, pp. 1139–1147 (cit. on p. 14).

[86]Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 21).

[87]Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. “Intriguing properties of neural networks”. In: *arXiv preprint arXiv:1312.6199* (2013) (cit. on p. 28).

[88]R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London, 2010 (cit. on pp. 1, 5, 6).

[90]Alberto Tellaeche, Xavier P BurgosArtizzu, Gonzalo Pajares, Angela Ribeiro, and César Fernández-Quintanilla. “A new vision-based approach to differential spraying in precision agriculture”. In: *computers and electronics in agriculture* 60.2 (2008), pp. 144–155 (cit. on p. 7).

[91]RD Tillett. “Image analysis for agricultural processes: a review of potential opportunities”. In: *Journal of agricultural Engineering research* 50 (1991), pp. 247–258 (cit. on p. 7).

[92]Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. “Selective search for object recognition”. In: *International journal of computer vision* 104.2 (2013), pp. 154–171 (cit. on pp. 23, 24).

[93]Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer. 2014, pp. 818–833 (cit. on pp. 20, 21).

[94]C Lawrence Zitnick and Piotr Dollár. “Edge boxes: Locating object proposals from edges”. In: *European Conference on Computer Vision*. Springer. 2014, pp. 391–405 (cit. on p. 23).

Websites

[89]Brett Szymik. *A Nervous Journey*. URL: http://askabiologist.asu.edu/neuron-anatomy. (accessed: 27.09.2017) (cit. on p. 10).

7 Appendices

7.1 Appendix A. Potato Disease Classification Using Convolutional Neural Networks

Conference: 11th European Conference on Precision Agriculture, Edinburgh, Scotland, 2017

Autors: Dor Oppenheim and Guy Shani

Advances in Animal Biosciences: Precision Agriculture (ECPA) 2017, (2017), **8:2**, pp 244–249 © The Animal Consortium 2017
doi:10.1017/S2040470017001376

advances in animal biosciences

Potato Disease Classification Using Convolution Neural Networks

D. Oppenheim[1†] and G. Shani[2]

[1]*Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel;* [2]*Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel*

Many plant diseases have distinct visual symptoms which can be used to identify and classify them correctly. This paper presents a potato disease classification algorithm which leverages these distinct appearances and the recent advances in computer vision made possible by deep learning. The algorithm uses a deep convolutional neural network training it to classify the tubers into five classes, four diseases classes and a healthy potato class. The database of images used in this study, containing potatoes of different shapes, sizes and diseases, was acquired, classified, and labelled manually by experts. The models were trained over different train-test splits to better understand the amount of image data needed to apply deep learning for such classification tasks.

Keywords: Plant disease detection and classification, Computer vision, Convolutional neural network, Potato diseases

Introduction

Potato (*Solanum tuberosum*) is the third most important food crop in the world, after cereals and rice. Global production exceeds 300 million metric tons and is an important nutrition and calorie provider for humanity (Pareek 2016). Potato production is threatened by several diseases resulting in considerable yield losses, and causing decrease in the quality and increase in the price of potatoes (Taylor *et al.*, 2008). An early disease detection system can aid in avoiding such cases. Moreover, it can improve the management of the crop and can further prevent the spread of diseases (Rich 2013). Manually detecting and sorting potatoes is difficult, costly, and time consuming, while computerized inspection may be more efficient and cost effective.

Computer vision and machine learning techniques for disease detection have been broadly researched in the last two decades (Garcia and Barbedo 2016). Diseases can be detected using expensive and bulky digital imaging sensors, such as spectral or near-infrared sensors. Using such sensors encumbers the widespread implementation of these methods due to its high costs and maintenance (Sankaran *et al.*, 2010). On the other hand, researchers using the visible light bandwidth, which can be captured by relatively low cost cameras, have usually focused on a single type of disease (Zhang *et al.*, 2014). A single case identification is insufficient for real-world applications, as a single tuber can be infected a number of diseases (Cubero *et al.*, 2016).

This paper leverages recent advances in computer vision and object recognition, for classifying multiple diseases in potatoes. In 2012 a group of researchers from Toronto won the Large Scale Visual Recognition Challenge (ILSVRC) competition by improving the classification of the ImageNet database by more than 10%. They achieved a top-5 error rate of 15.3% when using a deep Convolution Neural Network (CNN), while the second best achieved 26.2% error rate (Krizhevsky, Sutskever, & Hinton, 2012). Since then, CNN methods have improved and recently the classification error dropped to 3.73% by the winning team for the same task (Abdi and Nahavandi 2016). In the field of computer vision for agricultural applications, the use of CNNs and other deep neural networks is continuously increasing (Gongal *et al.*, 2015). A CNN was recently used for detecting and classifying seven fruits in field conditions, improving detection accuracy by 3% from the last state of the art (Sa *et al.*, 2016). CNNs used in classification tasks, such as disease classification of plant leaves or quality control of harvested fruit and vegetable, reached accuracy of more than 97% (Mohanty *et al.*, 2016; Tan *et al.*, 2015). In order to create successful CNNs, a large amount of training data is needed (Sermanet *et al.*, 2013). Therefore, the first aim of the current research was the collection of a sufficient dataset and classification of the displayed diseases. Results indicate a first step towards multiple disease classification for potatoes using CNN.

Materials and Methods

Data acquisition

Photos of 400 contaminated potatoes of different shapes, sizes and tones were acquired under normal uncontrolled illumination conditions. The tubers were manually classified by experts as a standard procedure of statistically estimating the rate of various diseases in seed potato tubers prior to planting them in the fields. This procedure is done annually independent of the current research. The potatoes were contaminated with

† E-mail: doropp@post.bgu.ac.il

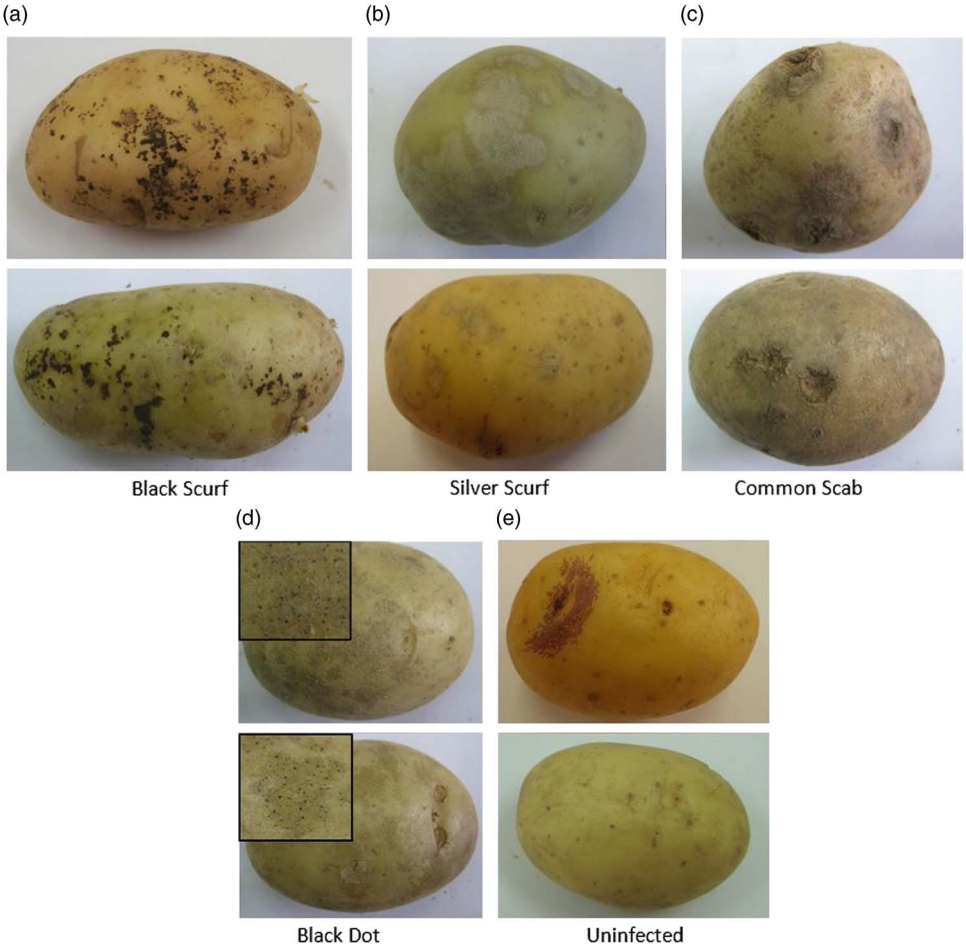

Figure 1 Examples of visual symptoms on potato diseases: (a) Black Scurf disease - irregular, black, scab-like marks on the skin of the tuber. (b) Silver Scurf disease - circular or irregular, tan to silvery gray lesions on the tuber's skin. (c) Common Scab disease - circular brown rough areas, with irregular margins which can coalesce into larger areas. (d) Black Dot disease - tiny black dots on the skin of the tuber (magnified in top left corner). (5) Uninfected tuber.

four different diseases, all with significant visual symptoms on the tuber's skin (see Figure 1). The images were acquired using multiple types of standard cameras, captured from one viewpoint only. The cameras used were Sony DSC-T200, the Apple iPhone 4 camera, and the Samsung Galaxy S3 camera.

Data preparation

The images acquired were used to create the training and tests sets for the CNN. Every visual symptom of a disease was marked and labelled using the image labeler application in MatLab 2014b. The labelling was done with rectangular bounding boxes encompassing the visual symptom but also much regular potato skin, as seen in Figure 2. The marked areas were cropped from the original image, transformed into grayscale, and resized to a standard 224×224 pixel square. After preprocessing, a total of 2,465 patches of diseased potatoes was gathered including: 265 Black Dot patches, 469 Black Scurf patches, 686 Common Scab patches, 738 Silver Scurf patches and 307 uninfected patches.

Performance Measurement

The experiment was designed to evaluate the performance of the CNN's learning algorithm in classifying four diseases and uninfected potatoes. As manually labelling diseased patches of potatoes is a tedious and time costly task, an important task was to determine the minimal amount of training data that provides sufficient classification accuracy. The CNN was trained with different sizes of training sets. The smallest training set used for training was 10% of the 2,465 images, incrementally increasing by 10% to 90% of the whole dataset as detailed in Table 1. In each increment the images were selected uniformly from the whole dataset. Testing of the algorithm was done on the remaining data. In total the training and testing phases was repeated 9 times over different training set sizes.

Each training set was trained for 90 epochs, where one epoch is defined as a one full training cycle on every sample in the training set. The choice of limiting to 90 epochs was made based on empirical observations that revealed that the learning converged well within 90 epochs (as can be seen in Figure 4). In order to compare between the different results over the 9 training sets, the error rate of the best scoring guess was calculated as the number of errors divided by the total number of test images in every epoch. The error was calculated both for test and train sets, in order to understand the over\under fitting of the procedure.

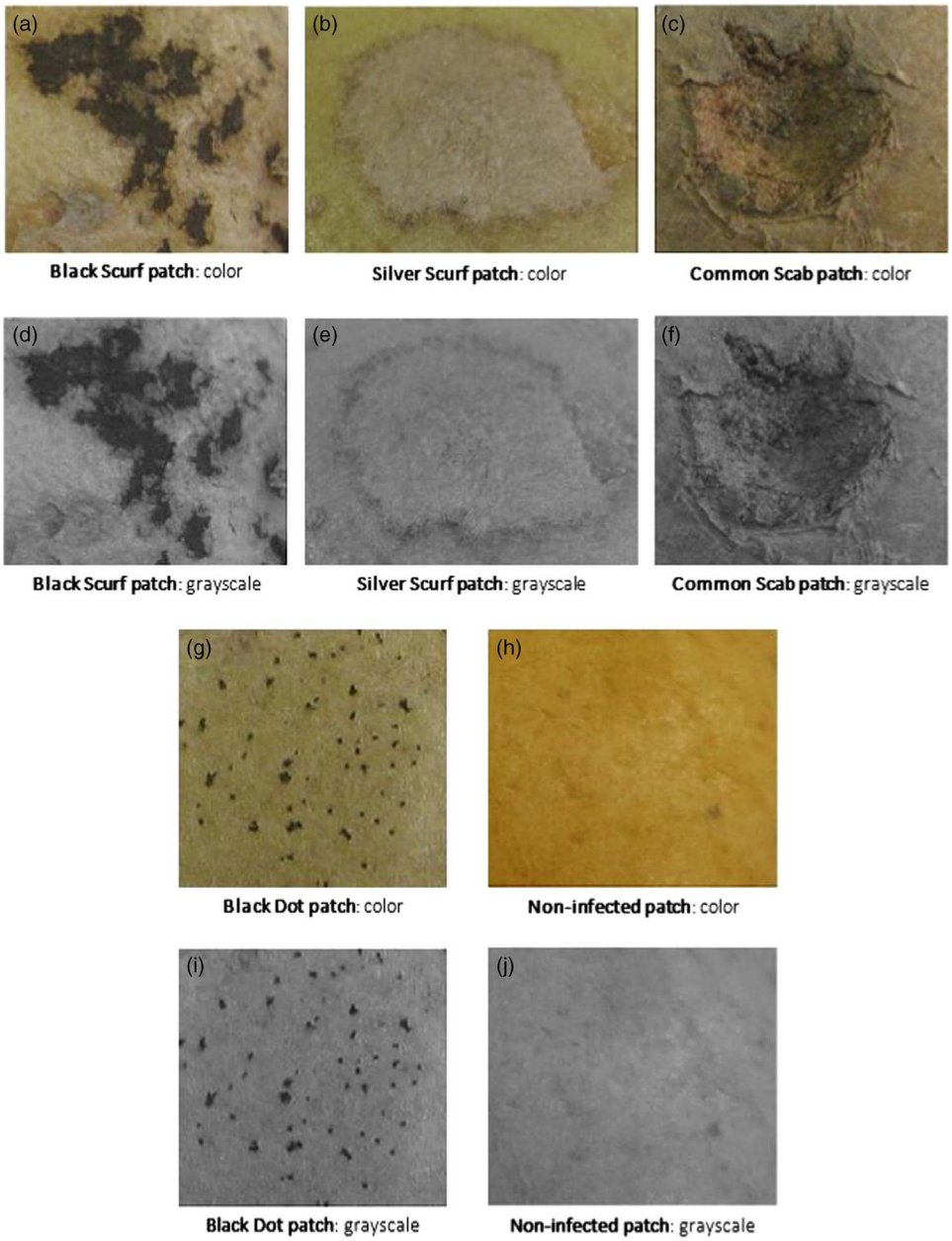

Figure 2 Examples of diseased potato patches before and after the transformation to grayscale. (a)–(c), (g) and (h) are the original RGB images from each class. (d)-(f), (i) and (j) are the same images after the conversion to grayscale, using Matlab's rgb2gray function.

Algorithm

The algorithm chosen for the image classification task was a deep convolutional neural network (CNN). The basic architecture chosen for this problem was a CNN developed by the Visual Geometry Group (VGG) from the University of Oxford named CNN-F due to its faster training time (Chatfield *et al.*, 2014). Several new dropout layers were added to the VGG architecture to deal with problems of over fitting, especially due to the relatively small dataset.

The required input image size for this network is a 224×224 matrix. The CNN comprises 8 learnable layers, the first 5 of which are convolutional, followed by 3 fully-connected layers and ending with a softmax layer (see Figure 3). The softmax layer normalizes the input received from the last fully-connected layer (fc3) producing a distribution of values, one for each class. The sum of these values add up to 1 and they represent the probability of the input image to belong to one of the five classes. This softmax layer was also altered and adapted, reducing its size from 1,000 to 5 to fit our classification task.

The hyper-parameters used in each training experiment were:

- Solver type: Stochastic Gradient Descent
- Learning rate: 0.0001
- Batch size: 50
- Momentum: 0.9
- Weight decay: 0.0005

Table 1 *Train and test set division.*

Training Set	% Train	% Test	# Black Scurf trained	# Black Scurf tested	# Common Scab trained	# Common Scab tested	# Silver Scurf trained	# Silver Scurf tested	# Black Dot trained	# Black Dot tested	# Uninfected trained	# Uninfected tested
1	90	10	423	46	618	68	665	73	239	26	277	30
2	80	20	377	92	550	136	592	146	213	52	247	60
3	70	30	331	138	482	204	519	219	187	78	217	90
4	60	40	285	184	414	272	446	292	161	104	187	120
5	50	50	239	230	346	340	373	365	135	130	157	150
6	40	60	193	276	278	408	300	438	109	156	127	180
7	30	70	147	322	210	476	227	511	83	182	97	210
8	20	80	101	368	142	544	154	584	57	208	67	240
9	10	90	55	414	74	612	81	657	31	234	37	270
Total	469	686	738	265	307							

Training CNNs usually requires a large amount of labelled data in order to perform a good classification. Therefore, two methods were used for data augmentation; Mirroring creates additional examples by flipping the images used in training randomly. As the direction of the photos was arbitrary, mirroring the image horizontally does not change the correctness of the data; Cropping was also used, cropping the image randomly to different sizes, while keeping the cropped image minimum size to 190×190, can achieve data diversity. The use of each data augmentation method was done randomly. Before each image was inserted into the net for training it was mirrored, cropped or inserted without altering in equal distributions. Therefore, two thirds of the images trained were altered.

Results and discussion

Results indicate, as expected, that using more data for the training phase improves the classification and reduces the error rate (see Figure 4). The best trained model (trained on 90% of the dataset and tested on the remaining 10%) classified correctly 96% of the images. Results indicate that for 8 out of the 9 training sets, accuracy does not drop below 90% as the training set size decreases (Figure 4); the average difference of error rates between the best training set (90% train-10% test) and the worst training set (20% train-80% test) in these 8 sets was 5.73%. There is a significant drop in performance when the CNN was trained on 10% of the dataset and tested on 90% of it. Correct classification for this training set decreased to 83% as opposed to 90% of the classifier obtained with 20% train and 80% test. The relatively small decrease in accuracy (Table 2) for most training set sizes, is an indicator that a small amount of potato images could suffice for training a sufficiently accurate CNN.

In order to further evaluate the CNN's classification a confusion matrix was calculated. The confusion matrix's columns represent the CNN's class classification while the rows represent the actual classes. This type of representation can help evaluate the CNN's classification of each class. Figure 5 shows a confusion matrix of the best performing CNN, trained on 90% of the dataset and tested on 10%. The confusion matrix shows that the CNN classified correctly and with high accuracy infected potato tubers; 100% of the tubers which were infected with Black Dot and Black Scurf were classified correctly; over 92% of the Silver Scurf and Common Scab infected tubers were classified correctly as well. The CNN's performance dropped when classifying uninfected tubers. Most of the CNN's misclassifications occurred when classifying uninfected tubers to the disease class – Silver Scurf. Silver Scurf's visual symptom are bright tan to silvery gray lesions on the tuber's skin that resemble uninfected skin.

These results indicate that the trained CNN can classify correctly and accurately the four diseases presented here. However, uninfected tubers were harder to classify. The fact that the diseases were classified with high accuracy makes it suitable for a system which identification of the disease is important. Most misclassifications occurred for the uninfected

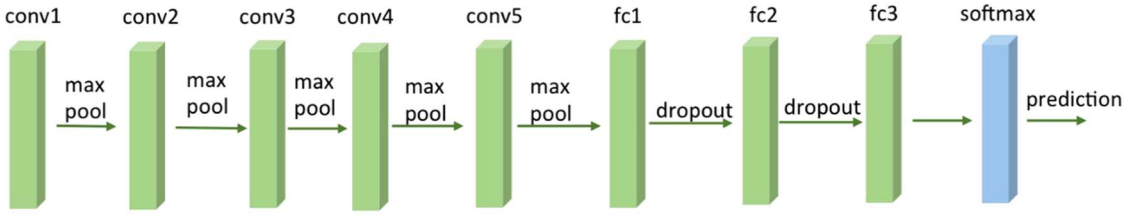

Figure 3 A simplified model of the CNN used.

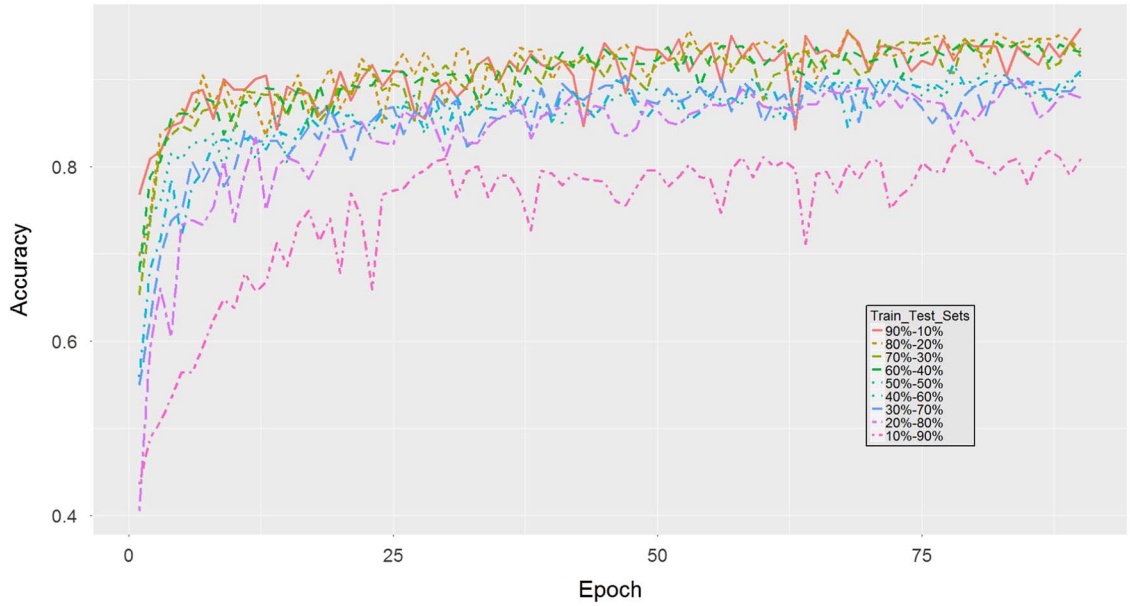

Figure 4 Results of the experiment. Top plot shows the accuracy of each training set (colored and dotted graphs) according to the epoch number. The bottom plot show the smoothed results.

Table 2 *Best performing model accuracy results for each train-test set*

Train- Test Set Division	90%–10%	80%–20%	70%–30%	60%–40%	50%–50%	40%–60%	30%–70%	20%–80%	10%–90%
Accuracy	0.9585	0.9567	0.9465	0.9454	0.9069	0.9183	0.9041	0.9012	0.8321

class, for practical use these mistakes have less affect since planting infected tubers can spread the disease and cause considerable damage while misclassifying uninfected tubers can be solved. In this experiment only 307 images of uninfected tubers were used, increasing the amount of data of uninfected tubers can increase classification accuracy.

Conclusions and future work

The applicability of a convolution neural network in classifying image patches of diseased potatoes into four disease classes and a uninfected class was examined. The 2,465 images classified by the trained CNN model varied in the acquisition device and conditions. Results indicate the robustness of the classification algorithm allowing for uncontrolled acquisition conditions. Results reveal that the correct classification of fully trained CNN models ranges from 83% for the model trained on the least amount of data, to 96%, when the model was trained on 90% of the data. To obtain classification rates higher than 90% it is sufficient to use 20% of the images (i.e., 493 images).

These results further show that combining the CNN introduced here with a sliding window algorithm could be utilized for classifying full images of potatoes to different diseases with little labelling work beforehand. Ongoing research is aimed to develop a classification algorithm with an expanded number of disease classes. Acquiring data can be done easily since there are no constraints on the data acquisition.

Acknowledgements

This work is supported by the Ministry of Agriculture and partially supported by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Initiative and the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering,

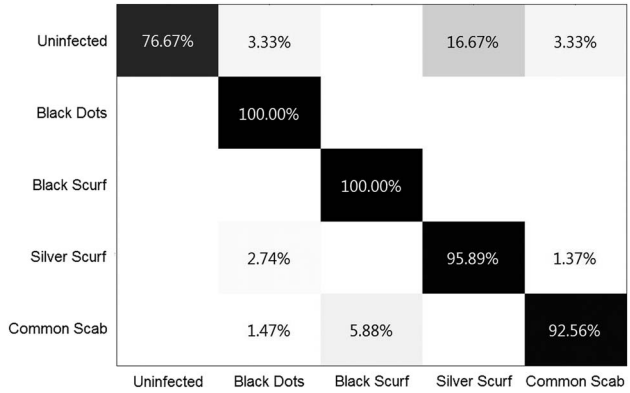

Figure 5 A confusion matrix of the CNN trained on 90% of the dataset and tested on the remaining 10%. Rows represent the actual classes of an image. Columns represent the CNN's class prediction. Each cell in the matrix represent the percentage of images of the row's class that were classified to the column's class.

both at Ben-Gurion University of the Negev. We thank Professor Yael Edan for her important comments.

References

Masoud A and Nahavandi S 2016. Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks, https://arxiv.org/abs/1609.05672.

Chatfield K, Simonyan K, Vedaldi A and Zisserman A 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. https://arxiv.org/abs/1405.3531.

Cubero S, Lee SW, Aleixos N, Albert F and Blasco J 2016. Automated Systems Based on Machine Vision for Inspecting Citrus Fruits from the Field to Postharvest—a Review. Food and Bioprocess Technology 1–17.

Garcia J and Barbedo A 2016. A Review on the Main Challenges in Automatic Plant Disease Identification Based on Visible Range Images. Biosystems Engineering 144, 52–60.

Gongal A, Amatya S, Karkee M, Zhang Q and Lewis K 2015. Sensors and Systems for Fruit Detection and Localization: A Review. Computers and Electronics in Agriculture 116, 8–19.

Krizhevsky A, Sutskever I and Hinton GE 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems 1–9.

Mohanty SP, Hughes D and Salathé M 2016. Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science 7.

Pareek S 2016. Postharvest Ripening Physiology of Crops. CRC Press. Taylor and Francis, FL, USA.

Rich AE 2013. Potato Diseases. Elsevier Science. Academic Press, New York, USA.

Sa I, Ge Z, Dayoub F, Upcroft B, Perez T and McCool C 2016. DeepFruits: A Fruit Detection System Using Deep Neural Networks. Sensors 16 (8), 1222.

Sankaran S, Mishra A, Ehsani R and Davis C 2010. A Review of Advanced Techniques for Detecting Plant Diseases. Computers and Electronics in Agriculture 72 (1), 1–13.

Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R and LeCun Y 2013. OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. https://arxiv.org/abs/1312.6229.

Tan W, Zhao C and Wu H 2015. Intelligent Alerting for Fruit-Melon Lesion Image Based on Momentum Deep Learning. Multimedia Tools and Applications 75 (24), 16741–16761.

Taylor RJ, Pasche JS and Gudmestad NC 2008. Susceptibility of Eight Potato Cultivars to Tuber Infection by Phytophthora Erythroseptica and Pythium Ultimum and Its Relationship to Mefenoxam-Mediated Control of Pink Rot and Leak. Annals of Applied Biology 152 (2), 189–199.

Zhang B, Huang W, Jiangbo L, Zhao C, Fan S, Wu J and Liu C 2014. Principles, Developments and Applications of Computer Vision for External Quality Inspection of Fruits and Vegetables: A Review. Food Research International 62, 326–343.

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Be'er Sheva, March 8, 2018

Dor Oppenheim