2020

Application of advanced algorithms and statistical techniques for weed-plant discrimination

Saman Akbar Zadeh
Edith Cowan University

Edith Cowan University

Copyright Warning

Application of advanced algorithms and statistical techniques for weed-plant discrimination

Thesis submitted in fulfilment of the requirements of the award of
Doctor of Philosophy

Saman Akbar Zadeh

Principle supervisor: Professor Kamal Alameh

Associate Supervisor: Dr Selam Ahderom

School of Science
Edith Cowan University
2020

Abstract

Precision agriculture requires automated systems for weed detection as weeds compete with the crop for water, nutrients, and light. The purpose of this study is to investigate the use of machine learning methods to classify weeds/crops in agriculture. Statistical methods, support vector machines, convolutional neural networks (CNNs) are introduced, investigated and optimized as classifiers to provide high accuracy at high vehicular speed for weed detection.

Initially, Support Vector Machine (SVM) algorithms are developed for weed-crop discrimination and their accuracies are compared with a conventional data-aggregation method based on the evaluation of discrete Normalised Difference Vegetation Indices (NDVIs) at two different wavelengths. The results of this work show that the discrimination performance of the Gaussian kernel SVM algorithm, with either raw reflected intensities or NDVI values being used as inputs, provides better discrimination accuracy than the conventional discrete NDVI-based aggregation algorithm.

Then, we investigate a fast statistical method for CNN parameter optimization, which can be applied in many CNN applications and provides more explainable results. This study specifically applies Taguchi based experimental designs for network optimization in a basic network, a simplified inception network and a simplified Resnet network, and conducts a comparison analysis to assess their respective performance and then to select the hyper parameters and networks that facilitate faster training and provide better accuracy. Results show that, for all investigated CNN architectures, there is a measurable improvement in accuracy in comparison with un-optimized CNNs, and that the Inception network yields the highest improvement (~ 6%) in accuracy compared to simple CNN (~ 5%) and Resnet CNN counterparts (~ 2%).

Aimed at achieving weed-crop classification in real-time at high speeds, while maintaining high accuracy, the algorithms are

uploaded on both a small embedded NVIDIA Jetson TX1 board for real-time precision agricultural applications, and a larger high throughput GeForce GTX 1080Ti board for aerial crop analysis applications. Experimental results show that for a simplified CNN algorithm implemented on a Jetson TX1 board, an improvement in detection speed of thirty times (60 km/hr) can be achieved by using spectral reflectance data rather than imaging data. Furthermore, with an Inception algorithm implemented on a GeForce GTX 1080Ti board for aerial weed detection, an improvement in detection speed of 11 times (~2300 km/hr) can be achieved, while maintaining an adequate detection accuracy above 80%. These high speeds are attained by reducing the data size, choosing spectral components with high information contents at lower resolution, pre-processing efficiently, optimizing the deep learning networks through the use of simplified faster networks for feature detection and classification, and optimizing computational power with available power and embedded resources, to identify the best fit hardware platforms.

Copyright Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

I. incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education;

II. contain any material previously published or written by another person except where due reference is made in the text; or

III. contain any defamatory material

IV. I also grant permission for the Library at Edith Cowan University to make duplicate copies of my thesis as required.

Signature: 

Date: 1, 10, 2020

Acknowledgment

I would like to thank my supervisors, Prof. Kamal Alameh and Dr. Selam Ahderom for their professional guidance during the course of my PhD studies. I would also like to thank Paul Symonds and Navid Mavaddat for their help in the experiments and fruitful discussion.

Additionally I would like to thank Edith Cowan University and the Electron Science Research Institute, the Grains Research and Development Corporation (GRDC), Australian Research Council, Photonic Detection Systems Pty. Ltd, Australia for the provision of scholarship funding, without which this project could not have been completed. I also would like to acknowledge Pawsey Supercomputing Centre for providing access to their facility for neural network optimization

Dedication

I dedicate my thesis to my parents (Fatema and Mohammad Reza) who, through thick and thin, have been there for me. Their drive and help is what has make me who I am.

I also dedicate it to my lovely wife Mona, whose unyielding love, support, and encouragement have enriched my soul and inspired me for a better life.

Additionally, I dedicate it to all of my family and friends; I learned from them the meaning of life and without them, none of my success would be possible.

Table of Content

List of Figures

Table of Tables

List of Publications

The following academic articles were published or under revision as a part of current dissertation [1-4].

1. Akbarzadeh S, Paap A, Ahderom S, Apopei B and Alameh K. Plant discrimination by Support Vector Machine classifier based on spectral reflectance. Computers and electronics in agriculture. 2018;148:250-8

2. Akbarzadeh S, Ahderom S and Alameh K. A Statistical Approach to Provide Explainable Convolutional Neural Network Parameter Optimization. International Journal of Computational Intelligence Systems. 2019;12 2:1635-48.

3. Akbarzadeh S, Ahderom S and Alameh K. Application of spectral reflectance for increasing plant discrimination speed in precision agriculture. In: *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)* 2019, pp.140-2. IEEE.

4. Akbarzadeh S, Ahderom S, Miller C, Rowe J and Alameh K. High-detection-speed weed sensors employing advanced machine learning algorithms. GigaScience. 2020; Under revision.

Chapter 1: Introduction

1.1 Background Motivation

Weeds are one of the most challenging problems for farmers, threatening their ability to produce good-quality food cost-effectively [5]. Relying only on traditional chemical weed control not only imposes high financial pressure on farmers, but also has negative impacts on the environment, creating herbicide-resistant weeds and polluted soils [6-8]. The cost of weed control is approximately $3.3 billion annually ($146/ha) in Australia (including yield losses), making the management of weeds a top priority for farmers and researchers. In addition, there are several non-monetary weeds-related negative impacts, such as, fire hazard, vermin attraction, animal entanglement and restriction of access.

Advances of automation technologies and machine learning systems open the way for cheaper and more accurate weed control technologies that can be retrofitted with modern farming practices, thus enabling lower weed control costs at increased eradication speeds. Therefore, automating weed control is crucial for achieving viable weed management and control [9, 10].

Different approaches can be used for real-time weed control. The first and easiest approach is to detect green from brown (weed from the soil) when the crop has not been grown actively in the field (before planting), or use an appropriate crop detection technology that detects the standing crop during harvesting and spray any green area other than crop-root areas after harvesting. The ideal method is to detect green from green (weeds from the standing crops) then apply an appropriate weed eradication method.

Most of the research carried out on automated weed-plant discrimination is based on the use of image recognition techniques [11-15]. While image recognition using typical cameras provides relatively high discrimination accuracies (> 90%), camera images are typically captured at visible wavelengths in the rage 300–700 nm.

However, some of the key plant characteristics used in plant discrimination fall outside the visible range [16].

Plant discrimination based on spectral reflectance properties of the illuminated vegetation has been investigated by different research groups [17-20]. Typically, this plant discrimination approach is based on using two wavelengths to automatically detect green plants from soil (i.e., green-from-brown) and applying pesticides in real time to eradicate green species. However, this green-from-brown sensing approach is incapable of discriminating weeds from crops (i.e., green-from-green). Askraba, et al. [21] have reported the first real-time green-from-green discrimination sensors based on the use of a quad bike in conjunction with a 3-wavelength spectral reflectance sensor. While this sensor demonstrated canola-from-wild-radish discrimination with an accuracy of 90% at low vehicular speeds /hr, the accuracy dropped significantly at speeds exceeding 7.5 km/hr.

1.2 Literature review

Support Vector Machines (SVM) are type of supervised machine learning (ML) algorithms that can be used for classification and regression. SVM algorithms are based on the idea of finding the best separators (hyperplanes) that divide a dataset into classes. Therefore, SVM is an optimisation algorithm that tries to find linear or nonlinear hyperplanes as far away from the data points as possible. By having those hyperplanes, the new testing data point can be classified according to the side of the hyperplane it located. There can be several issues in using SVM. One is when the data is not separable, based on a linear hyperplane. In this situation, a kernel trick is used to make a nonlinear margin for separation. A possible solution is to map the data into higher dimensions until a hyperplane can be developed to segregate the classes. Another issue is when the dataset has some noises and have overlap between classes. To address this issue, a punishment factor can be added to the algorithm for overlapped data and then apply the optimisation algorithm [22-26]. The ability of SVM to classify objects has been

demonstrated experimentally in many fields of study, however, it was never investigated for weed-plant discrimination [27].

Another promising tool for classification is artificial neural networks (ANNs). ANNs have the capability of self-learning which enable them to model linear and nonlinear data for classification, regression, and pattern recognition when the mapping and solving a problem is Intrinsically difficult for human or traditional statistical standard. In an ANN, there are artificial neurons as processing units which are interconnected to each other. Each neuron has inputs and outputs and all the networks are made up of inputs and outputs. The mapping between the network inputs and outputs is trained by adjusting the internal weighting system, which comprises the weights of connections between neurons. The method to adjust the weighting system is called backpropagation. The first model of ANN was introduced by McCulloch and Pitts [28] and was inspired by a biological neural system. ANNs have since developed into convolutional neural networks (CNNs) to further provide deep learning and transform the realm of artificial intelligence (AI) [29]. CNNs, usually trained on general image datasets, have been shown to be effective in plant image recognition and classification [30-33]. Camera images have been widely used in agriculture to date [34-37]; however, there has been little research on the application of spectral sensors for increasing the detection speed, while maintaining high detection accuracy. Since spectral datasets are significantly different from imaging datasets, a special CNN optimization process is required to attain high plant classification accuracy.

To date, designing state-of-the-art CNN algorithms for specific datasets, such as images, has been time-consuming (manually engineered) and computationally intensive [38-43]. Some techniques have been proposed for automatically designing CNNs, however, these techniques either require huge computational resources or do not match manually-engineered accuracies [44-48]. In addition, for many applications, the complexity of automatically designing CNN

algorithms renders them neither interpretable nor explainable enough to be viable [49, 50].

Using a statistical approach to optimize the performance of CNN models not only helps build optimum networks according to the required criteria, but also makes these CNN models more explainable. Statistical experimental designs have been widely applied to various optimization problems [51-53] including parameter optimization of perceptron neural networks [54-56]. However, for proper optimisation of the hyper-parameters of deep learning networks, the design of experiment (DOE) is crucial, and, to our knowledge, this DOE approach has never been investigated.

Machine learning, and specifically deep learning algorithms, have been widespread use in automation [57-59] including smart farming [30-33, 60]. These algorithms can help improve the utilization of a large amount of sensor data for actionable decision, such as the use of herbicides and other farming management decisions [33, 61-65].

Camera images have been widely used in agriculture to date [34-37]; the data collected from cameras has been steadily increasing as the resolution of CCD cameras has increased over time, with an assumption that increased resolution would lead to higher classification accuracy for plant image recognition tasks [66]. However, the increase in the size of the spatial data has had undesirable effects of increased complexity and decreased processing speed necessary for use in embedded systems, such as precision agricultural equipment [67-69]. The use of powerful processors increases the complexity of the weed sensors and hence their cost and on-site power requirements.

To date, the optimization of neural network architectures and their hyper-parameters for specific applications has been computationally intensive [44, 48, 70, 71] or time-consuming (when manually engineered) [38, 40, 41]. There is a clear gap in the

literature on the development of fast plant discrimination CNN algorithms that optimize the network size and complexity, while maintaining adequate accuracy. In addition, the implementation of optimized CNNs on a hardware platform is another challenge for real-time operation at high vehicular speeds [67, 72, 73].

1.3 Thesis organization

In Chapter 2, we propose, for the first time, the use of Support Vector Machines (SVMs) in conjunction with spectral reflectance measurements for the development of a high-accuracy, high-speed plant discrimination sensor. In all experiments, a weed sensor engine, developed by Electron Science Research Institute (ESRI) at Edith Cowan University, is used to collect the intensities of the laser beams reflected off vegetation and soil at three different wavelengths, and two Normalised Difference Vegetation Indices (NDVIs) are then calculated from these measured intensities. In particular, two different investigations were carried out, namely: (1) a comparison between the accuracies of the weed detection method employing machine-learning-based Support Vector Machine (SVM) method and the conventional method of dual-NDVI-based plant discrimination [74]; (2) a comparison between the discrimination accuracies of the SVM method, with the input data being the raw reflected laser beam intensities and the NDVI values.

In Chapter 3, we propose DOE algorithms based on applying Taguchi statistical methods [75] for optimizing CNN parameters, whereby spectral agricultural data is evaluated using three simplified CNN architectures. Finally, to validate our new approach the results are compared with award-winning CNN architectures.

In Chapter 4, we propose a new approach, based on using fewer reduced dimensionality spectral reflectance data in conjunction with neural network architecture optimization, which can dramatically increase the speed of weed detection without sacrificing the accuracy. This approach comprises three steps: (i) collection and

preparation of reduced dimensionality data inputs based on two red and one near-infrared spectral reflectance components (ii) design of simplified convolutional neural networks to reduced computational costs and (iii) optimization of the computation platforms to meet the required detection speed and accuracy. A proof-of-concept system is investigated experimentally for two different platforms: (i) a small low-power embedded board, NVIDIA Jetson TX1, which is series of embedded computing boards comprising an ARM architecture central processing unit (CPU), and (ii) GeForce GTX 1080Ti board, which is a full PC based GPU board, based on the NVIDIA Pascal™ architecture.

Chapter 2: Plant discrimination by Support Vector Machine classifier based on spectral reflectance

This chapter was published as an article in Journal of Computers and Electronics in Agriculture, Volume 148, May 2018, Pages 250-258 [1]. This article appears as it does in print, with the exception of minor changes to the layout, number formats, font size and font style, and introduction and conclusion removal, which was implemented to maintain consistency in the formatting of this thesis

2.1 Abstract

Support Vector Machine (SVM) algorithms are developed for weed-crop discrimination and their accuracies are compared with a conventional data-aggregation method based on the evaluation of discrete Normalised Difference Vegetation Indices (NDVIs) at two different wavelengths. A testbed is especially built to collect the spectral reflectance properties of corn (as a crop) and silver beet (as a weed) at 635 nm, 685 nm, and 785 nm, at a speed of 7.2 km/h. Results show that the use of the Gaussian-kernel SVM method, in conjunction with either raw reflected intensities or NDVI values as inputs, provides better discrimination accuracy than that attained using the discrete NDVI-based aggregation algorithm. Experimental results carried out in laboratory conditions demonstrate that the developed Gaussian SVM algorithms can classify corn and silver beet with corn/silver-beet discrimination accuracies of 97%, whereas the maximum accuracy attained using the conventional NDVI-based method does not exceed 70%.

2.2 Methodology

2.2.1 System description

2.2.1.1 Plant Discrimination Unit

Figure 1 shows the layout of the spectral-reflectance-based Plant Discrimination Unit (PDU) that was used in the experiments to

collect the intensities of the laser beams reflected off the investigated plants and background. This PDU was developed by Askraba et al. [74, 76].

The real-time Plant Discrimination Unit (PDU) shown in Figure 1 comprised two sets of three-laser modules, two symmetric coated optical cavities, plus a linear array of high-speed photo detectors (a line-scan camera) and a motherboard housing six sub-modules including a laser driver, a central processing unit, a temperature controller, a board for a nozzle activator, a driver for the line-scan camera, and analogue and digital power supplies. The PDU unit was robustly boxed, using a rigid container and a light-weight dust shield, to overcome tough operational conditions including vibrations, shocks, and high temperatures [74].

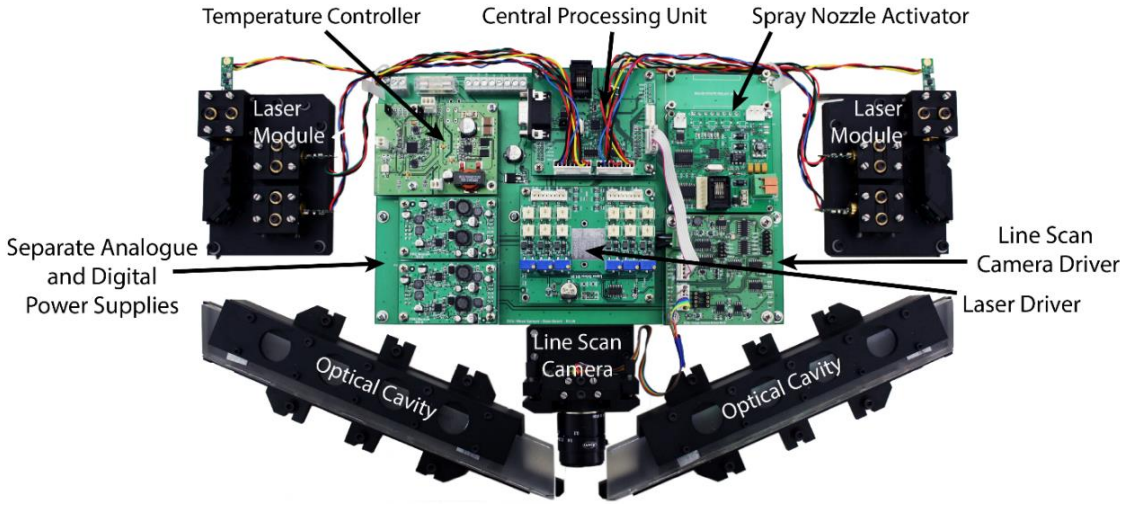


Figure 1. A picture of the PDU developed for plant discrimination. The PDU has two sets of three-laser modules, two symmetric optical cavities, a line scan camera (which is an array of high-speed linear photo detectors),plus a motherboard comprising six daughter-boards, including a central processing unit, a laser driver, a temperature controller, a line scan camera driver, a spray nozzle activator, and analogue and digital power supplies.

2.2.1.2 Vegetation illumination

Error! Reference source not found. shows the schematic of the PDU layout and shows how laser beams illuminate the vegetation.

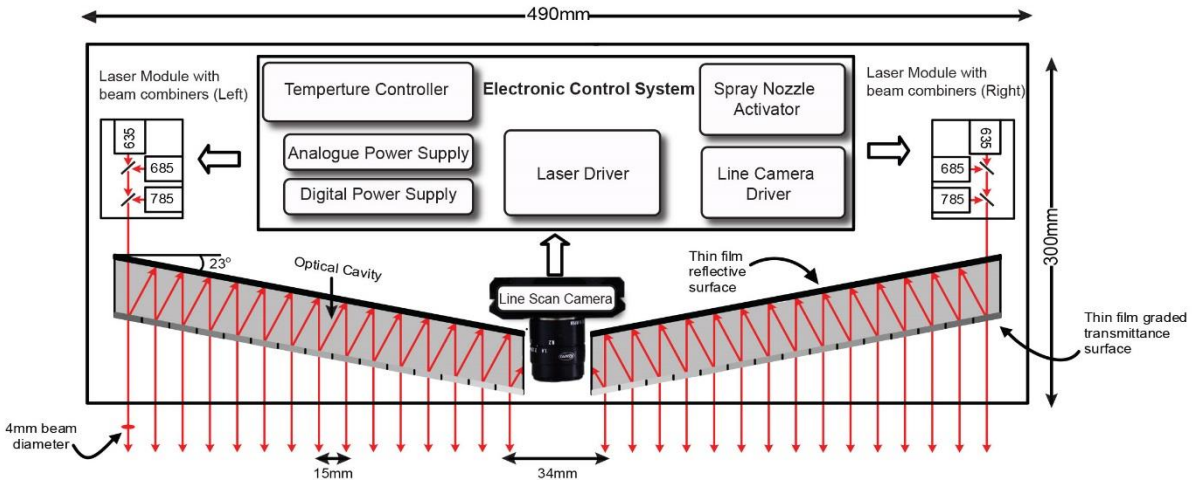


Figure 2. Schematic of the PDU, showing the generation of 1 mm collimated multi-spot beams over a span of 490 mm and the laser beam optical path (Askraba et al., 2011, and Symonds et al., 2015).

2.2.1.3 Beam Generation

Each laser module used three 1 mm collimated laser beam sources including two red (635 nm and 685 nm) lasers and one near-infrared (785 nm). Two thin-film beam combiners were used in order to combine the laser beams, as described by Askraba et al. (2013). All lasers were aligned so that the beams emitted from the laser module were collinear, overlapped, and had identical polarisation directions.

The collimated beams emitted from each laser module entered an optical cavity. An optical cavity was used to generate multiple beams from a laser source in each side. The cavity was tilted by 23 degrees to cover a span of 490 mm. The top (back) of the optical cavity was coated with a reflective surface and the bottom (front) of the cavity was coated with a non-uniform transmissive surface [77], so that all the beams emitted from the cavities had almost the same intensities [74].

The embedded controller of the PDU employed a dsPIC33F microcontroller that controlled the lasers and image sensor and carried out the data processing needed to determine the spectral properties of the plants and the background soil. The distance between two adjacent laser beams was 15 mm and the gap between

the two optical cavities was 34 mm. The total number of laser beams emerging from both cavities at one time was 30 beams (15 beams for each cavity). Each laser was driven by a constant current driver that controlled the power of each laser diode. The optical power for the 635 nm, 685 nm, and 785 nm lasers at the entrance to the optical cavity was set to 20 mW, 25 mW, and 15 mW, respectively. The line scan sensor recorded the intensities of the reflected beams. The line scan sensor was a Hamamatsu S9227-03 sensor, comprising an array of 512 photodiodes of size 250×10 µm. The analogue output voltage was converted to using a 10-bit analogue to digital converter (ADC).

2.2.1.4 Physical layout of the experiment

All the experimental data were collected using the custom-designed testing facility (referred herein as the 'testbed') shown in Figure 3, which was built and installed at the Electron Science Research Institute (ESRI) by Festo, Western Australia[1].

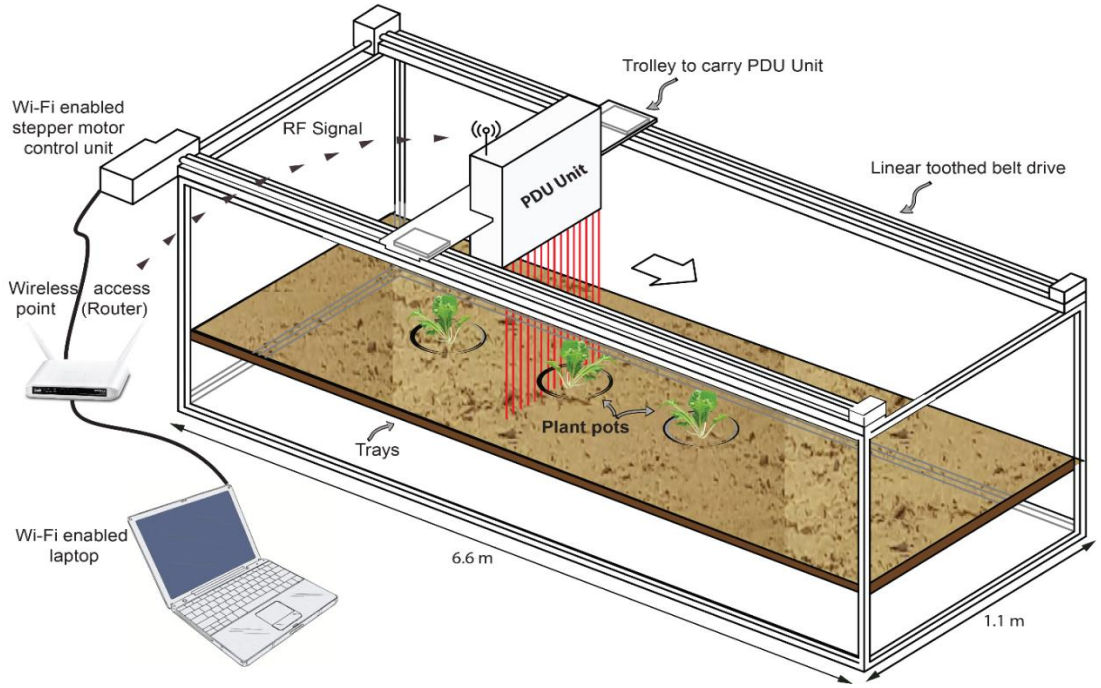


Figure 3. Physical layout of the testing facility used to collect the spectral reflectance data for evaluating the accuracies of the proposed weed discrimination algorithms.

[1] Festo: https://www.festo.com/cms/en-au_au/index.htm

The testbed shown in Figure 3 enabled data to be collected at speeds of up to 20 km/h with submillimetre accuracy. The PDU unit was placed (looking straight down, i.e. 90º. from horizontal) on a trolley which carried the PDU unit and moved it via a stepper motor. A laptop communicated via a router to control the stepper motor. Communication with the PDU was via Wi-Fi using the router, which enabled the speed of the PDU to be controlled, as illustrated in Figure 3.

2.2.2 Data description

2.2.2.1 Data collection

Corn (*Zea mays*) leaves and broad silver beet (*Beta vulgaris subsp*) leaves were used in the experiments to evaluate the performance of the developed algorithms. Data were selected representative to broad/narrow leaf combinations for the experiment. All data were captured on 6 March 2017, three weeks after germination for the corn leaves and four weeks after germination for the silver beet leaves. For each experimental run, three plants (grown in pots) were individually placed along the central area of the tray pots. In order to be able to generalise the results, training plants and testing plants were kept separately. The PDU was moved to capture the spectral reflectance data for each set of the three plants at a spatial resolution of 1 mm along the traveling speed of the PDU. The total number of scanned lines per run was 550. Data augmentation was achieved by randomly rotating the plants through ten different orientations.

2.2.2.2 NDVI calculation

The measured intensities of the laser beams reflected off the vegetation were used for the calculation of two different NDVIs, defined as

$$NDVI_{635} \triangleq \frac{P_{785}-P_{635}}{P_{785}+P_{635}} \quad (1)$$

$$NDVI_{685} \triangleq \frac{P_{785}-P_{685}}{P_{785}+P_{685}} \quad (2)$$

where, P_{635}, P_{685}, and P_{785} are the detected intensities of the 635 nm, 685 nm, and 785 nm laser beams reflected off the vegetation or background. Equations (1) and (2) show that NDVIs are normalized slop between two detected intensities. The NDVI is typically a number between -1 and 1. However, sometimes the NDVIs are multiplied by a factor of 1000 to expand its range. The intensities of the reflected laser beams were sequentially detected by the 512-pixel line-scan sensor and converted to digital data. To eliminate the effect of ambient light, the background light intensities (noise) were measured for each pixel and stored in a memory bank immediately before switching a laser source on, and then subtracted from the measured beam intensities after the laser source had been switched on.

The spatial profile of the detected beams was approximately Gaussian, and each beam occupied around 13 pixels. However, Gaussian fitting was not easy to implement for the measurement of the beam intensities due to the complexity of non-linear regression. Instead, quadratic fitting was chosen and implemented as a seven-point quadratic fit centred on the maximum pixel value, as illustrated in Figure 4. Peak detection was performed to calculate the intensities of the reflected laser beams [78]. The location of the peaks was recorded. Each peak existed in a region above the predetermined threshold. The starting and ending pixels of the region in which the peak exists were also recorded, as shown in Figure 4.

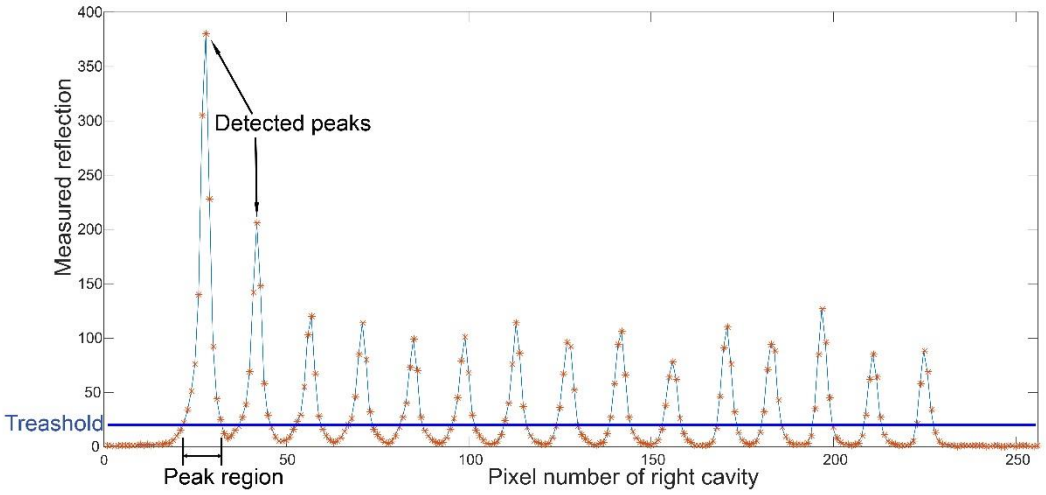


Figure 4. Illustration of the detection of peaks and recording of peak locations and regions of interest during the generation of the raw data.

Subsequently, the peak value was normalised with respect to the optical power of the incident beams emerging from the cavities, which were previously recorded by a Newport power meter, as follows [78]:

$$P_\lambda = \frac{R_\lambda}{D_\lambda} \quad (3)$$

where R_λ is the measured peak of the reflected beam, D_λ is the power of the laser beam emerging from the cavity. Figure 5 shows a typical spectral reflectance curve for a green leaf and the slopes representing the two NDVIs expressed by equations (1) and (2).

Since chlorophyll in plants exhibits high absorption properties in the red region and low absorption properties in the NIR region, measuring the NDVI around the red edge (i.e., between the visible and the near-infrared regions) enables the health status and key properties of the plants to be identified. Note that intense sunlight might influence the NDVI value, and hence, a shield is typically used to minimise the impact of atmospheric conditions in real-time weed detection runs.

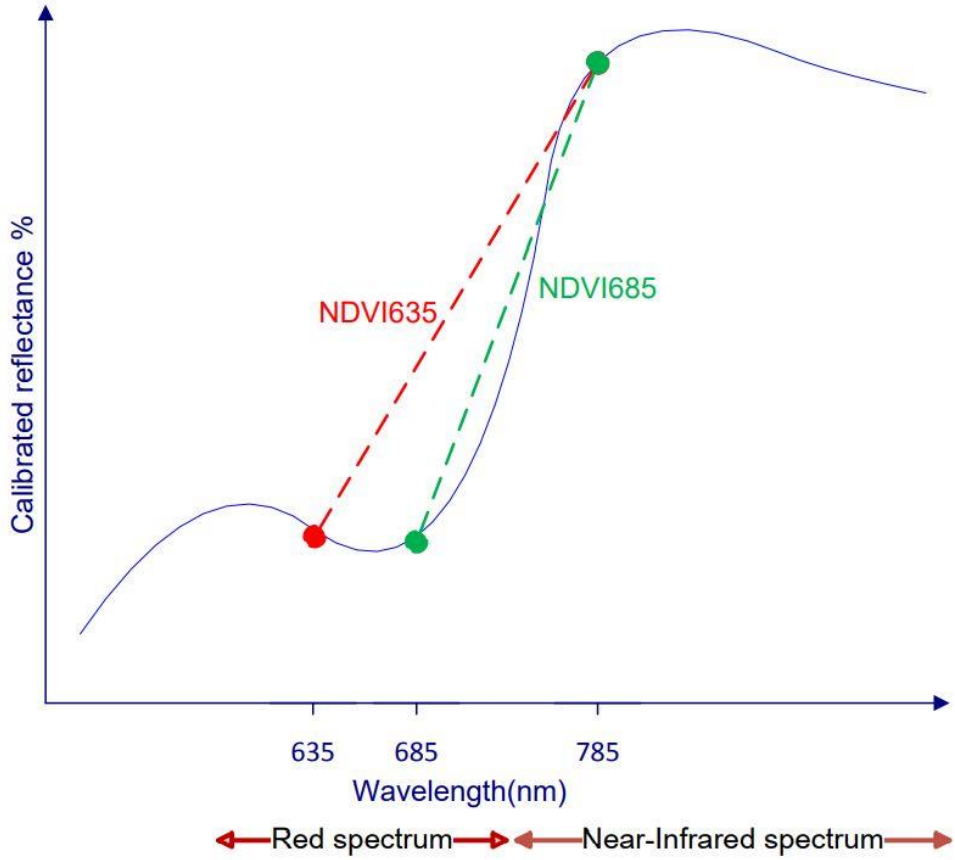


Figure 5. A typical spectral reflectance curve for a green leaf. The green and red slopes represent the two NDVIs expressed by Equations (1) and (2).

2.2.2.3 Input structure

Since the aim of the research was to compare the discrimination accuracies of machine-learning SVM methods based on the use of linear and Gaussian kernels with conventional NDVI-based discrimination methods, only the last beams emerging from the left and right cavities of the PDU (i.e., the central two beams) were

considered for data collection. The scanned span for each run was 550 mm, and the data was collected at 1 mm intervals.

Thirteen pixels were allocated for the calculation of the power of each detected laser beam. Since the laser beam typically has a Gaussian profile, the intensities of the reflected beams $\{P_{\lambda,1}, P_{\lambda,2}, \dots, P_{\lambda,13}\}$ detected by the 13 pixels were different in values, with one pixel of the 13-pixel set exhibiting the highest photocurrent, which corresponds to the peak power of the laser beam. A peak detection algorithm was specially developed to determine the maximum photocurrent, which enabled the peak power value of the reflected laser beams to be calculated.

The SVM algorithm was trained and tested with an NDVI input (generated from the measured right and left beams), given by

$$A_{right,NDVI} = \begin{bmatrix} NDVI_{635\_right,1} & NDVI_{635\_right,2} & \dots & NDVI_{635\_right,N} \\ NDVI_{685\_right,1} & NDVI_{685\_right,2} & \dots & NDVI_{685\_right,N} \end{bmatrix} \quad (4)$$

$$A_{left,NDVI} = \begin{bmatrix} NDVI_{635\_left,1} & NDVI_{635\_left,2} & \dots & NDVI_{635\_left,N} \\ NDVI_{685\_left,1} & NDVI_{685\_left,2} & \dots & NDVI_{685\_left,N} \end{bmatrix} \quad (5)$$

The SVM algorithm was also trained and tested with raw data input, given by

$$A_{right,raw} = \begin{bmatrix} \overline{P_{635_{right},1}} & \overline{P_{635_{right},2}} & \overline{P_{635_{right},N}} \\ \overline{P_{685_{right},1}} & \overline{P_{685_{right},2}} & \overline{P_{685_{right},N}} \\ \overline{P_{785_{right},1}} & \overline{P_{785_{right},2}} & \overline{P_{785_{right},N}} \end{bmatrix} \quad (6)$$

$$A_{left,raw} = \begin{bmatrix} \overline{P_{635_{left},1}} & \overline{P_{635_{left},2}} & \overline{P_{635_{left},N}} \\ \overline{P_{685_{left},1}} & \overline{P_{685_{left},2}} & \overline{P_{685_{left},N}} \\ \overline{P_{785_{left},1}} & \overline{P_{785_{left},2}} & \overline{P_{785_{left},N}} \end{bmatrix} \quad (7)$$

where

$$\overline{P_{\lambda,i}} \triangleq [P_{\lambda,1,i} \quad P_{\lambda,2,i} \quad \dots \quad P_{\lambda,13,i}] \quad (8)$$

Figure 6 shows the flowchart used to calculate the NDVI and raw data inputs.

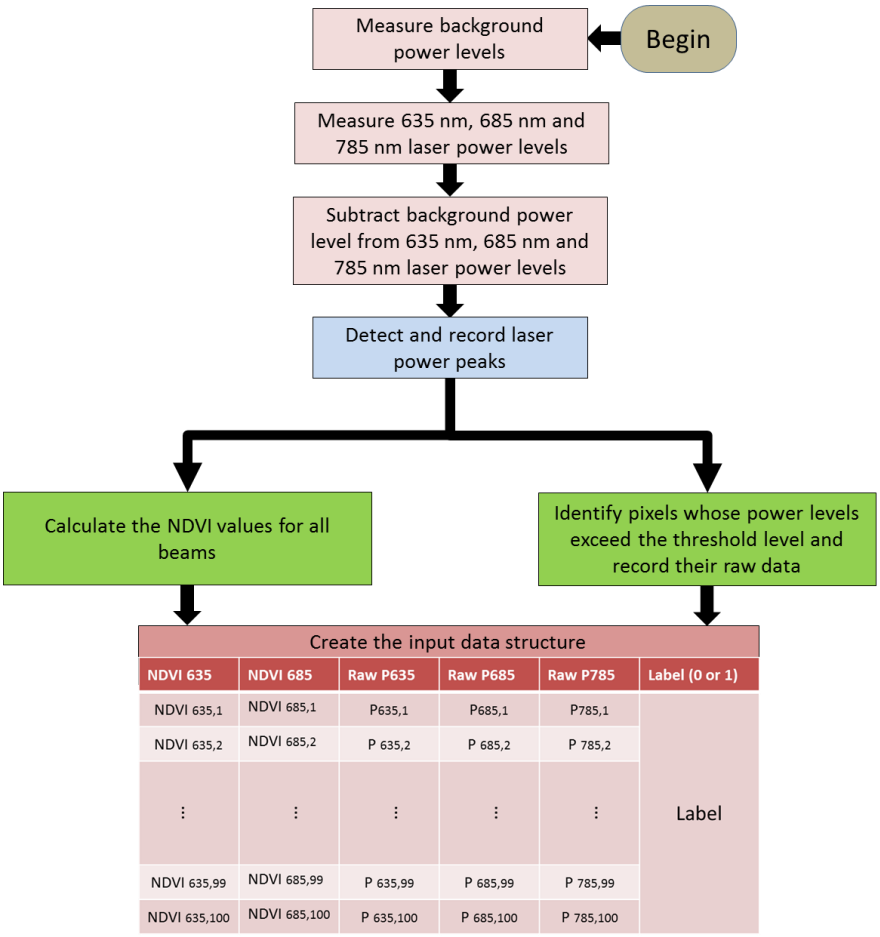
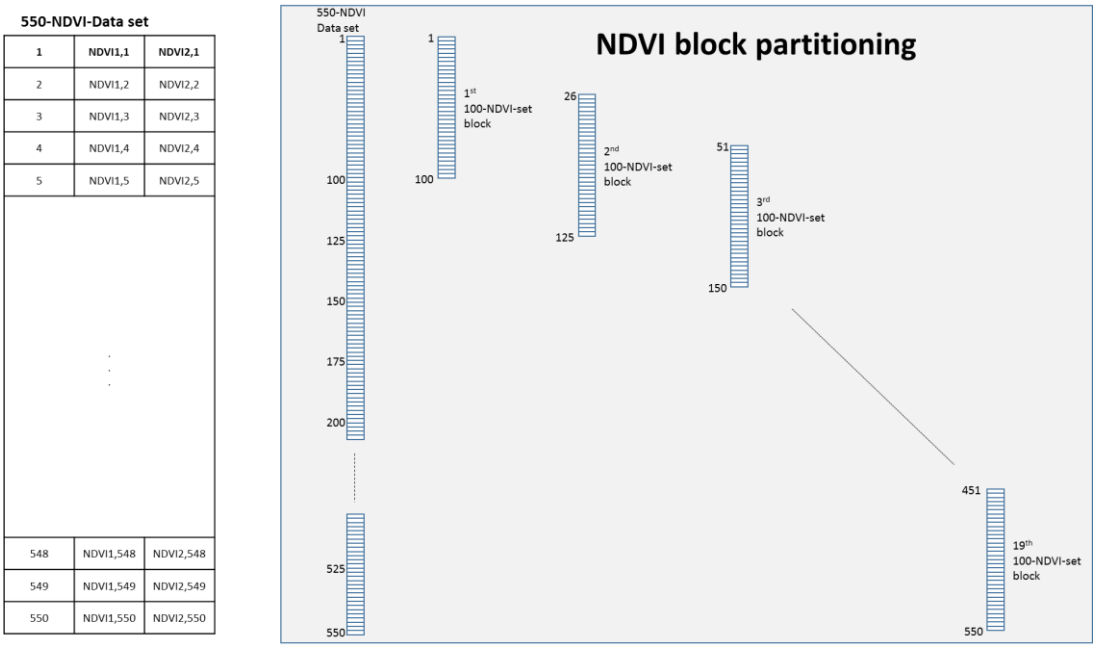


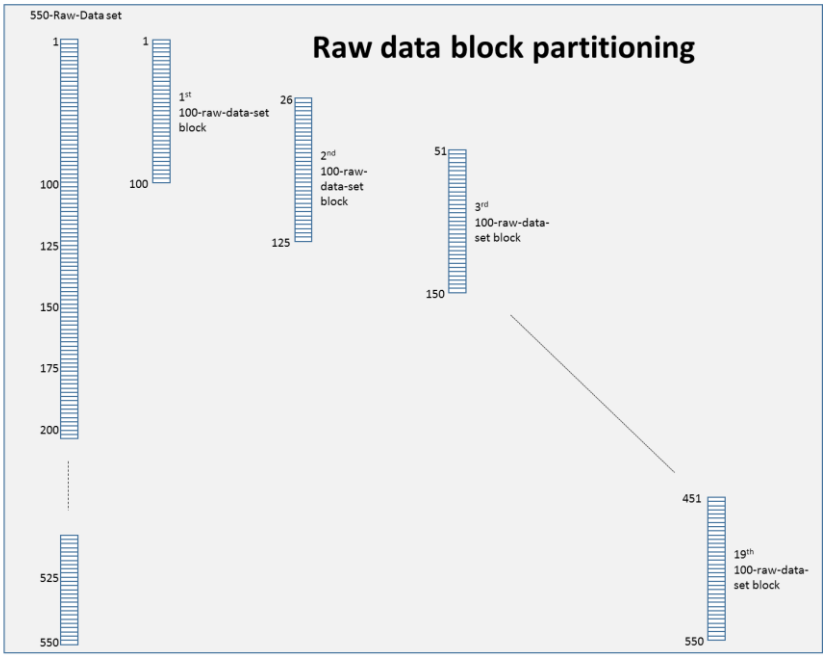
Figure 6. Flowchart illustrating the input data structure based on the calculation of the NDVI and raw data inputs. This structure is used for training and testing the SVM and parallelogram algorithms.



(a)

550-Raw-Data set

1	P635,1	P685,1	P785,1
2	P635,2	P685,2	P785,1
3	P635,3	P685,3	P785,3
4	P635,4	P685,4	P785,4
5	P635,5	P685,5	P785,5
548	P635,548	P685,548	P785,548
549	P635,549	P685,549	P785,549
550	P635,550	P685,550	P785,550



(b)

Figure 7. (a) NDVI-set block partitioning, and (c) Raw data set block portioning. Partitioning into 100-data-set block is used to generate Matlab®-based input structure of data.

2.2.3 Data Analysis

Three 15-cm-diameter pots, containing either silver beet or corn plants, were distributed over a span of 550 mm. The size of the input data set for each scan was 550, corresponding to a spatial resolution of 1 mm. Since the material filling the gaps between the pots was soil (not green), data-set blocks that exhibited less than 15% green content (i.e., where the soil content was dominant) were discarded. This reduced the number of data-set blocks from 19 to between 7 and 12. Note that species were considered green when their NDVI values were in the range 400 to 800; otherwise, they were considered soil [79]. Data augmentation was subsequently carried out by randomly rotating the plant pots in the test rig 10 times, thus increasing the input data-set blocks to 70-120. The data augmentation procedure used for training and testing the algorithm is illustrated in the flowchart in Figure 8.

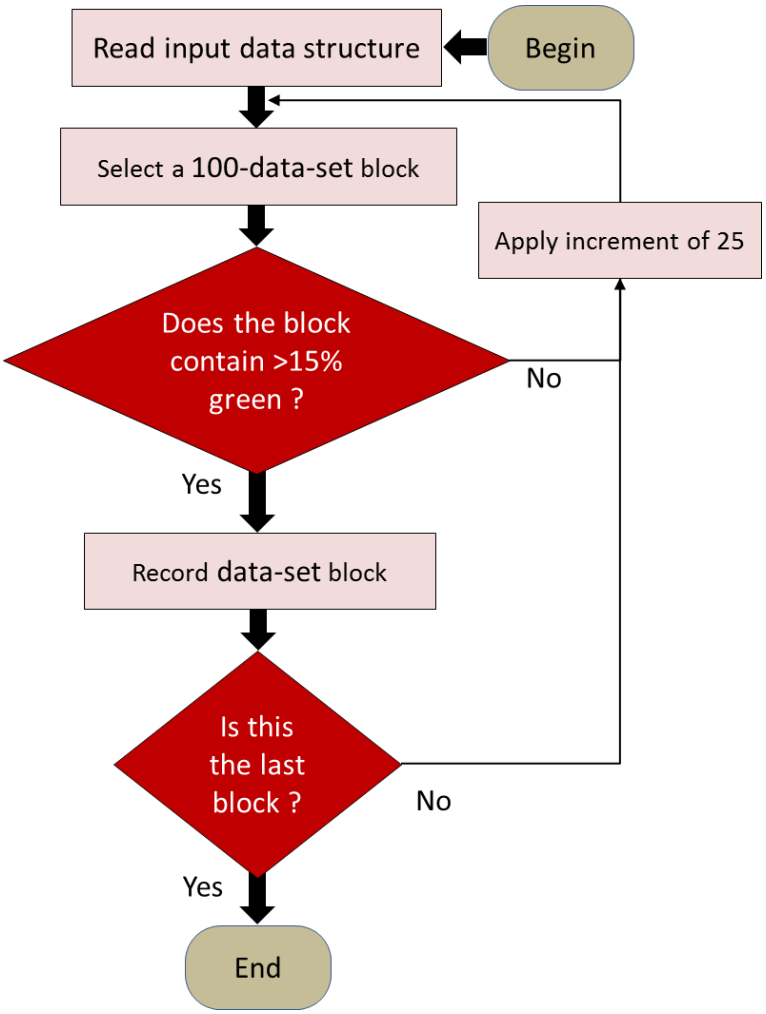


Figure 8. Flowchart showing the procedure of saving only the NDVIs and raw data sets for which the green content exceeds 15%..

2.2.4 Algorithms

SVM algorithms were applied to the NDVI values as well as the raw data to compare the performance of the dual-NDVI-based plant discrimination algorithm recently reported by Symonds, et al. [74].

2.2.4.1 Dual-NDVI-based plant discrimination algorithm

The dual-NDVI-based plant discrimination algorithm has recently been used to classify green plants. It is based on measuring two NDVI values at different wavelengths ($NDVI_{635}$ and $NDVI_{685}$) and generating a scatter NDVI plot for each plant. Scatter plots generated by Symonds et al. have shown that for a particular green leaf, the NDVI values typically fall within a parallelogram region [74], and that

the discrimination of a plant A from a plant B is only possible when their NDVI parallelogram regions do not overlap. The boundaries of the parallelogram region for each plant are typically determined through trial and error (the criterion is based on minimising the false negatives). This trial-and-error approach is time consuming and generally inaccurate, because the calculation of the boundaries of the parallelogram region is user-dependent.

In this section, we modify the approach for defining the boundaries of the parallelogram region by calculating the statistical properties of the NDVIs of the training data – namely, the mean values and standard deviations of the NDVI values, and by using linear regression to evaluate the slope of the parallelogram. To accurately evaluate the boundaries of the parallelogram region, all measured NDVI values that fell outside 400 and 800 (which corresponded to soil and non-green objects) were discarded. **Error! Reference source not found.** graphically illustrates the steps used to define the boundaries of the parallelogram region for a plant. The coordinates of the centre of the parallelogram region are the mean values of $NDVI_{635}$ and $NDVI_{685}$. The breadths of the parallelogram region are the standard deviations of $NDVI_{635}$ and $NDVI_{685}$, and the slope is defined through linear regression.

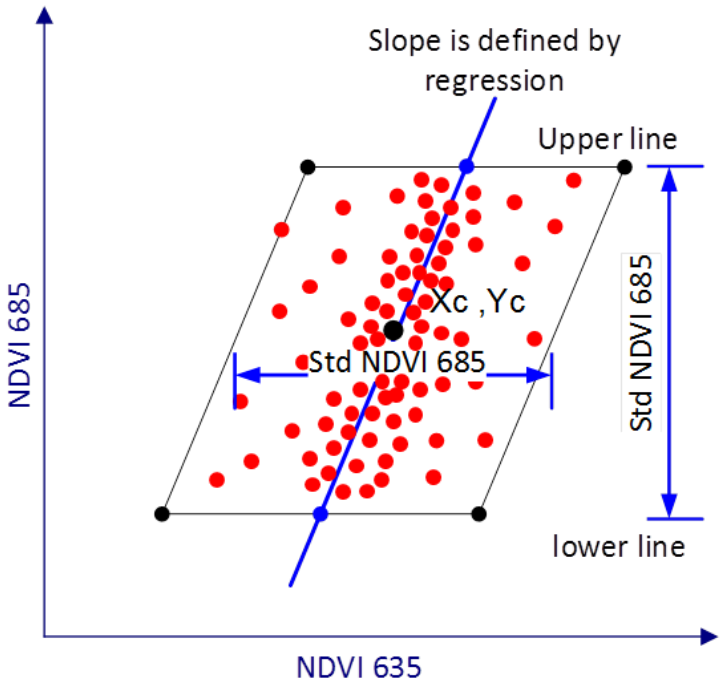


Figure 9. Illustration of the steps used to define the boundaries of the parallelogram region for a plant. The coordinates of the centre of the parallelogram region are the mean values of the measured $NDVI_{635}$ and $NDVI_{685}$. The widths of the parallelogram region are the standard deviations of the measured $NDVI_{635}$ and $NDVI_{685}$, and the slope is defined through linear regression.

2.2.4.2 Support Vector Machine (SVM) algorithm

An SVM algorithm is based on defining a surface in a multi-dimensional input space and maximising the arithmetical margin between two input data sets [31].

A label was first given to each input data according to the type of data (e.g., corn or silver beet in the case of plants). Then, the testing and training datasets were separated. The training datasets were shuffled randomly before training. The SVM algorithm was implemented in MATLAB Version 9.1.0.441655 (R2016b). Matlab Built-in function of fitcsvm was applied to the shuffled training dataset to train binary classification. Kernel Function was adjusted to linear and Gaussian. The kernels were adjusted on auto scale to select the appropriate scale using Matlab heuristic procedure. Sequential Minimal Optimization (SMO) was used to solve the quadratic programming (QP) problem.

2.3 Results

The performances of all algorithms were compared by calculating, for each algorithm, a confusion matrix, which is a table showing the number of true positives and negative as well as false positives and negatives after prediction.

2.3.1 Comparison of dual-NDVI-based plant discrimination and SVM algorithms with NDVI inputs

A set of 100 dual-NDVI sets (corresponding to a linear distance of 100 mm) was used in conjunction with the dual-NDVI-based plant discrimination algorithm for the classification of corn and silver beet. For this algorithm, the discrimination of plant X from plant Y was based on the following criteria: After generating the scatter plot, if the data counts falling within the parallelogram region of plant X exceed those falling within the parallelogram region of plant Y, then the detected plant is X. Otherwise, the detected plant is Y.

On the other hand, for the Gaussian SVM classifier, 100 consecutive dual-NDVI measurements were used as input during training, and a hyper-sphere for non-probabilistic binary classification was formed. Tables 1 and 2 show the confusion matrices for the dual-NDVI-based plant discrimination and SVM algorithms.

In all tables, the following terminology is used: True Positive (TP) for a plant that was correctly identified as a crop (corn); True Negative (TN) for a plant that was correctly not recognised as a crop (silver beet); False Positive (FP) for a plant incorrectly identified as a crop, False Negative (FN) for a plant incorrectly not recognised as a crop.

Note that for algorithm training, out of the 190 augmented data-set blocks, 73 data-set blocks exhibited more than 15% green content for corn, and 74 data-set blocks exhibited more than 15% green content for silver beet. On the other hand, for algorithm testing, out of the 190 augmented data-set blocks, 115 and 73 data-set

blocks exhibited more that 15% green content for corn and silver beet, respectively. Therefore, in order to compare the accuracies of the algorithms, 73 data-set blocks were selected for both training and testing.

It is obvious from Tables 1 and 2 that while both algorithms produced almost similar numbers of true positives and false negatives, the Gaussian-kernel SVM algorithm predicted less false positives (10) and higher true negatives (63) than the dual-NDVI-based plant discrimination algorithm (39 and 34, respectively). Tables 1 and 2 also show that out of the 146 data-set blocks (73 for corn and 73 for silver beet), the dual-NDVI-based plant discrimination algorithm predicted 96 as corn and 50 as silver beet, whereas the SVM algorithm predicted 82 as corn and 64 as silver beet.

Table 1. Confusion matrix based on using NDVI values for the dual-NDVI-based plant discrimination algorithm.

		Predicted Plants		
		Corn	Silver beet	Total
Actual Plants	Corn	TP = 57	FN = 16	73
	Silver beet	FP = 39	TN = 34	73
	Total	50	50	146

Table 2. Confusion matrix based on using the Gaussian-kernel SVM algorithm with NDVI values as input.

		Predicted Plants		
		Corn	Silver beet	Total
Actual Plants	Corn	TP = 56	FN = 17	73
	Silver beet	FP = 10	TN = 63	73
	Total	66	80	146

The sensitivity, specificity, precision and accuracy of a discrimination algorithm are as following [80]:

$$Sensitivity = \frac{TP}{(TP+FN)} \quad (9)$$

$$Specifity = \frac{TN}{(TN+FP)} \quad (10)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (11)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (12)$$

Error! Reference source not found. shows the sensitivities, specificities, precisions and accuracies attained using the dual-NDVI-based plant discrimination and SVM algorithms. It is obvious from Figure 10 that the SVM algorithm outperforms the dual-NDVI-based plant discrimination algorithm, exhibiting significant improvements in specificity (85%), precision (43%) and accuracy (31%).

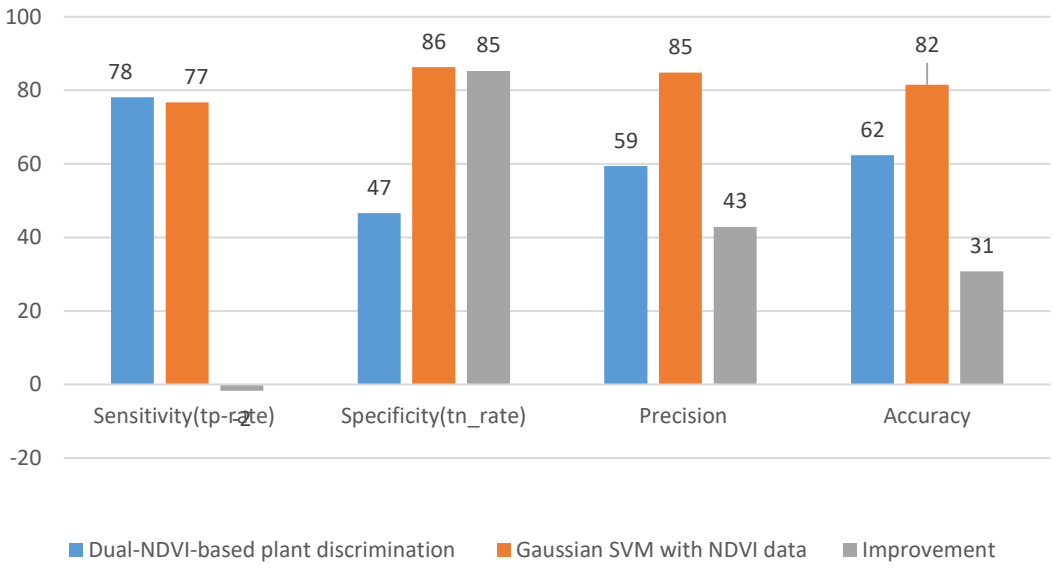


Figure 10. Sensitivities, specificities, precisions and accuracies of the dual-NDVI-based plant discrimination and SVM algorithms.

2.3.2 SVM classifier performance comparison using NDVI and raw input data

In this section, we focus on the improvement in the performance of the SVM algorithm when raw data, rather than NDVI data, was used as input. The intensities of the beams reflected off the investigated vegetation (raw data) and the corresponding NDVI values were stored in an input data structure. This enabled (i) training of the SVM algorithm with input raw as well as NDVI data and (ii) testing it to assess its performance for both raw and NDVI input data. Note that the raw data sets are given by Eqs (6-8), whereas the NDVI

data sets are given by Eqs (4-5). Table 3 shows the confusion matrix for the SMV algorithm, obtained using raw data as input. It is obvious from Table 3 that out of the 146 test data sets (73 for corn and 73 for silver beet), the SVM algorithm predicted 76 as corn (73 correctly and 3 incorrectly) and 70 as silver beet (all correctly).

Table 3. Confusion matrix based on using the Gaussian-kernel SVM with raw data as input

		Predicted Plants		
		Corn	Silver beet	Total
Actual Plants	Corn	TP = 73	FN = 0	73
	Silver beet	FP = 3	TN = 70	73
	Total	76	70	146

Error! Reference source not found. shows the sensitivities, specificities, precisions, and accuracies of the Gaussian-kernel SVM algorithm, for both raw and NDVI input data. It is clear from Figure 11 that by using raw data, rather than NDVI data, as the input for the SVM algorithm, significant improvements in sensitivity (11%), specificity (13%), precision (22%), and accuracy (20%) are achieved.

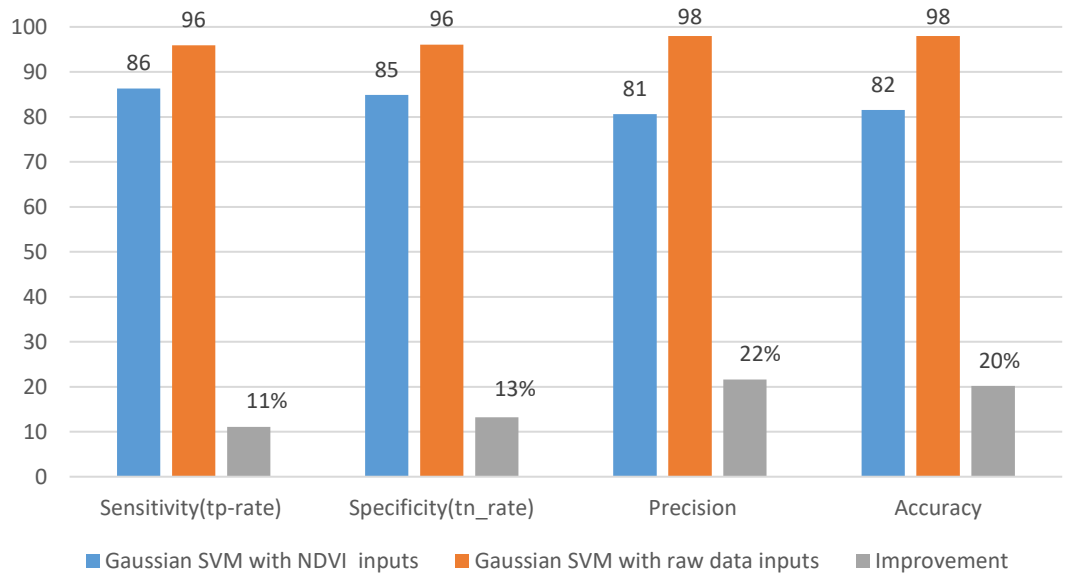


Figure 11. Sensitivities, specificities, precisions, and accuracies attained by the SVM algorithm for both NDVI inputs and with raw data.

2.4 Discussion

The main finding of the current study was that using the linear SVM algorithm with NDVI values as inputs enables significant improvements in specificity, precision, and accuracy compared to the conventional dual-NDVI-based plant discrimination algorithm (even when the boundaries of the plant-scattering plot parallelograms were defined automatically, based on measuring the means and standard deviations of the NDVI values). A possible explanation for this is as follows: Both the dual-NDVI-based plant discrimination and the linear SVM algorithms use a set of 200 data for input (100 for the $NDVI_{635}$ and 100 for the $NDVI_{685}$). However, for the dual-NDVI-based algorithm, the $NDVI_{635}$ and $NDVI_{685}$ values are defined independently in a two-dimensional space, and plant identification is based on aggregating the 100 readings for each NVDI value. In contrast, the SVM algorithm considers a set of 200 inputs in a 200-dimensional space. This enables better plant classification, since the correlation between the patterns of the NDVI values can be evaluated while the SVM is being trained and tested.

In addition, the number of input data per set for the NDVI-based SVM algorithm was 200, while that for the raw-data-based SVM algorithm was 7800 (200 data sets ×13 intensities per laser source × 3 laser sources). Having a space with a larger number of dimensions not only helps the SVM algorithm to be trained with more data patterns, but also reduces the errors arising from different noise sources, such as the fluctuations in sensor data or incorrect peak detection.

Chapter 3: A statistical approach to provide explainable convolutional neural network parameter optimization

This chapter was published as an article in the International Journal of Computational Intelligence Systems, Volume 148, December 2019 [2], Pages 1635 – 1648. This article appears as it does in print, with the exception of minor changes to the layout, number formats, font size and font style, and introduction and conclusion removal which was implemented to maintain consistency in the formatting of this thesis.

3.1 Abstract

Algorithms based on convolutional neural networks (CNNs) are attracting great attention in image processing due to their ability to find patterns and recognize objects in a wide range of scientific and industrial applications. Finding the best network and optimizing its hyper-parameters for a specific application are one of the central challenges of CNNs. Most state-of-the-art CNNs are manually designed, while techniques for automatically finding the best architecture and hyper parameters are computationally intensive, and hence, there is a need to severely limit their search space. This study proposes a fast statistical method for convolutional neural network parameter optimization, which can be applied in many CNN applications and provides more explainable results. We specifically applied Taguchi based experimental designs for network optimization on our basic network, a simplified inception network and a simplified Resnet network, and conducted a comparison analysis to assess their performances and select the hyper parameters and networks that facilitate faster training and provide better accuracy. Results show that before and after parameter optimization up to a 5 percent increase in classification accuracy can be achieved.

3.2 Experimental methods

The proposed primitive cells, that are the building blocks of the proposed CNN network, are described and compared with those used in the current state-of-the-art conventional CNN network architectures. Next, the Taguchi method is introduced, and four experiments conducted to optimize the various parameters of the CNN network.

3.2.1 The convolutional neural network primitives

Most CNNs require a huge memory and computational resources. Therefore, the size of the CNN becomes important for applications running on limited resources (e.g., limited speed and memory of mobile phones). In this study, in order to build up an efficient CNN network, primitive blocks from well-known CNN architectures are adapted and used, namely, the VGG network, the Inception network and the Resnet network [40, 41, 43].

3.2.1.1 Simple primitive block

The simple primitive block of the proposed CNN was designed to include a convolutional layer with batch normalization, Rectified Linear Unit (ReLu) function, and max pooling. The advantage of using ReLu instead of traditional neural network functions, like sigmoid or hyperbolic, is that training is typically faster. The ReLu formula is f(x)= max(0,x) [75]. The batch normalization, which is normalization of the network's weights in each layer, has many advantages, including faster learning-rate convergence, efficient weight initialization, and additional regularization, all of which result in faster training with lower overfitting [81]. Max pooling reduces the computation by extracting the most important features from a kernel, hence producing a smaller feature representation [82, 83].

3.2.1.2 The Inception primitive blocks

The Inception network is based on the concept that many activations in deep convolution layers are either excessive (zero value) or redundant (highly correlated). Therefore, instead of going

deeper in the CNN network, the balance between network width and depth should be kept in order to maintain optimal performance. The necessity of having efficient dimension reduction leads to a sparse network with various filter sizes. As a result, the required computational reduction leads to bottleneck convolution [84]. Szegedy, et al. [43] have suggested that replacing n × n convolution by asymmetric convolutions with lower kernels and having generally n × 1 convolution after 1 × n convolution can significantly save the computational costs. They have also noted that, applying such factorization on early layers is not very effective. Therefore, to build Inception networks, we first used the two of the simple primitive blocks (described in Section 2.1.1) as the CNN base followed by one of the five asymmetric Inception blocks shown in **Error! Reference source not found.**.

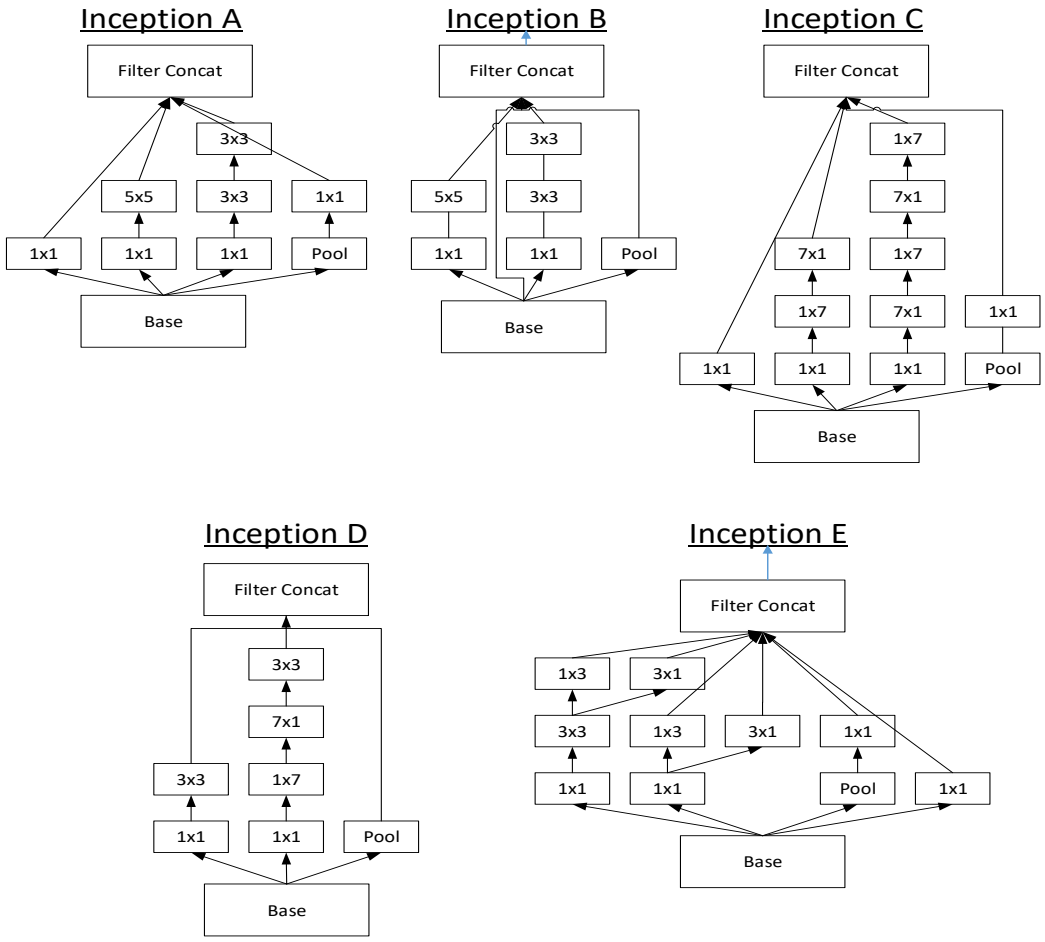


Figure 12. Five asymmetric configurations used inception blocks for primitive blocks.

3.2.1.3 The Resnet primitive block

The Resnet network was designed to address the challenge of overcoming the vanishing gradient problem in order to build deeper CNNs[41]. Instead of simply stacking layers, the middle layer learns the residual mapping of the input. This is done by adding a residual block as the identity connection from previous layers, thus allowing the residual mapping to be trained, [41], as illustrated in Figure 13.

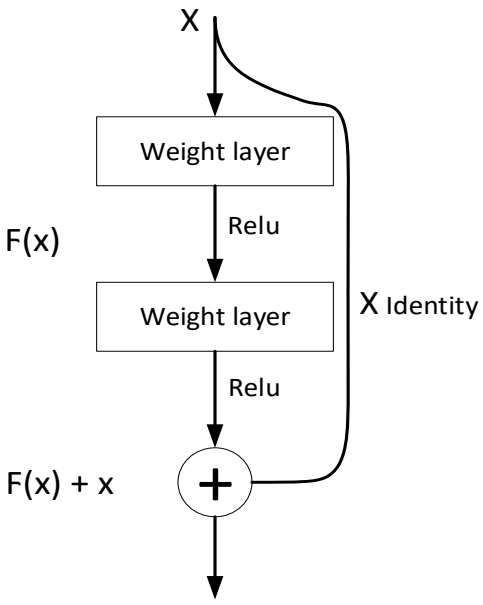


Figure 13. Illustration of a residual block where the middle layer learns the residual mapping of the input, thus allowing the residual mapping to be trained [41].

In the current study, two simple primitive blocks were used as the first layers. Following this, residual primitive layers were added to the primitive blocks. In this structure, each residual layer can build up the desired residual primitive block via down sampling from its previous layers.

3.2.2 Taguchi method

The Taguchi method is a statistically robust technique, proposed by Geninchi Taguchi [85], which fulfils two main roles, namely, finding factors that provide more variation in responses, and

finding the best levels of these factors that optimize the CNN response. In the Taguchi method, initially a decision is made on which factors can be effective for the required response. Then, the levels of those factors are set. In Figure 14, the Taguchi process block diagram is shown, and the initial process of finding the effective factors is depicted as the preparation block.

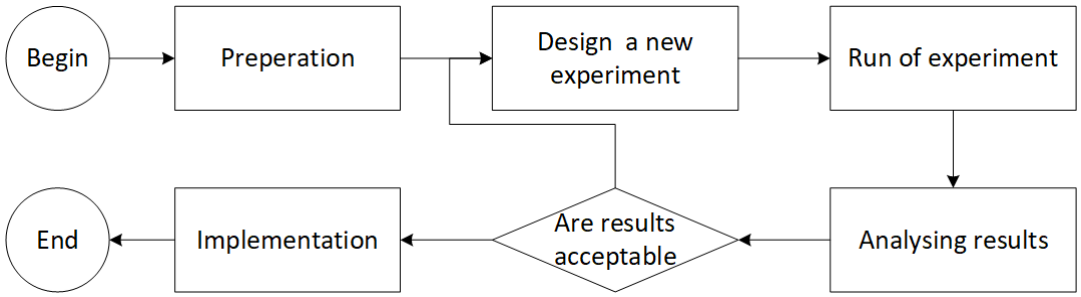


Figure 14. The process of implementation of Taguchi design.

The next step in the Taguchi procedure is designing the experiment. This is based on the orthogonal matrix, determining in each run the levels of the control factors that must be taken. In the statistical experimental design, orthogonality involves providing a matrix of runs that have statistically independent factors in their columns, where the levels in the columns of the independent factors are orthogonal to one another. Consequently, if the factor levels are considered as a vector, the inner product of mutual factor vectors in an orthogonal array must be zero [86].

Another property of the orthogonal matrix is that all the levels in each column must appear the same number of times. The above-mentioned two properties effectively reduce the number of experiments from full factorial design into a minimal number of runs, which still can provide enough knowledge in describing the effect of factors on the CNN performance. The minimum number of experiments needed to conduct the Taguchi method can be calculated based on the following equation:

$$N = \sum_{i=1}^{Number\ of\ variables} (Lev_i - 1) + 1 \qquad (1)$$

where N is the number of experiments and Lev_i is the number of levels in the ith factor.

The orthogonal matrices in a standard Taguchi design are predefined based on the number of controlling factors and levels. Consequently, the experiment is run according to the defined orthogonal array in order to determine and prepare the CNN responses for analysis. This step is shown as “design a new experiment” in Figure 14. The relative percentage difference (RPD) in accuracy was used as the CNN response indicator that enables better understanding of the impact of the factors on the performance of the CNN architecture. For the generated results, the RPD was calculated as follows:

$$RPD = \frac{x_i - x_{min}}{x_{min}} * 100 \quad (2)$$

where x_i is the responses obtained from the ith run of the experiment and x_{min} is the minimum response obtained.

The results were analysed by comparing the mean controlling factors and signal to noise ratios, as well as evaluating the analysis of variance (ANOVA), in order to investigate the significant factors. The preliminary design of the experiment was sometimes slightly changed after analysing the results mainly to determine the best controlling factor or the best levels of the controlling factors. After obtaining acceptable RPD results, the optimum level of each factor was implemented to obtain the best CNN according to defined objectives.

3.2.3 Design of experiment

Before determining the factors and levels for designing the CNNs, few preliminary designs were run to evaluate some hyper-parameters. Two controlling factors, namely batch size with level values 8, 16, 32, and number of epochs with level values 5, 10, 20, were examined. It was found that neither the batch size nor the number of epochs affected the RPD accuracy. Their p-values in the ANOVA were 0.23 and 0.71, respectively. However, it was necessary

to define their values in the network. As the larger batch size resulted in faster training and less over fitting, the selected batch size value was 32. Smaller values for the number of epochs typically led to faster running of the whole procedure of learning. Therefore, the chosen number of epochs was 5.

The following hyper parameters were selected as the factors in our search space: learning rate, augmentation types, number of filters, and size of kernel.

Learning rate: In order to have fast network convergence during training, the optimum selection of the learning rate was essential. When the chosen learning rate was too small, training took too long to converge or the optimizer became trapped in local minima, hence the loss function could not be updated to generalize the network. When the chosen learning rate was too large, the network did not always converge as it might have overpassed the minimum loss function, and hence, made the loss function worse.

Augmentation: Large data in deep learning normally yields better performance; however, it is not always possible to have large data. As neural networks have invariance characteristics, data augmentation can generate more data without the need for additional data collection. In order to select the type of data augmentation, it is necessary to consider the nature of the data. In the proposed study, the implemented data was linear spectrum reflectance. According to the nature of the data, flipping (horizontal and vertical) and normalization were the most suitable transformations for data augmentation.

Number of Filters and Kernel Sizes: In the first two layers of the first design, two simple primitive blocks were used. In the first layer, lower levels for the number of filters were used (4, 8, 16, 32), since the network typically extracts basic features in this layer. The larger levels for the number of filters were selected in the second layer (32, 64,128,256), because this layer combines the basic

features and produces more complex features for classification. The next factors investigated in the proposed study were the kernel sizes. The levels of 3 and 5 were selected for the investigation.

Four experiments were designed, based on (i) simplified primitive cells, (ii) modified inception cells, (iii) modified resnet cells, and (iv) an experiment comparing the performances of all three primitive cells.

The pytorch deep learning library was used for all trainings. GeForce GTX 1080Ti Nvidia GPU was the hardware platform for data processing, which has a 11 GB memory, a 352-bit interface used in conjunction with an Intel® Core™ i7-7800X X-series Processor.

3.2.3.1 First design of experiment

First experiment: Two simple structures were suggested for the first design, as shown in Figure 15. The first structure had one fully connected classification output layer with soft max activation. The second structure had an extra fully connected layer with a ReLu activation.

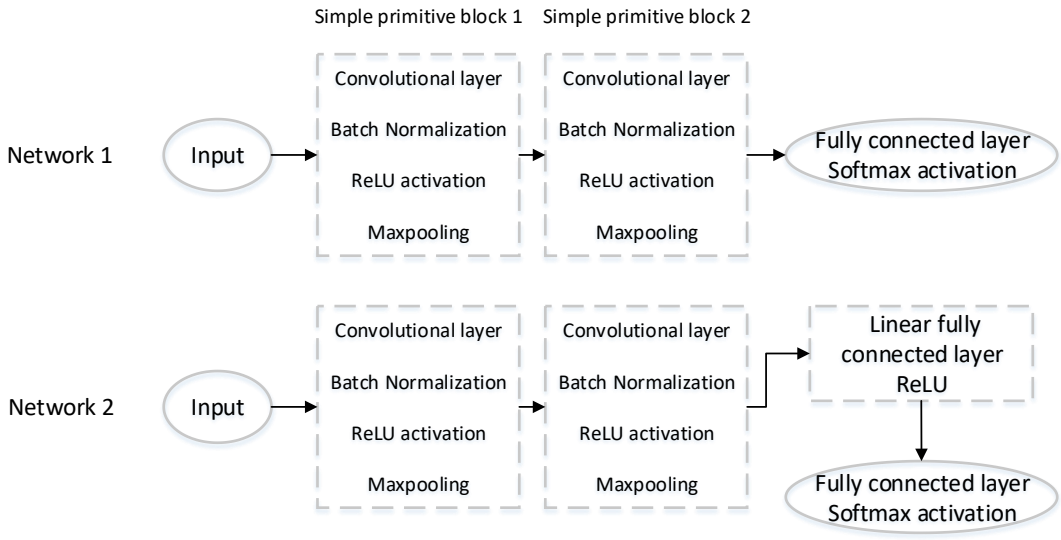
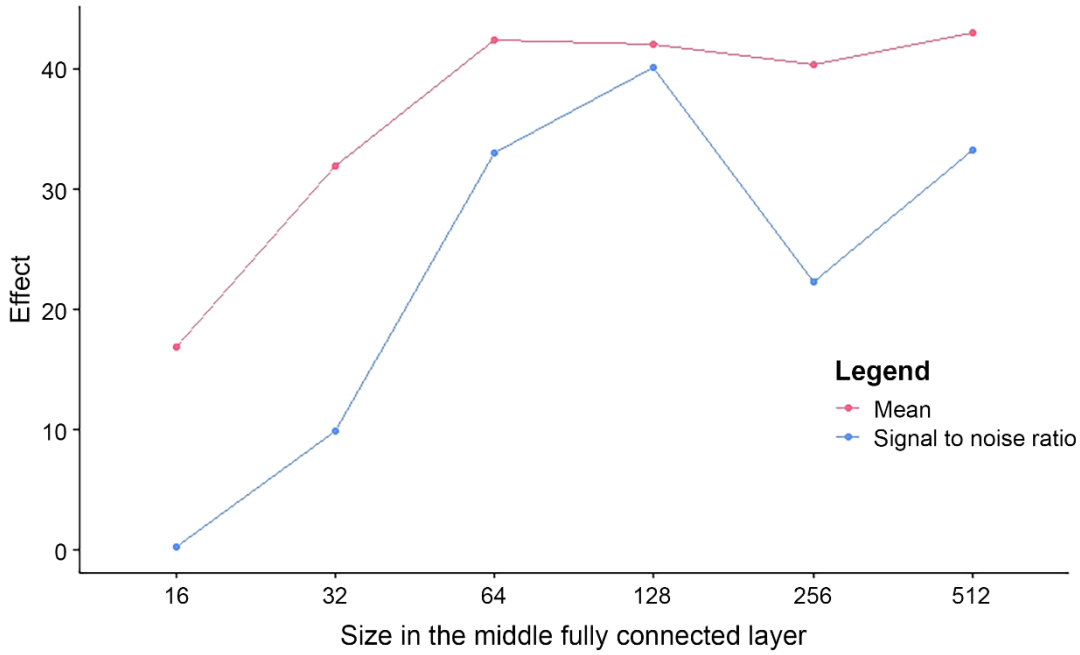


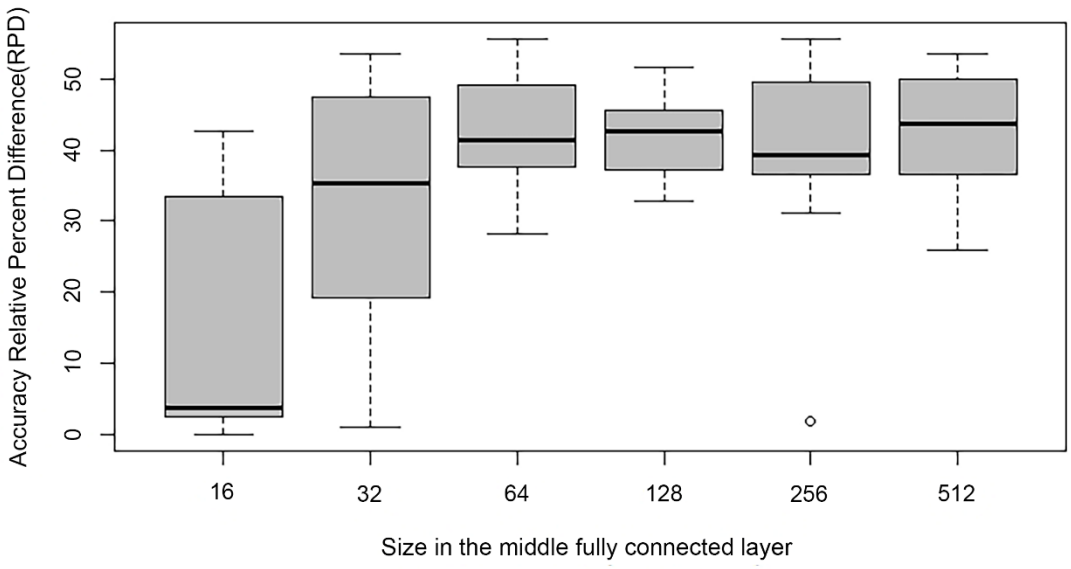
Figure 15. Two different CNN structures comprising simple primitive blocks.

The effect plot and accuracy versus the size of the middle connected layer, which are the results of the preliminary experiment

that was designed to choose the number of nodes in the extra fully connected layer, are shown in Figure 16.



(a)



(b)

Figure 16. Preliminary experiment results for determining the value of middle fully connected layer in the first design. (a) Effect and (b) accuracy versus the size of the middle connected layer.

Table 4. Analysis of variance for comparing the different number of nodes in the middle fully connected layer, the response was the relative percentage difference in accuracy.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Fully connected size	1	2388.6	2388.6	10.803	0.001457**
Residuals	88	19457.7	221.11		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

According to the Bonferroni post-hoc test, which measures the significance of the pairwise differences between factor levels, there was no significant difference between the levels with values of 32, 64, 128, and 256. Therefore the 128 value was selected based on the best signal-to-noise-ratio performance. With this result, the first DOE was designed as illustrated in Table 5.

Table 5. The design of experiment for the simple convolution network.

Control Factor	Learning rate	Type of augmentation	Number of filters in the first layer	Number of filters in the second layer	Kernel size in layer 1	Kernel size in layer 1	Type of network
Level 1	0.00001	No Augmentation	4	32	3	3	Without intermediate fully connected layer
Level 2	0.0001	Horizontal	8	64	5	5	
Level 3	0.001	Vertical	16	128			With intermediate fully connected layer
Level 4	0.01	Horizontal and Vertical	32	256			

3.2.3.2 Second design of experiment

The next DOEs were designed to examine the effect of using various Inception networks. The same controlling factors were

selected for the Inception networks, however, they were chosen in three levels, except the type of network, which was selected according to the various Inception design. These levels are shown in Table 6.

Table 6. The design of experiment for the Inception convolution networks.

Control Factor	Learning rate	Type of augmentation	Filters in the first layer	Filters in the second layer	Kernel size in layer 1	Kernel size in layer 2	Type of network
Level 1	0.00001	Horizontal and Vertical	4	32	3	3	Inception A
Level 2	0.0001	Vertical	8	64	5	5	Inception B
Level 3	0.001	Vertical and normalization	16	128	7	7	Inception C
Level 4							Inception D
Level 5							Inception E

3.2.3.3 Third design of experiment

Another design introduced here, was based on the Resnet network. This included two simple primitive blocks following two primitive residual layers. Each residual layer can downsample from its previous layer and produce the required number of residual blocks. The learning rate was used in exactly the same way as the previous design. There were two residual layers. In each residual layer, the number of blocks and the number of filters in each block and its kernel size were investigated to determine how effective they are and what are their optimum values for the current network and dataset. Various types of augmentation were also investigated, namely horizontal, vertical, and vertical and normalized augmentation, as illustrated in Table 7.

Table 7. The design of experiment for the residual convolution network.

Control Factor	Learning rate	Number of Resnet blocks used in both Resnet layers	Augmentation	Number of filters in the first Resnet layer	Number of filters in the second Resnet layer	Kernel size in layer 1	Kernel size in layer 2
Level 1	0.00001	1	Horizontal	8	32	3	3
Level 2	0.0001	2	Vertical	16	64	5	5
Level 3	0.001	3	Horizontal and vertical	32	128	7	7
Level 4		4					
Level 5		5					
Level 6		6					

3.2.3.4 Fourth design of experiment

The last DOE was designed to compare the simple network, the Inception network, and Resnet. According to our previous investigation, the most effective levels for the learning rate were chosen to be (0.01, 0.001, 0.0001). Additionally, the appropriate number of filters in each layer and kernel sizes were selected accordingly. All types of augmentation introduced in the previous designs were researched in this design as well. All those factors that results from the design are shown in Table 8.

Table 8. Design of experiment for comparison of Resnet, Inception and simple network

Control Factor	Learning rate	Type of augmentation	Filters in the first layer	Filters in the second layer	Kernel size in layer 1	Kernel size in layer 2	Type of network
Level 1	0.00001	Horizontal	8	32	3	3	Simple network
Level 2	0.0001	Vertical	16	64	5	5	Inception network
Level 3	0.001	Vertical and normalization	32	128	7	7	Resnet
Level 4		Horizontal and normalization					
Level 5		Horizontal and vertical					
Level 6		horizontal and vertical and Normalization					

According to the number of factors and levels, the orthogonal array L16 was chosen for the first Design since it has four factors with four levels, and three factors with two levels. The orthogonal array L18 was chosen for the following designs because it has six factors with one three levels and one factor with six levels.

In order to significantly reduce the possibility of random results, all the designs were run with five replications. The replication was produced by using a random seed to ensure the same results are produced in future runs. All the DOEs and results were designed and analysed in Software R, Version 3.5.1 (2018-07-02).

3.3 Results and discussions

3.3.1 Evaluation metrics

Since for an orthogonal array the design is balanced, the factor levels are weighed equally, making the effects of factors independent from one another. Therefore, each factor can be assessed independently, hence reducing the time and cost of the experiment. This characteristic helps to easily compare the mean of each level in a particular factor to determine how effective each level is in changing the response. This characteristic can typically be displayed through the mean in the effect plot. Similarly, the signal-to-noise ratio is another metric that can evaluate the factor levels. In the current study, the aim was not to only determine which factor levels were effective in producing better accuracy, but also to ensure stable factor levels in order to generalize these factors for future classification problems. Therefore, the nominal-the-best approach was chosen to measure the signal-to-noise ratio as it evaluates the levels around the mean and considers changing other factors according to the orthogonal array. The signal-to-noise ratio can be calculated as follows:

If we consider the data to be $y_1, y_2, \dots, y_n$ then

$$\frac{S}{N} = 10 * log \frac{\bar{y}^2}{s^2} \quad (3)$$

where $\underline{y}$ and s are the average and standard deviation of the data, respectively.

Finally, ANOVA was used to analyse the results. Since the aim of the proposed study was to reduce the computational cost while improving the accuracy, if a factor was significantly important, the level that produced the best accuracy was selected. Similarly, if the factor was not statistically significant, the optimum level of the factor was selected from the effect plots in order to reduce the computational cost.

3.3.2 Designs evaluation

3.3.2.1 Design 1

Figure 17 shows the effect plots of mean and signal-to-noise ratio of controlling variables introduced in the first design (design of experiment for the simple CNN).

The first controlling factor was the learning rate. Since the aim of learning rate is to control and adjust the network weights to reduce network loss, having appropriate initial learning rate helps the network to converge faster, hence reducing the computation time. According to the ANOVA, the effect of the learning rate was significant. Similarly, the effect plots demonstrated that the best level of learning rate was 0.0001. For the rest of the DOEs, the best obtained leaning rate was still 0.0001.

The next factor investigated was the augmentation type. In the first design, four types of augmentation were selected; vertical, horizontal, vertical and horizontal, and no augmentation. These augmentations were chosen according to the nature of the data. The data applied in the current study were three spectral laser reflectance values (at three different wavelengths) collected by plant discrimination unit (PDU). PDU collects linearly three laser reflectance values [1]. Accumulation of these linear responses resulted in three layers of two-dimensional inputs. Wild leaf plants (Canola and Radish) were considered as two input classes. Two hundred plants were grown. 2,400 spectral data were collected for each plant stage. Data were collected in five different stages, three days after early germination for five consecutive weeks. A total of 12,000 data were generated for each class. 60% of the data set was kept for training and the rest was equally divided for validation and testing. Training, validation and testing plants were kept strictly separated to ensure the generalization of the study. A sample data used for training is shown in Figure 18. Vertical augmentation changed the order of the laser beams from end to beginning. The

horizontal augmentation changed the direction of 2D arrays of data. The results show that the combination of horizontal and vertical augmentation significantly changed the value RPD accuracy.

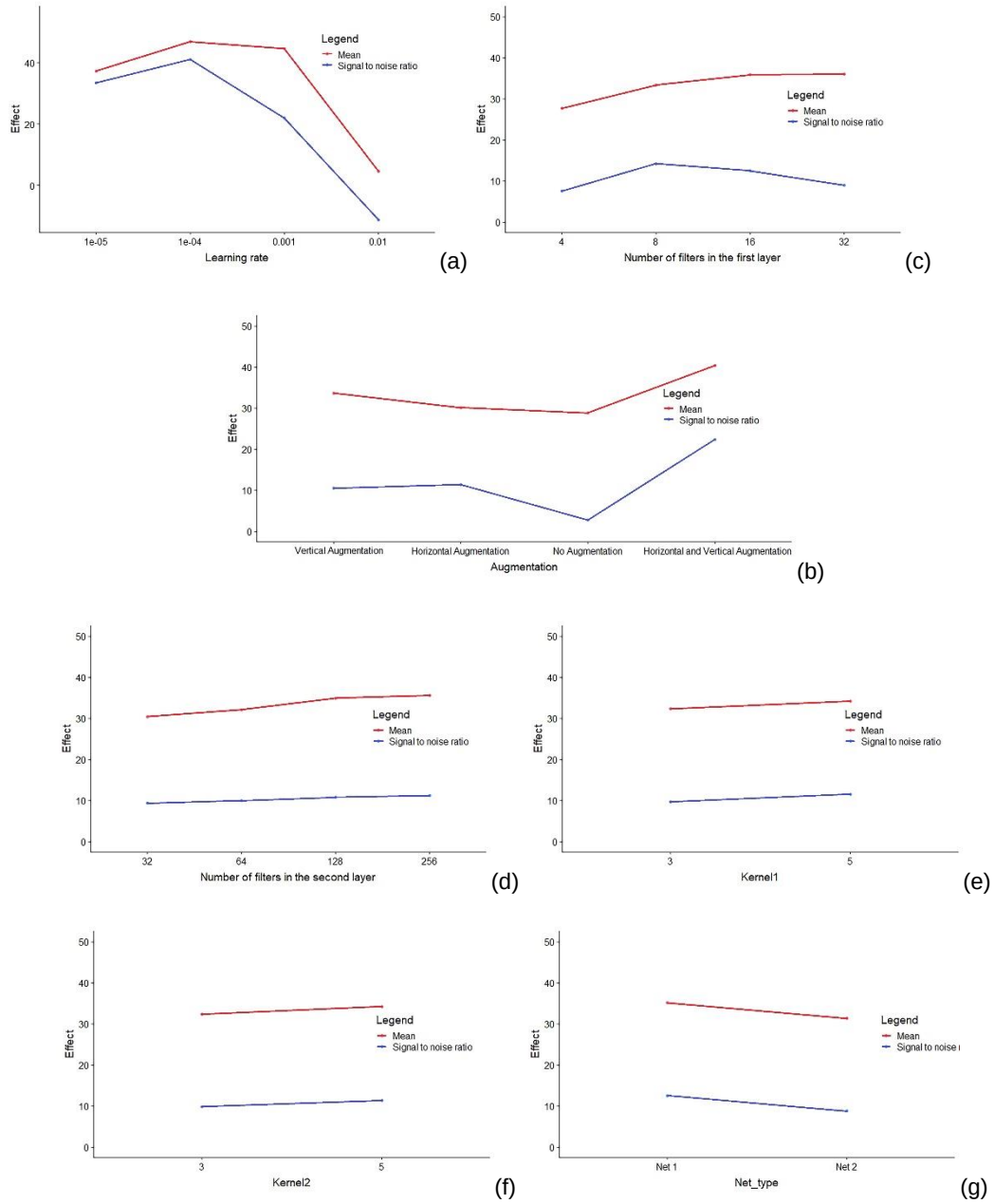


Figure 17. Effect plots of mean and signal to noise ratio for seven controlling factors of the simple CNN used in the first design. Each plot is for one of the controlling factors, namely: A)Learning rate, B)Type of augmentation C)Number of filters in the first simple primitive block D)Number of filters in the second simple primitive block E)Convolution kernel size in the first block F)Convolution kernel size in the second block G)Type of network (Net1 represent the simple CNN without middle fully connected layer and Net2 represent simple CNN with middle fully connected layer).

The role of filters in the first layer was to help learning the basic features and in the following layer and combine these features

to produce deep learning according to more complex features. When there were not much basic features in the data, the number of filters in the first layer could be reduced to minimize the computational cost. Results showed that with 16 or 32 filters in the first layer and 128 or 256 filters in the second layer, better RPD accuracy could be obtained. While the difference between the number of filters in the first layer and second layer were not statistically significant, the interaction between the filters in these two layers provided significant differences in RPD accuracy.

The next two controlling factors introduced were the kernel sizes. According to effect plots, there was slight improvement using 5x5 kernel rather than 3x3 kernel, due to nature of the data. As can be seen from Figure 18, there is gap between laser reflectance responses. This is because at the end of each block there was a max pooling layer which returned the maximum of a 2x2kernel, where this gap did not provide much information for the network. Therefore, 5x5 kernels were able to provide slightly better accuracy than the 3x3 kernels, however, this difference was not significant.

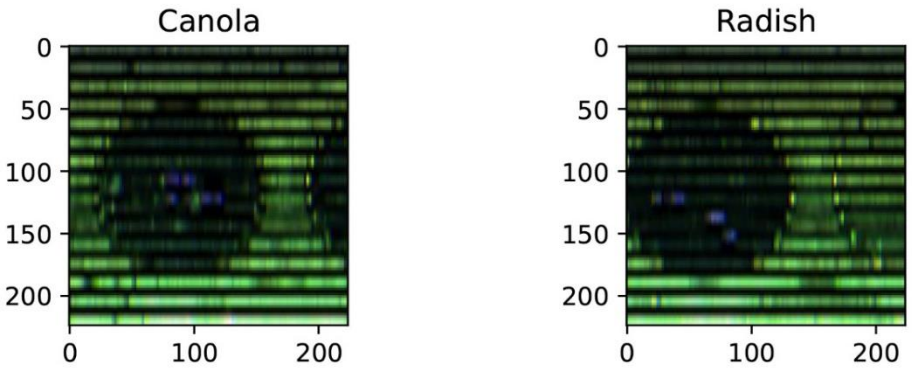


Figure 18. A sample data used as combination of three-layer laser spectral reflectance.

In the last controlling factor, the two simple CNNs shown in Fig. 4 were compared. One CNN was with two simple blocks and a fully connected layer for classification, while the other CNN with an extra middle fully connected layer, to be able to combine all the extracted feature for classification following ReLu function. The

results show that there was no significant difference between these factors, with the CNN without the middle fully connected layer providing better RPD accuracy. This is due to the low number of features in our selected dataset, meaning that when a dataset is simpler according to its number of features, there is no need to make the network more complex and increase the computational expenses.

Table 9. Analysis of variance for comparing factors according to the first DOE, the response was the relative percentage difference in accuracy

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Learning_rate	1	21387.0	21387.0	240.3860	< 2.2e-16 ***
Augmentation	3	1201.2	400.4	4.5005	0.006079 **
Number of filters in the first block	1	524.0	524.0	5.8902	0.017841 *
Number of filters in the second block	1	33.1	33.1	0.3722	0.543827
Kernel size in block 1	1	8.0	8.0	0.0895	0.765775
Kernel size in block 2	1	100.4	100.4	1.1282	0.291857
Type of network	1	2.3	2.3	0.0264	0.871462
Interaction between filters	1	556.4	556.4	6.2541	0.014769*
Residuals	69	6138.9	61.38.9		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

3.3.2.2 Design 2

Figure 19 shows effect plots of the means and signal-to-noise ratios for the factors of the second design. As can be seen from Figure 19, the 0.0001 value was still the best for initialization of learning rate. In this design, three types of augmentation were investigated, namely vertical, vertical and normalized, and horizontal and vertical augmentation. It is obvious from Figure 19 that using horizontal and vertical augmentation still provided significantly better RPD accuracy in terms of both mean plot and signal-to-noise ratio.

Note that while only the number of filters used in the first layer significantly changed the RPD accuracy, the results show that the interaction of the number of filters used in the first layer and the

number of filters used in the second were significantly important in the second design based on the use of inception primitive blocks. This shows that in the second DOE, the combination of features made the second layer more effective in achieving higher accuracy. The optimum number of filters in the first and second layers according to effect plot were 16 and 64, respectively.

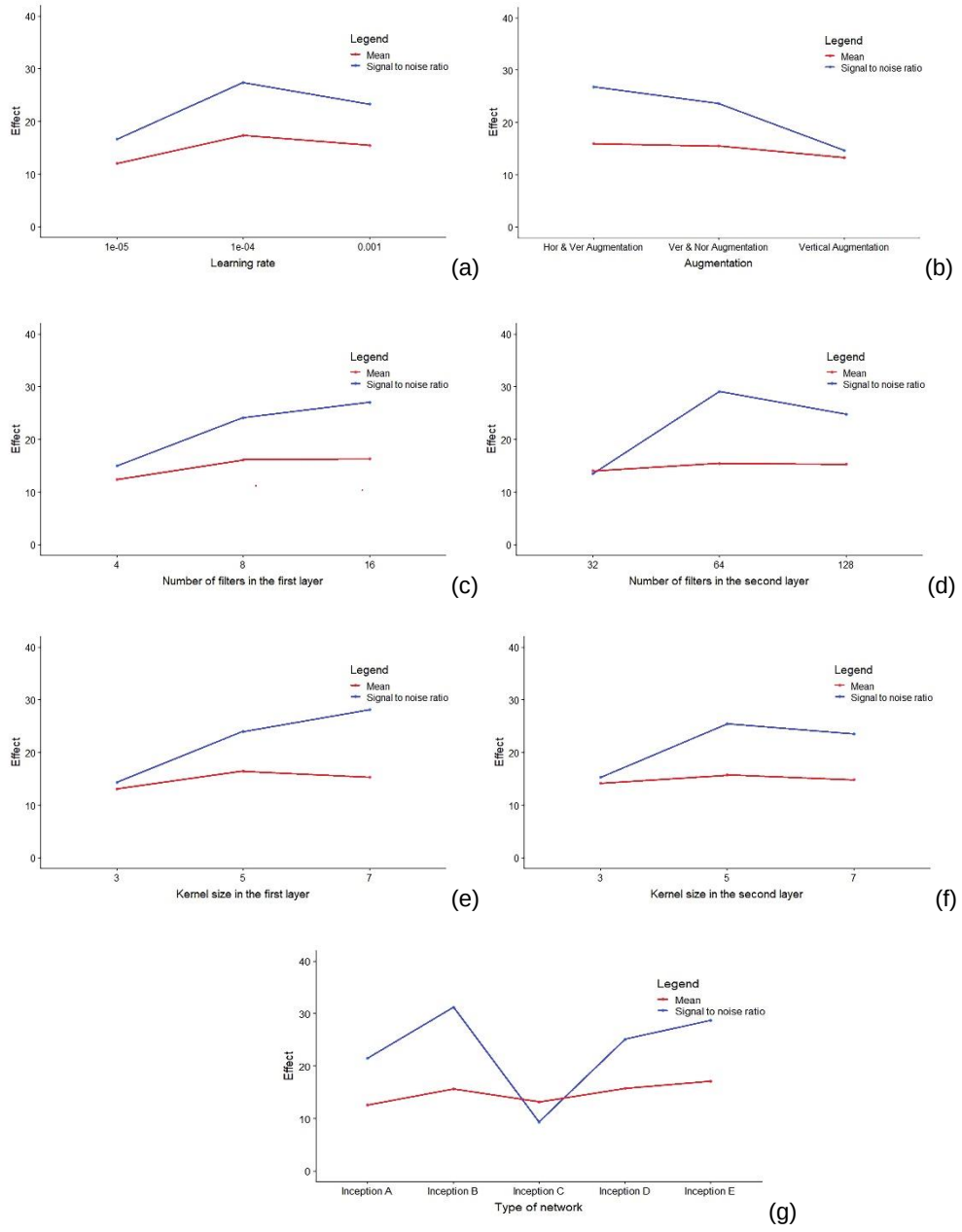


Figure 19. The effect plot of mean and signal to noise ratio for the second design. The effect plot of seven variable investigated in here. A)Learning rate, B)Type of augmentation C)Number of filters used in the first layer D)Number of filters use in the second layer E)Convolution kernel size in the first layer F)Convolution kernel size in the second layer G)Type of inception block used in the inception layer.

According to the mean of effect plots, the best kernel size was the 5x5. The kernel size in the first layer significantly changed the value of RPD accuracy, while in the second layer, the change was not statistically significant. This is attributed to the asymmetry property of the inception blocks used after the second layer, which automatically combines the various kernels and does not need to particularly fix their size in the second layer.

The best RPD accuracy was obtained by the network that used the Inception B block (shown in **Error! Reference source not found.**), which had the simplest Inception block structure with minimum combination of 3x3 kernels. High accuracy was attained in this case because replacing 5 × 5 convolution by asymmetric convolutions with lower kernels and having 3 × 1 convolution after 3 × 3 convolution significantly reduced the computational costs. Similarly, the worst results occurred in CNN with Inception C block, which went deeper in the network (up to five layers after the base layer) and used more complex combinations of kernels with 1x7 sizes and 7x1 sizes. This result could be because applying more complex kernel size such as 1x7 and 7x1 to extract more complex feature was not apparently suitable for the dataset used in the proposed research.

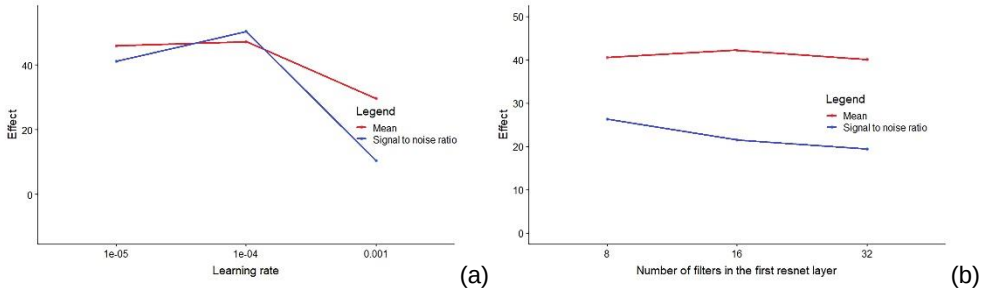
Table 10. Analysis of variance for comparing factors according to the second DOE, the response was the relative percentage difference in accuracy.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Learning **rate**	1	24.53	24.53	1.5470	0.217459
Augmentation	2	123.43	61.71	3.8914	0.024664*
Number of filters in the first block	1	175.63	175.63	11.0742	0.001358**
Number of filters in the second block	1	14.73	14.73	0.9290	0.338209
Kernel size in block 1	1	71.95	71.95	4.5366	0.036458*
Kernel size in block 2	1	6.71	6.71	0.4231	0.517368
Type of network	5	221.46	44.29	2.7928	0.022859*
Interaction between filters	1	120.34	120.34	7.5881	0.007369**
Residuals	75	1189.45	15.86		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3.3.2.3 Design 3

Figure 20 shows effect plots of the means and signal-to-noise ratios for the factors of the third design. Consistent with previous results, the best learning rate in the third designed network was 0.0001. The augmentation types selected were horizontal, vertical, and horizontal and vertical augmentation. The combination of horizontal and vertical augmentation provided significantly better RPD accuracy since it provided more scenarios for the network learning process.



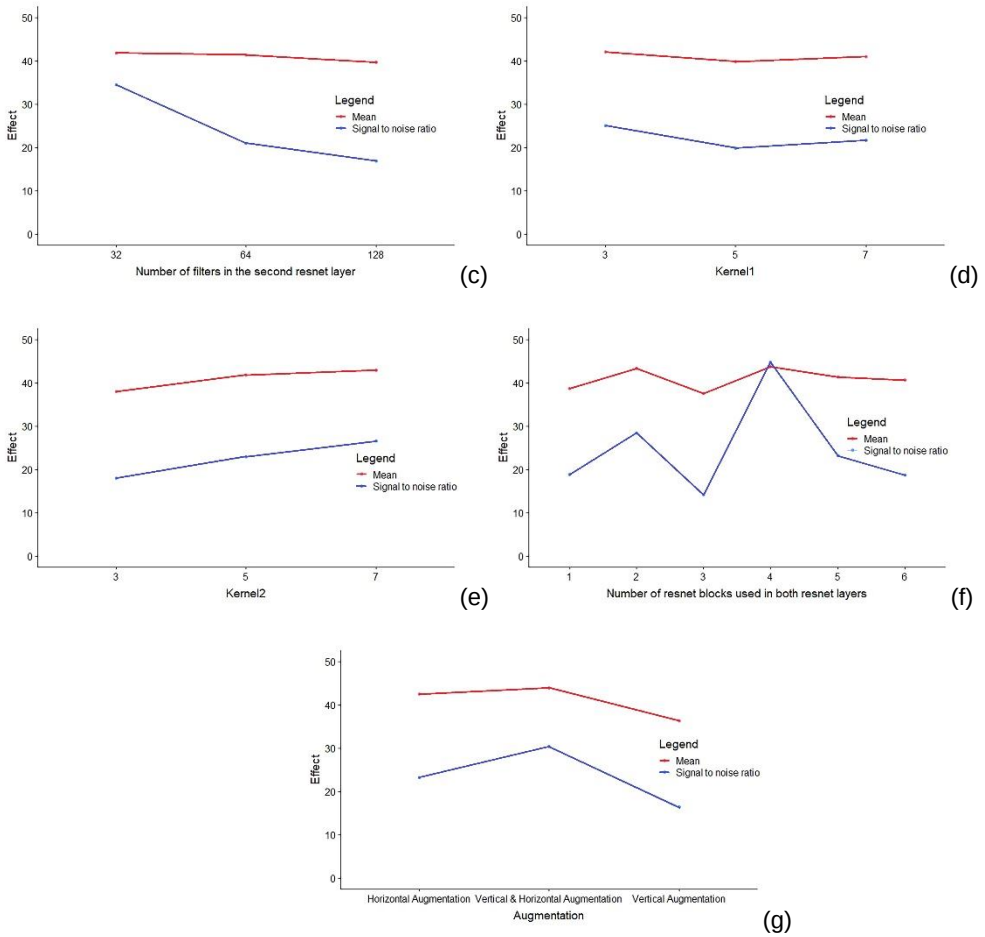


Figure 20. The effect plot of mean and signal to noise ratio for the third design. The variable investigated were A)Learning rate, B)Number of blocks used in the resnet layers, C)Type of augmentation D)Number of filters used in the first resnet layer, E)Number of filters used in the second resnet layer, F)Convolution kernel size in the first resnet layer G)Convolution kernel size in the second resnet layer.

In each residual layer, the effects of using various numbers of residual blocks were investigated to determine the importance of going deeper for classification. The results showed that the best mean RPD accuracy was obtained by using four blocks, while the difference of using various blocks were not statistically significant. The other insignificant parameters here were the number of filters and kernel sizes. As can be seen from Figure 20, the difference between the means of RPD accuracy was insignificant. This is because when the residual layer can learn from the previous layer and possesses a low number of features, it is not required to have more filters. Instead, one can go deeper and use the residual layers, which make the effect of number of filters and kernel sizes less important for a dataset of low features. Therefore, in this case, the

minimum level of filters can be selected to reduce the computational cost.

Table 11. Analysis of variance for comparing factors according to the third DOE, the relative percentage difference in accuracy was considered as response

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Learning_rate	1	5655.7	5655.7	50.4665	4.245e-10***
Augmentation	2	974.0	487.0	4.3456	0.01612 *
Number of filters in the first block	1	12.0	12.0	0.1067	0.74481
Number of filters in the second block	1	72.9	72.9	0.6504	0.42232
Kernel size in block 1	1	22.2	22.2	0.1978	0.65772
Kernel size in block 2	1	361.8	361.8	3.2281	0.07611 .
Resnet blocks	1	20.0	20.0	0.1783	0.67392
Residuals	81	9077.5	112.1		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Design 4

Figure 21 shows effect plots of the means and signal-to-noise ratios for the factors of the comparison design. In the last DOE, three designed CNNs were compared. The best learning rate obtained was 0.0001, which was in accordance with the previous results. The variations between using various augmentations were slight, while the vertical and horizontal augmentation resulted in an acceptable signal-to-noise ratio and a mean effect plot. According to the effect plots, a suitable number of filter size can be chosen, 16 for the first layer and 64 for the second layer. While the impact of each filter was not statistically significant, the interaction between them showed that in order to have proper feature extraction, the interaction between filters is important. Based on the effect plots, the optimum kernel sizes was 5x5 kernel, which was the result of the type of the used data. In comparing the three CNNs, the simple designed CNN showed significantly better effect plot of RPD accuracy and lower computational cost since the number of parameters in this CNN is lower than those in the other CNNs.

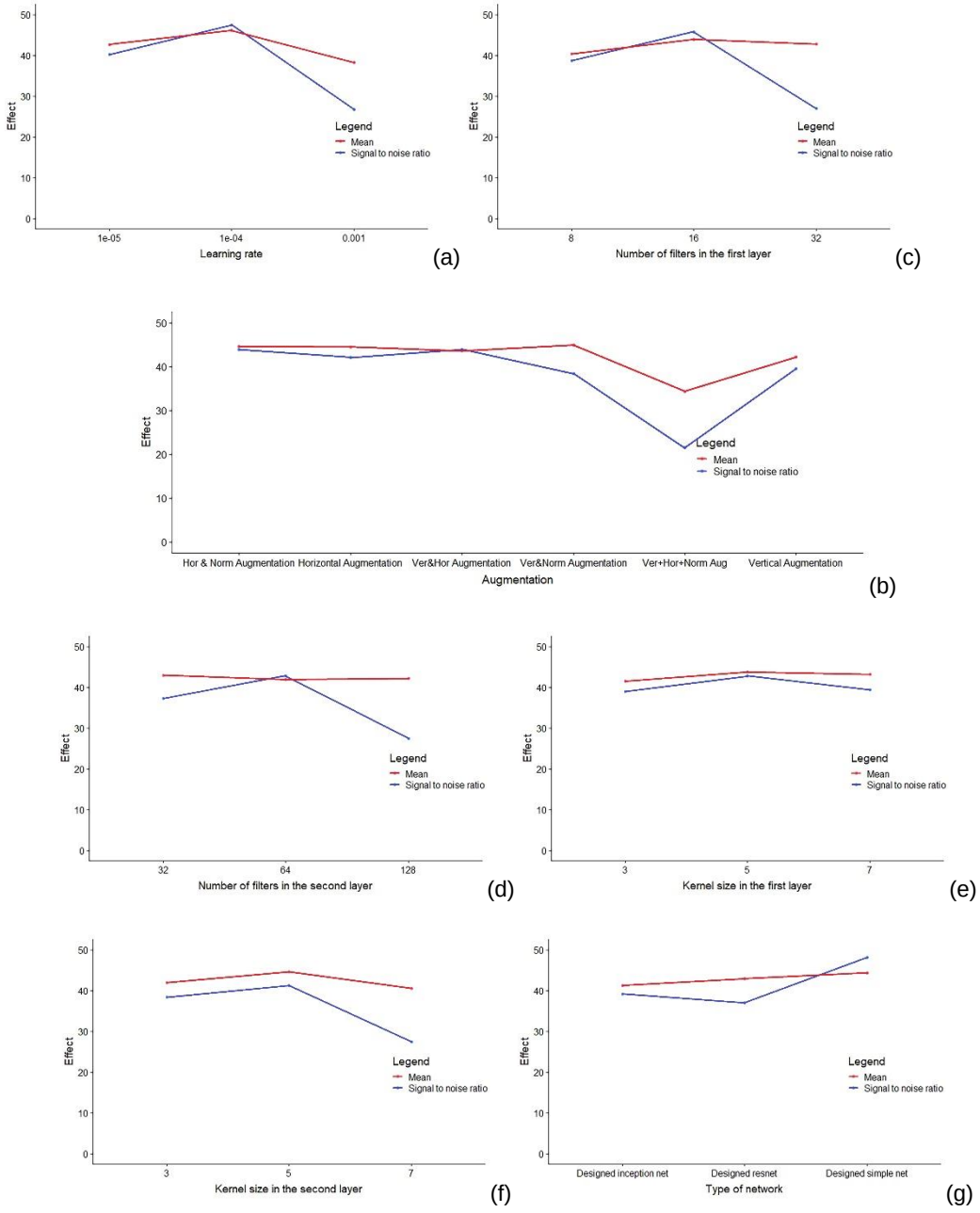


Figure 21. The effect plots of mean and signal to noise ratio of the fourth design for comparison of designed simple CNN, designed inception net, and designed Resnet. Each plot belong to one of the controlling variable namely: A)Learning rate, B)Type of augmentation C)Number of filters in the first primitive block D)Number of filters in the second primitive block E)Convolution kernel size in the first block F)Convolution kernel size in the second block G)Type of network (designed Inception, designed Resnet, designed simple net).

Table 12. Analysis of variance for comparing factors according to the fourth DOE, the relative percentage difference in accuracy was considered as response

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Learning_rate	1	689.74	689.74	22.4571	9.973e-06***
Augmentation	5	1234.51	246.90	8.0388	4.122e-06***
Number of filters in the first block	1	57.39	57.39	1.8685	0.175736
Number of filters in the second block	1	3.45	3.45	0.1123	0.738422
Kernel size in block 1	1	43.93	43.93	1.4304	0.2345461
Kernel size in block 2	1	30.04	30.04	0.9780	0.325866
Type of network	2	439.45	219.72	7.1539	0.001434**
Interaction between filters	1	250.80	250.80	8.1659	0.005522**
Residuals	75	2303.52	30.71		

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

3.3.3 Comparison of CNNs with and without optimizations

The aim of the last investigation was to introduce an approach that determines the simplest network that could be classified as efficient as the manually-engineered network. The previously-discussed agricultural dataset was also chosen to evaluate the optimisation procedure. The designed CNNs were compared based on their testing accuracies before and after parameter optimization. In order to have better comparison, the network was activated with five various seeds, namely cuda and pytorch random seeds. Fig. 11 shows the average testing accuracies of the investigated CNNs activated with five different random seeds. The comparison results show that, for all the cases, there were improvements in the testing accuracy. Generally, there was a correlation between the number of significant controlling factors in the optimization process and the amount of improvement in accuracy. The Inception network had five significant controlling factors, resulting in the highest improvement in accuracy, while the Resnet network had only three significant factors, displaying the lowest improvement in accuracy.

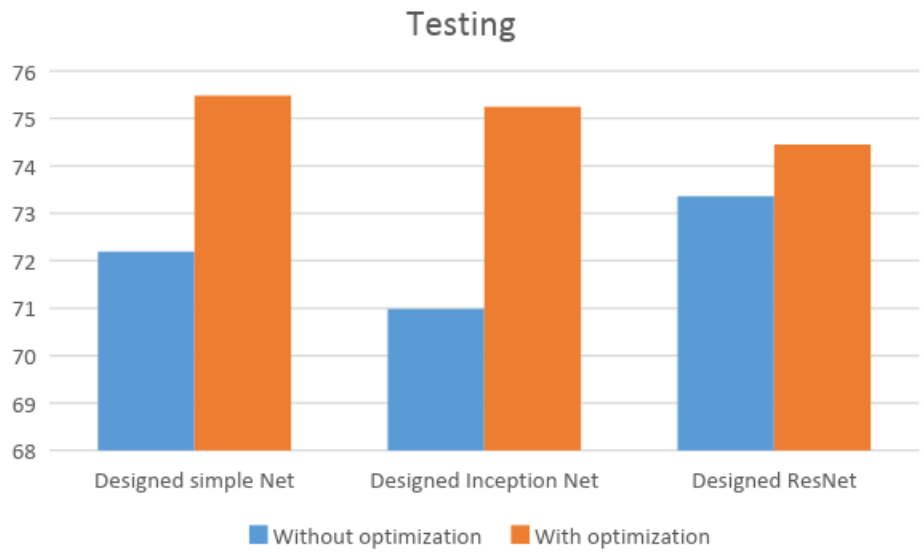


Figure 22. Average testing accuracies of the CNNs activated with five different random seed.

Chapter 4: High-detection-speed weed sensors employing advanced machine learning algorithms

This chapter has recently been re-submitted (after revision) and is still under review of GigaScience journal. This article appears as it does in print, with the exception of minor changes to the layout, number formats, font size and font style, and introduction and conclusion removal which was implemented to maintain consistency in the formatting of this thesis.

4.1 Abstract

Precision agriculture requires automated systems for weed detection as weeds compete with the crop for water, nutrients, and light. The two practical factors in weed detection and spraying are speed and accuracy. Aimed at achieving weed-crop classification in real-time at high vehicular speeds, while maintaining high accuracy, a specifically-constructed plant discrimination unit is utilised, employing near-red and infra-red spectral data in conjunction with camera images. After data preparation, three specifically designed algorithms are introduced based on state-of-the art algorithms and then optimized for a particular data set. Then, the algorithms are uploaded on a small embedded NVIDIA Jetson TX1 board for real-time precision agricultural applications, and a larger high throughput GeForce GTX 1080Ti board for aerial crop analysis applications. Experimental results show that the weed detection sensing speed can be improved by up to thirty times up to 60 km/h, and up to 2300 km/h for precision agriculture and aerial-based crop analysis, respectively, while maintaining an adequate accuracy.

4.2 Methodology

4.2.1 Data collection setup

Two types of data were collected, namely, spatial data and spectral data. Both data sets were collected simultaneously on a movable platform comprising of a Xilinx Zynq ZC702 for processing, using different instruments for data capturing. Spatial data was captured using an On-Semi VITA 2000 camera, which is a 2/3 inch Widescreen Ultra Extended Graphics Array (WUXGA) CMOS image sensor configurable in HD format (1920 x 1080) or 4:3 format (1600 X 1200). The camera had a focal length of 9mm and was located at 980mm from the ground.

A Plant Discrimination Unit (PDU) developed in ECU's Electron Science Research Institute, Australia, was specifically designed to collect spectral reflectance data at 635nm, 685nm and 785nm [1, 74, 78]. Figure 23 shows the schematic of the PDU.

The PDU has two three-laser modules. Each laser module sequentially generates three 1mm collimated pulsed laser beams, two red (635 nm and 685 nm) beams and one near-infrared (785 nm) beam. The pulsed laser beams are spatially aligned using two beam combiners, resulting in three sequentially pulsed, spatially overlapping beams of similar polarization. These beams are then propagated into an optical cavity that generates 15 collimated beams to illuminate the vegetation. The optical cavity comprises a reflective coating on one side and gradient transmissive coating on the other side. A linear array detector with 8-bit resolution and line rate of 10kHz scans the intensity of the reflected beams from the vegetation samples. The PDU unit has a system controller that controls temperature, the laser module drivers, detectors and the nozzle activator for weed spraying [1, 4, 21].

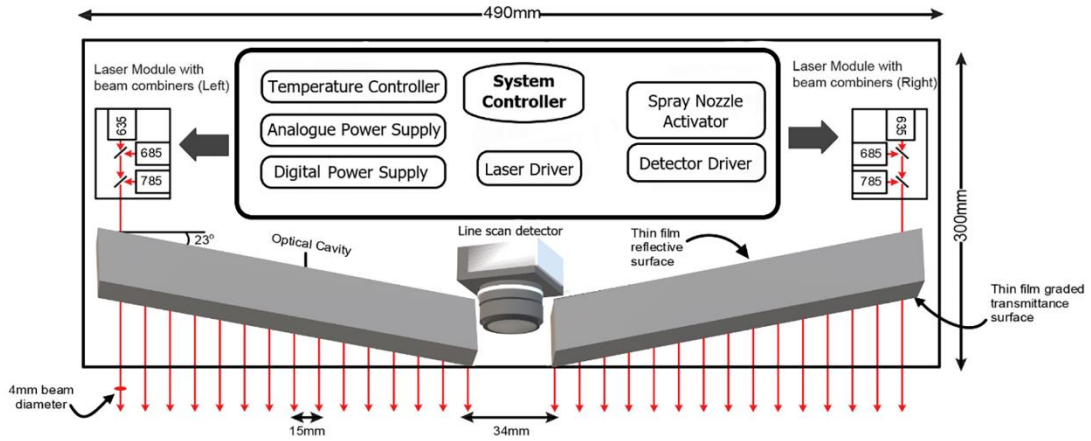


Figure 23. Schematic of plant discrimination unit (PDU) detecting spectral reflectance of laser beams over the span of 490mm.

Radish (Raphanus raphanistrum subsp) and canola (Brassica napus) plants were chosen as the weed and crop samples, since they have resembling broad leaf shapes. Canola was considered as the crop and radish as a representative of a common problematic wild radish weed in Western Australia. Two hundred plants of each type were grown, and spatial and spectral data sets were collected in five different stages, three days after early germination for five consecutive weeks.

Data collection started on 6 March 2019, three days after germination of the plants. Two facilities, shown in Figure 24, enables the data collection, namely, (i) a spectral reflectance sensor unit installed on a quad bike for in-the-field data collection, and (ii) a specifically designed testing facilities (testbed), built at Electron Science Research Institute (ESRI), Edith Cowan University [1, 78] 2- a system for real field data collection shown in Figure 24.

The 6600cmx1100cm testbed enables data collection at speed of up to 20 km/h with submillimeter accuracy along the longitudinal direction. The illumination in the testbed is supplied through fluorescent lights with high frequency ballast (>20kHz) and background light is suppressed using thick isolating black curtains. For the in-the-field sensor, a camera and a spectral reflectance sensor are installed onto an enclosure to ensure excellent lighting

condition with minimum background light. Data was collected with both systems. However, since the aim of this study is to evaluate the maximum operating speed, the accurate calculation of the covered distance is crucial. Therefore, we conducted experiments using an indoor test-rig, which enabled accurate speed calculation since the distance along the longitudinal direction can be measured with submillimeter accuracy. To ensure the generalization of results, training, validation and testing plants were separated.

a)

b)

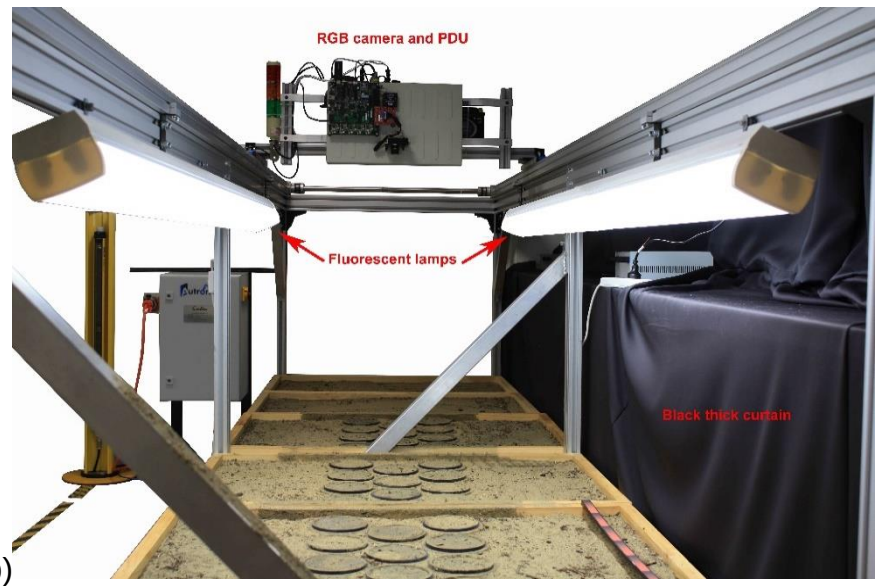


Figure 24. Data collection facilities for (a) Real-field condition, (b) laboratory condition.

4.2.2 Data pre-processing

The camera's RGB image resolution was set to 456×456 pixels. Image colour calibration has been implemented to have consistency in different light conditions including daylight and fluorescent light for both of the systems introduced in Figure 24. A X-Rite ColorChecker® 24 Patch Classic target was used to measure and adjust image colour production by VITA 2000 camera. Several runs in different conditions such as various speed has been done to provide optimum colour correction. However, because of robustness of CNN algorithms in variation of light, the classification can even work with poor or without colour calibration [87, 88].

The PDU data consisted of 30 spectral reflectance data points on a line for each laser wavelength, as well as a background signal captured when the lasers were off, which represented the background illumination noise. The background signal was subtracted from the laser reflectance data giving a three-set input of background-noise-free spectral reflectance data. Six hundred lines of three-set inputs were stacked together to provide an input data source of 30×600×3. The size of the PDU data was then reshaped to 224×224×3 using interpolation. Figure 25 shows typical sample data collected by the RGB camera and PDU. Based on insights from previous results [35], the raw spectral reflectance data was used directly, instead of computing and capturing the normalized difference vegetation index (NDVI) values, as this data leads in more accurate results.

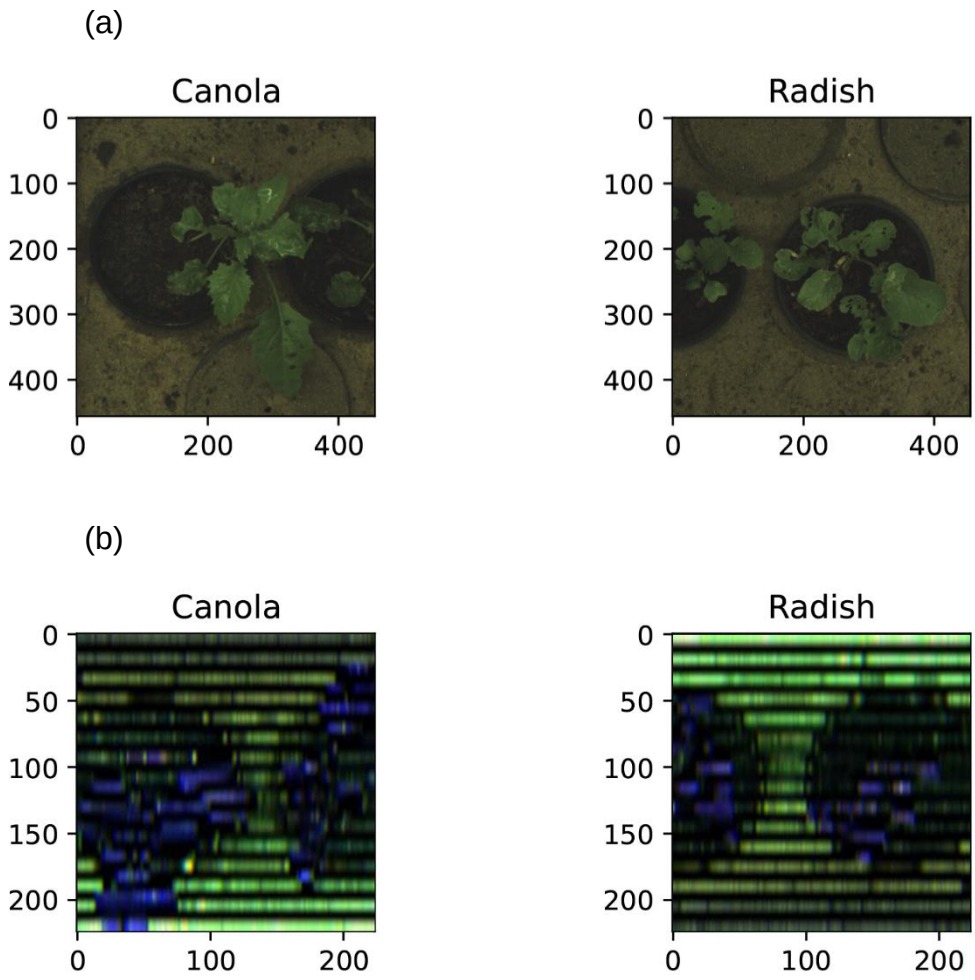


Figure 25. Sample data collected by: (a) Spatial camera, (b) Plant discrimination unit.

4.2.3 Neural Network Models

Unlike conventional neural networks, convolutional neural networks (CNNs) are type of machine learning algorithms, which comprise deeper neural networks with hierarchical data representation based on the use of convolution [64]. CNNs can efficiently learn complex patterns and features, if adequate number of data provided [65].

Two types of deep learning models were applied to both spatial and spectral data for weed-crop classification: (i) state-of-the-art pretrained CNNs, and (ii) simplified networks designed using a Taguchi optimization methods [89]. The state-of-the-art models:

VGG, Inception and Resnet networks, were pretrained on the large 10 million+ images of the ImageNet database [40, 41, 43, 90].

The VGG architecture is the winner of the ImageNet ILSVRC-2014 competition, and has been trained on data with input size of 224×224 [40]. It has 16-19 layers with 3×3 kernel-sized filters. Having smaller kernel size, in comparison to the previous ImageNet winner (AlexNet), assists in learning more complex features at lower computational cost and also enables retention of finer features of input data. CNN blocks with the same kernel size and repeated in multiple layers were used to extract more complex features.

The VGG architecture typically requires massive computational power, since the convolutional operations are massive in dense connection (from every output to every input in next layer). Inception networks were designed based on the concept that most activations in dense connections are either needless (zero value) or avoidable due to correlations between them. Therefore, more efficient networks must have sparser connection between activations. Inception layers typically have increasing kernel sizes but sparser connections between them, and this helps reduce the computational through a wider rather than deeper network [43].

The third investigated CNN model , ResNet, was built on the idea of “identity or skip connections” in order to reduce the vanishing gradient problem [41]. Double or triple identity connections are typical in ResNet implementation. Skipping connections successfully simplifies the network and speeds the training, because fewer layers are used at the beginning of training, thus requiring fewer layers to propagate through. Note that identity connections reduce the effect of the vanishing gradient. During training, the network gradually learns the feature space and restores skipped connections, and these deep transferring features enable the CNN to be trained faster. In this study, the state-of-the-art NN architectures of similar parameters

were used but fine-tuning was performed to the last three layers to maximize the weed discrimination accuracy for the current data set.

In addition to pretrained state-of-the-art CNN models, three simplified CNNs were designed by a modified Taguchi optimization process, using the reduced dimensionality PDU data, in order to reduce the computational complexity. Three networks were designed and optimized: a simple network, a network utilizing Inception blocks, and a network utilizing ResNet blocks. All network parameters were optimised based on statistical designs of the experiments in order to reduce the computational cost without compromising the detection accuracy [89].

The Simple Net included three simple primitive blocks following a fully connected layer with softmax activation function. Each primitive block comprised of a convolutional layer with batch normalization, ten percent dropout, Rectified Linear unit (ReLu) function, and max pooling with 2×2 kernel size. The max pooling had the stride of two by two without padding. The numbers of filters were 16, 128, 256 in the first, second and third layer, respectively. The convolution kernel size was 5×5 for spectral data as more informative features were provided every other line by the spectral data (shown in Figure 25). The convolutional kernel size was 3×3 for image data. For all the convolutional layers the strides were equal to one by one, then the size of zero padding were 2×2 for spectral data and 1×1 for image data. This design is illustrated in the first part of Figure 26.

For the proposed Inception network, the design was almost the same in the first two layers and fully connected layer, however, the dropout was not applied in the first layer and its probability of retention were 20 percent (p=0.2) in the second layer. For the Inception network, various structures were investigated and the optimum design that was obtained is shown in the second part of Figure 26.

The first simple primitive block in the Resnet was similar to the previous design but with 7×7 kernel size in maxpooling with stride of 1×1 and with 1×1 zero padding. The second simple block included only the convolutional layer with 64 filters and batch normalization. The network built-in with two Resnet blocks, including four residual layers in each, is shown in the third part of Figure 26.

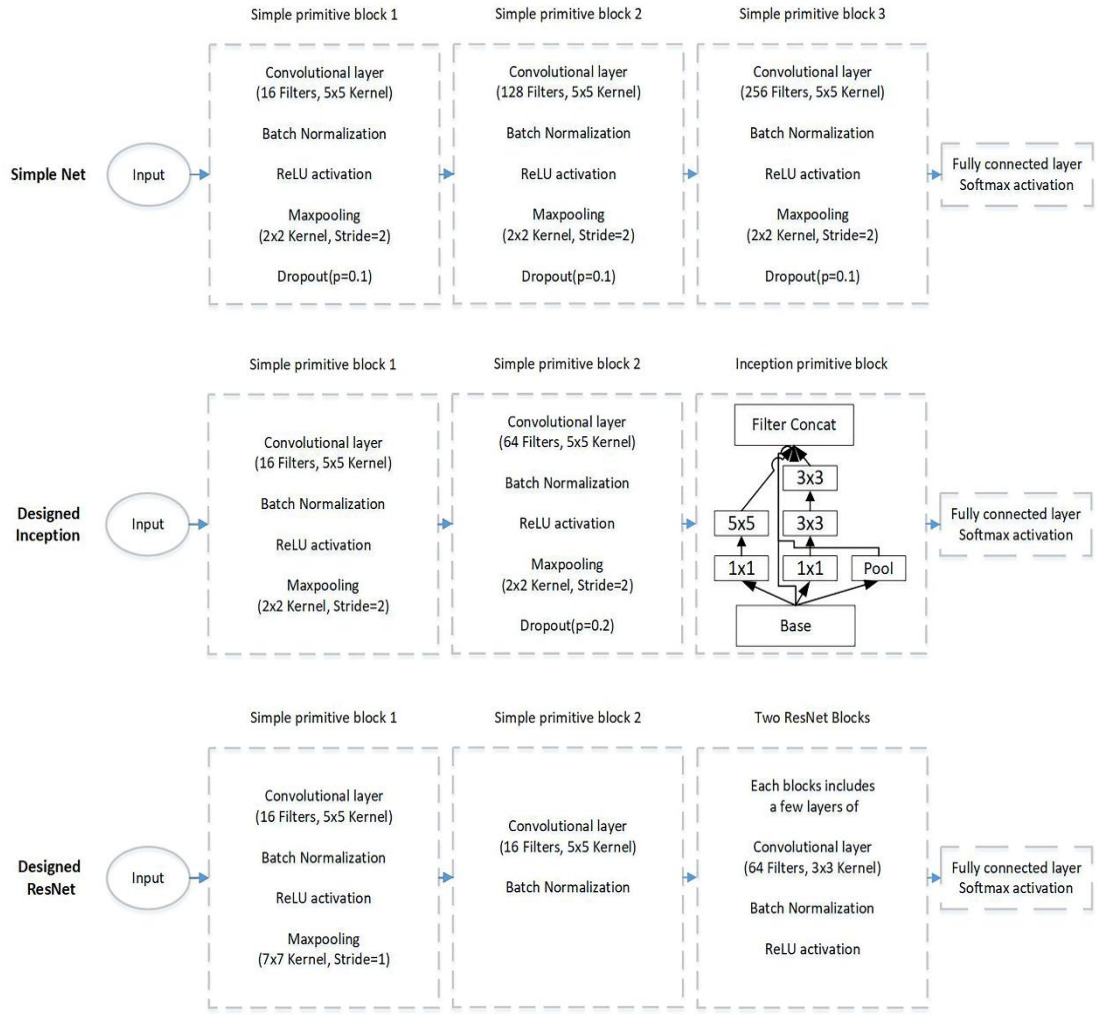


Figure 26. Structures of three optimized designed for weed-crop classification: (a) according to simple primitive blocks, (b) Based on the Inception network concept, (c) based on Resnet concept.

4.2.4 Experimental Hardware Setup

All the developed algorithms were implemented and tested on two different processors, (1) NVIDIA Jetson TX1 (2) and (2) GeForce GTX 1080Ti (NVIDIA GPU). Jetson TX1 is a compact small board that is typically used for real-time trials. It is built with a 4GB memory,

256 CUDA cores, a 64-bit Quad-Core ARM® CPU delivering performance of over 1 TeraFLOPs. On the other hand, GeForce GTX 1080Ti has a 11 GB memory, a 352-bit interface used in conjunction with an Intel® Core™ i7-7800X X-series Processor delivering over 10 TeraFLOPs.

4.3 Results

The data was collected from the camera and PDU unit, as described in the methodology. The camera collected 10,000 images per class (radish and canola). The image data sets were collected from plants at 5 different stages of development, with 2000 images per stage of each plant. The PDU data was collected at a linear scanning resolution of 30 pixels per line. For each run of 20 plants, 200 spectral images at resolution of 30×600 were generated. From a total of 10 runs per plant stage, 2000 PDU images were collected. Therefore, a total of 20,000 camera and 20,000 PDU images were generated during the five stages for machine learning. As per the data processing algorithm, camera and PDU images were converted to 224×224×3 size data.

The six deep-learning models, including the three pretrained models, VGG, Inception and ResNet, as well as the three simplified models, corresponding to a generic simple CNN network, the network based on Inception blocks and the network based on ResNet blocks were trained on the collected data. The data training set consisted of 60% of the data, and the remaining data was divided equally between the validation and testing sets. The pytorch deep learning library was used for training, with 10 epochs for each model, with Adam optimizer with an initial learning rate of 10^{-3}.

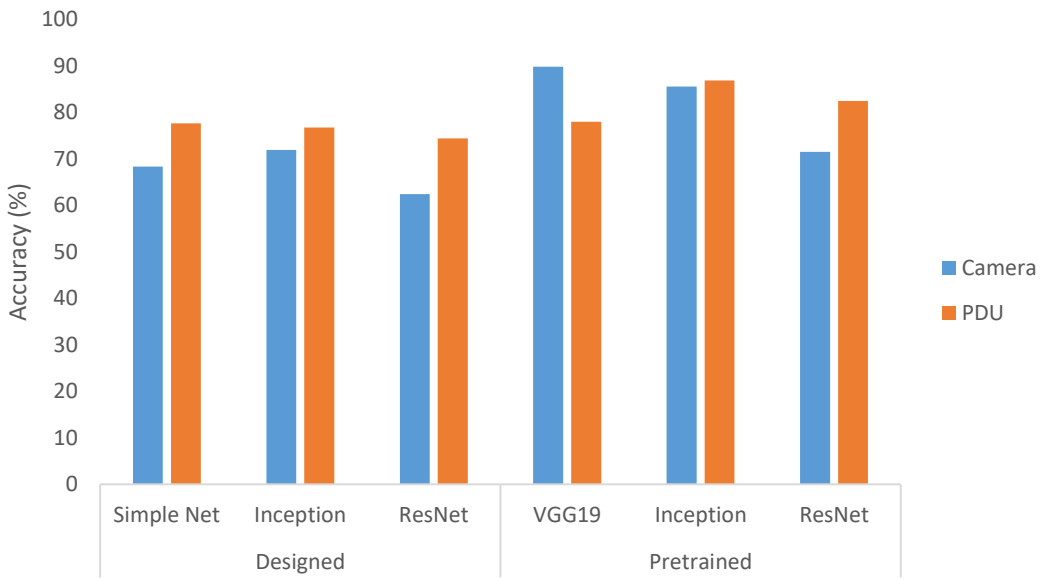


Figure 27. Testing accuracy of trained models using camera (blue) and PDU (orange) datasets.

Figure 27 compares the discrimination accuracy using both the spatial data acquired by the camera and the spectral data collected by the PDU. The highest accuracy attained was with the VGG network using the camera data. However, for all the other networks, the PDU dataset provided higher accuracies.

The trained models were installed into the two hardware platforms: the stand-alone JetsonTX1 development board and the GTX 1080Ti Nvidia GPU board connected to a personal computer.

Table 13 and Table 14 show the number of parameters, testing time and the corresponding physical length travelled for all investigated CNNs, for the JetsonTX1 development board and GTX 1080Ti Nvidia GPU, respectively. For the Jetson TX1 board, 80 sample images were classified to test each image prediction time (Table 13), and for the GTX 1080Ti board, 12,000 test images were used, due to the increased speed per single prediction (Table 14).

Table 13. Number of parameters, testing times and corresponding length covered for each model for Jetson TX1 platform.

Data	Method	Network	Number of Parameters	Testing time(s) on JetsonTX1 (80 data samples)	Covered Length (m)	Covered Area (mm^2)
Camera	Designed	Simple Net	2536290	41.3893	18	225*225*80
		Inception	3803810	9.8715		
		ResNet	25556306	24.4147		
	Pretrained	VGG19	21630066	50.2243		
		Inception	25128650	28.1008		
		ResNet	42831938	30.8135		
PDU	Designed	Simple Net	988002	3.3878	48	600* 225*80
		Inception	633378	2.8402		
		ResNet	1677650	7.364		
	Pretrained	VGG19	20425842	14.7517		
		Inception	27161264	9.9403		
		ResNet	42504258	10.292		

Table 14. Testing times and corresponding length covered for each model for GTX 1080Ti platform.

Data	Method	Network	Testing time(s) on NVIDIA GeForce GTX 1080Ti (12000 data samples)	Covered Length (m)	Covered Area (mm^2)
Camera	Designed	Simple Net	18.8273	2700	225*225*8000
		Inception	46.8124		
		ResNet	356.8608		
	Pretrained	VGG19	61.5718		
		Inception	64.8565		
		ResNet	59.0395		
PDU	Designed	Simple Net	4.5658	7200	600* 225*8000
		Inception	4.121		
		ResNet	13.759		
	Pretrained	VGG19	18.8967		
		Inception	46.239		
		ResNet	35.0663		

To calculate the average real-time detection speed, two factors needed to be determined. One is the testing time and the other is the distance covered during this time. The area covered by each camera image was 225×225 mm^2 and the area covered by each PDU image was 600×225 mm^2. Testing time for the JetsonTX1

board was calculated using 80 data samples corresponding to a length of 0.225×80 (18m) covered by the camera and a length of 0.6×80 (48m) covered by the PDU.

Figure 28 and Figure 29 show the results of the detection speed for the JetsonTX1 and GTX 1080Ti boards, respectively. What stands out in these figures is the increase in speed that is achieved by replacing the camera data with the PDU data. For the Simple Net, an improvement in detection speed of thirty times is displayed without a significant reduction in detection accuracy. This is because the PDU is able to collect data at lower resolution allowing a large distance to be covered. Furthermore, improvement in detection speed is achieved by using the simplified models compared to the pretrained models. This improvement occurs both in the case of both the PDU and camera datasets. This occurs as the simplified models were optimized to improve the computational speed and reduce the number of parameters, simultaneously. The numbers of parameters for each algorithm are shown in Table 13. The inception Network attained the highest speed for both the simplified and pretrained networks. These results are attributed to the efficient network pruning in the Inception network [43].

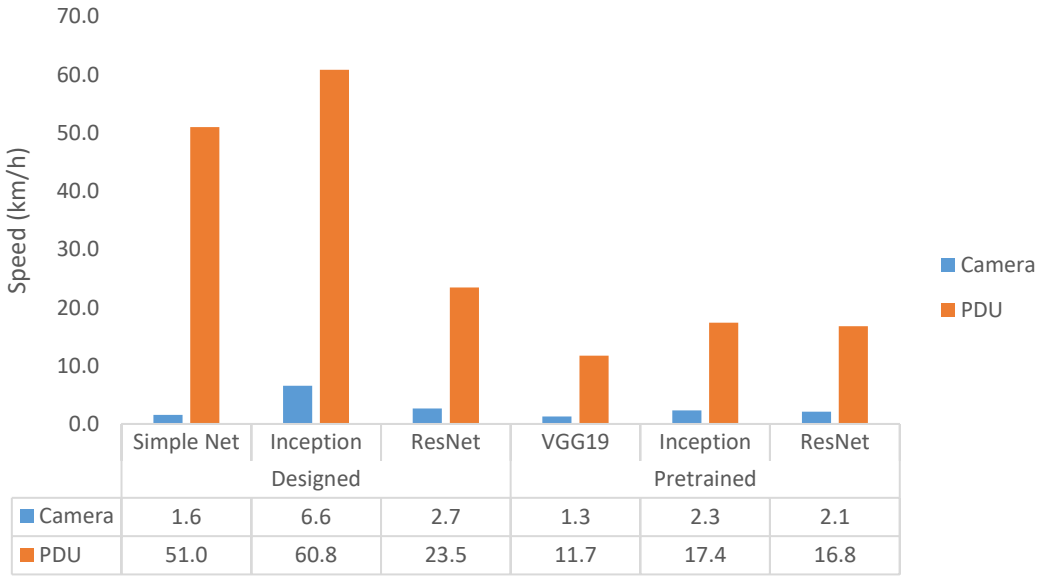


Figure 28. Speed of detection for each network deployed in the JetsonTX1 embedded platform.

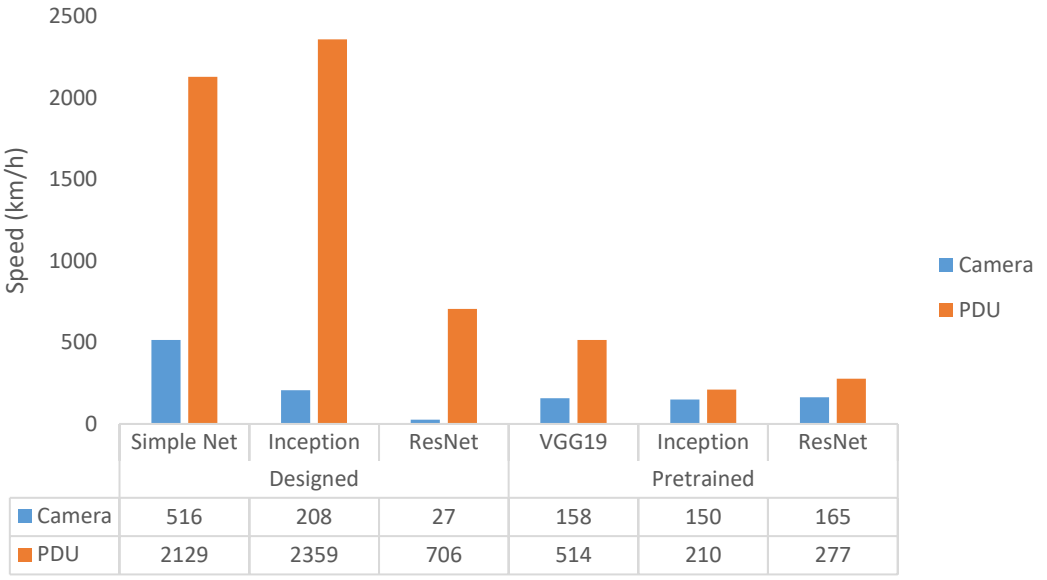


Figure 29. Speed of detection for each network deployed in the GTX1080Ti platform.

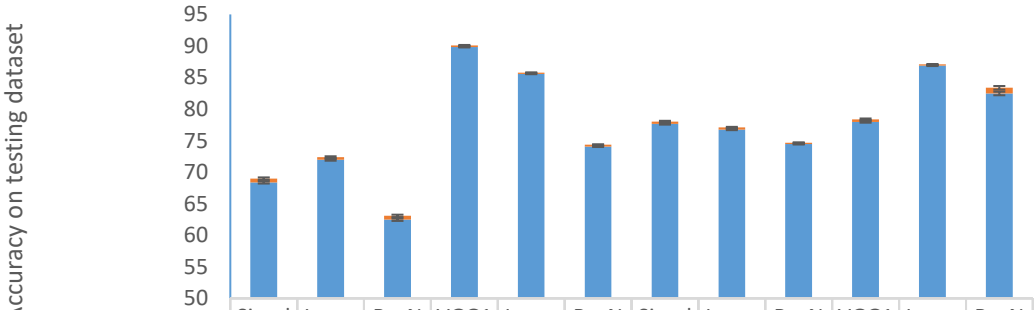
The detection speeds attained with the GeForce GTX 1080Ti platform for both pretrained and simplified networks are shown in Figure 29. There are still significant differences in detection speed between using the PDU data and the camera, and also between using the pretrained networks and the simplified networks. Generally, the GeForce GTX 1080Ti platform provides much faster classification, however, if sufficient resources are available, a weed-crop discrimination speed of 700-2400km/h can be achieved using GeForce GTX 1080Ti platform.

4.4 Discussion

To demonstrate the ability of the neural nets to generalize, the plants carefully separated in three different categories namely: training, validation and testing. Training the NNs with the training data was validated with the validation data during training. However, testing data set reserve separately to evaluate the overall performances of a NN algorithms. If model can predict the data has not been train on (testing dataset) but the models can predict them correctly, the model has the capacity of generalization [91, 92]. The calculated testing accuracies of the trained models using camera (blue) and PDU (orange) datasets, for the various NNs, are shown in

Figure 27. The generalization comes from the fact that CNNs algorithms have not seen testing data but provide appropriate accuracies.

To further validate the generalization of our approach, we made a pool of 4000 testing data and randomly selected 400 data from that pool to evaluate the accuracy CNNs models. The process of selecting randomly 400 data from a pool of test dataset and evaluating accuracy was repeated 10 times to provide mean and standard deviation of testing dataset accuracies. Figure 30 shows the average accuracies and standard deviations calculated using ten random testing datasets for the different CNNs. As can be seen in Figure 30, the standard deviations, and consequently the standard errors (shown by error bar), are much lower than the CNN accuracies, and this demonstrates the generalization of our approach.



	Camera						PDU					
	Designed			Pretrained			Designed			Pretrained		
	Simple Net	Inception	ResNet	VGG19	Inception	ResNet	Simple Net	Inception	ResNet	VGG19	Inception	ResNet
Standard Deviation	0.60	0.43	0.61	0.24	0.18	0.27	0.37	0.32	0.21	0.41	0.20	0.90
Accuracy on Test data	68.38	71.95	62.46	89.84	85.58	74.07	77.65	76.76	74.45	77.97	86.88	82.47

Figure 30. Average accuracy and standard deviation of ten random testing dataset with different CNNS.

To demonstrate the robustness of the CNN algorithms against fluctuations in illumination variation, the CNN algorithms were tested with and without lighting augmentation using 4,000 testing images for each case. Lighting augmentation was carried out using a PyTorch with randomly varied brightness, contrast and saturation and hue.

The accuracies of the algorithms for both scenarios are shown in Table 15. The results shown in Table 15 indicate that there was no significant difference between the accuracies of the algorithms with and without lighting augmentation.

Table 15. Comparison of accuracy on collected testing dataset and colour augmented dataset.

	Simple Net	Inception Designed	ResNet	VGG	Inception Pretrained	ResNet
Accuracy of collected data	68.38 ± 0.61	71.95 ± 0.86	62.64 ± 1.02	89.84 ± 0.17	85.85 ± 0.22	74.7 ± 0.59
Accuracy of augmented data	68.26 ± 0.58	69.91 ± 0.69	62.1 ± 1.80	91.32 ± 0.62	83.25 ± 0.43	73.06 ± 0.64

The average speed is defined as the ratio (X/t) of the travelled distance (X) and the testing time (t) measured by each board. Therefore, the following formula can be used to estimate the error of the calculated speed:

$$(\frac{\Delta V}{V})^2 = (\frac{\Delta X}{X})^2 + (\frac{\Delta t}{t})^2 \quad (1)$$

Where X is travelled distance by instruments and t is testing time of each instrument and ΔX and Δt are uncertainty of covered distance and testing time and can be calculated by root-mean-squared error of runs of the experiment. Knowing the traveling speed (V) the uncertainty of speed(ΔV) can be calculated using Eq. (1). by calculating the uncertainty of speed and considering the errors have normal distribution, then approximately with 95% confident interval, the speed measurement lie between V-2ΔV and V+2ΔV. Table 16 shows the calculated errors for both the Jetson TX1 and the GTX1080Ti platforms. Note that for the results shown in Table 16, the distance and testing time were extracted from Table 14 and Table 15 and the speeds were extracted from Figure 28 and 29. As can be seen from Table 16, the range of errors are mostly two to three orders of magnitude less than the calculated speed, and this further demonstrates the generalization of our approach.

Table 16. Estimation of uncertainty of speed with 95% confidence interval for different boards and collecting instruments.

	Instrument	Method	Designed			Pretrained		
		Network	Simple Net	Inception	ResNet	VGG	Inception	ResNet
Speed with 95% confidence interval	Camera	Jetson TX1	1.6 ± 0.002	6.6 ± 0.012	2.7 ± 0.003	1.3 ± 0.001	2.3 ± 0.002	2.1 ± 0.002
	PDU		51 ± 0.357	60.8 ± 0.392	23.5 ± 0.129	11.7 ± 0.063	17.4 ± 0.097	16.8 ± 0.092
	Camera	GTX1080Ti	516 ± 8.923	208 ± 1.310	27 ± 0.119	158 ± 0.798	150 ± 0.808	165 ± 1.068
	PDU		2129 ± 117.70	2359 ± 116.97	706 ± 13.944	514 ± 7.398	210 ± 2.074	277 ± 2.781

Generally, the proposed plant discrimination unit can operate at higher detection speeds due to the lower image resolution and higher distance coverage per image capture. The high accuracy attained by the PDU is attributed to the facts that (i) the 1mm diameter of the discrete illuminating laser spots generated by PDU unit is smaller than the plant leaf size (at many growth stages) and (ii) the distinctive chlorophyll absorption at the three laser wavelengths used in in the PDU unit (red, far-red and the near-infrared). In addition, the use of lower amount of PDU data led to faster convergence during NN training, and this provided better accuracy in just 10-epoch training. Statistical t-tests indicated that there was no significant difference in accuracy between the camera (M=75.35 , SD=10.41) and spectral reflectance sensor (M= 79.36, SD=4.5), t(5)= 1.134 , p-value= 0.28, d= 0.5.

These two target plants, or classes, were especially selected because they represent a practical issue in Western Australia and their discrimination is more challenging, as they have similar shapes and colours. Note that, this approach can be easily used for the classification of more than two classes, for example ImageNet ILSVRC [90] was used for the classification of 1000 classes, and with straightforward modification, it can be applied to different areas, including disease detection, soil fertility measurement and crop monitoring.

Chapter 5: Conclusion

5.1 Summary of the main research findings

The aim of the current study was to investigate various algorithms and statistical techniques in order to facilitate weed-plant discrimination. The investigation were conducted through three steps, each reported in a separate chapter (Chapters 2-4).

In Chapter 2, algorithms based on Support Vector Machine (SVM) learning with NDVI and raw data inputs were developed and their weed-crop discrimination performances were evaluated and compared with a conventional plant discrimination algorithm based on the measurement of discrete NDVIs and the use of data aggregation. Data was collected by measuring the spectral reflectance properties of corn (as a crop) and silver beet (as a weed) at 635 nm, 685 nm, and 785 nm, at a speed of 7.2 km/hr. The results of this work have shown that the discrimination performance of the linear-kernel SVM algorithm, with either raw reflected intensities or NDVI values being used as inputs, provides better discrimination accuracy than the conventional discrete NDVI-based aggregation algorithm. Experimental results, carried out in laboratory conditions, have demonstrated that the Gaussian SVM algorithms can classify corn from silver beet with a discrimination accuracy of 97%, whereas the maximum accuracy attained using the conventional NDVI-based method does not exceed 70%.

In Chapter 3, four main Designs of Experiment (DOE) have been proposed and investigated to determine and optimize the significant factors that affect the performance of three simplified Convolutional Neural Network (CNN) architectures, namely, the VGG network, the Inception network and the Resnet network.

DOE algorithms based on utilizing Taguchi statistical methods have been developed, for optimizing the CNN parameters, and tested using spectral agricultural data. Results have shown that, for all investigated CNN architectures, there was an obvious improvement

in accuracy in comparison with un-optimzed CNNs, and that the Inception network yields the highest improvement in accuracy compared to the simple and Resnet CNN counterparts. Results have also shown that, generally, based on proposed Taguchi experimental designs, the optimum hyper parameter of a CNN can be determined in order to provide simpler and faster trainable network with improvement on its performance.

In Chapter 4, the weed detection speed that can be attained by using Convolutional neural network algorithms (VGG, Inception and Resnet and their corresponding simplified algorithms) was investigated. Convolutional neural network algorithms were implemented onto both a small embedded NVIDIA Jetson TX1 board for real-time precision agricultural applications, and a larger high throughput GeForce GTX 1080Ti board for aerial crop analysis applications. Experimental results have shown that for a simplified CNN algorithm implemented on a Jetson TX1 board, an improvement in detection speed of thirty times (60 km/hr) can be achieved by using spectral reflectance data rather than imaging data. Furthermore, with an Inception algorithm implemented on a GeForce GTX 1080Ti board for aerial weed detection, an 11-fold improvement in detection speed (~2,300 km/hr) can be achieved, while maintaining an adequate detection accuracy above 80%. These high vehicular speeds have been attained by reducing the data size, choosing spectral components with high information contents at lower resolution, pre-processing efficiently, optimizing the deep learning networks through the use of simplified faster networks for feature detection and classification and optimizing computational power with available power and embedded resources, to identify the best fit hardware platforms.

An adapted Taguchi algorithm was used, which allowed network parameter optimization and provided faster processing time without compromising the accuracy.

The importance of the results is that small boards, such as the compact JetsonTX1 board, have the potential to attain real-time plant discrimination speeds (up to 60 km/h) without the need of significant hardware resources, whereas the more complex GTX1080Ti board can achieve much faster detection speeds (up to 2,300 km/h), however, it is bulky and its hardware is more complex .

5.2 Recommendations for further research work

The development of accurate automated weed detection and eradication systems using quad bikes, tractors and areal drones requires comprehensive data on a wide variety of weed and crops shapes, colour, and sizes, collected in practical scenarios, e.g., partially-shaded plants, different plant heights and overlaped plant leaves. In order to address the detection challenge, it is recommended to use of Diverse Enhanced Capsule Networks (DE-Caps-Nets), which is a type of artificial intelligence with a spatial group-wise enhance (SGE) mechanism, which emulates the human brain but is much faster.

Another recommendation for future work is the use of computer vision and artificial intelligence in conjunction with embedded products and internet of things (IoT) for disease detection, soil fertility measurement, and crop monitoring. This big data collection, management and analytics enable farmers to analyse crops remotely and make decisions.

References

[1] S. Akbarzadeh, A. Paap, S. Ahderom, B. Apopei, and K. Alameh, "Plant discrimination by Support Vector Machine classifier based on spectral reflectance," *Computers and electronics in agriculture,* vol. 148, pp. 250-258, 2018.

[2] S. Akbarzadeh, S. Ahderom, and K. Alameh, "A Statistical Approach to Provide Explainable Convolutional Neural Network Parameter Optimization," *International Journal of Computational Intelligence Systems,* vol. 12, no. 2, pp. 1635-1648, 2019.

[3] S. Akbarzadeh, S. Ahderom, C. Miller, J. Rowe, and K. Alameh, "High-detection-speed weed sensors employing advanced machine learning algorithms," *GigaScience,* vol. Under revision, 2020.

[4] S. Akbarzadeh, S. Ahderom, and K. Alameh, "Application of spectral reflectance for increasing plant discrimination speed in precision agriculture," in *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)*, 2019: IEEE, pp. 140-142.

[5] E. C. Oerke, "Crop losses to pests," *Journal of Agricultural Science,* Review vol. 144, no. 1, pp. 31-43, 2006, doi: 10.1017/S0021859605005708.

[6] M. D. K. Owen, "Diverse Approaches to Herbicide-Resistant Weed Management," *Weed Science,* Article vol. 64, no. sp1, pp. 570-584, 2016, doi: 10.1614/WS-D-15-00117.1.

[7] M. W. Ramsden, S. L. Kendall, S. A. Ellis, and P. M. Berry, "A review of economic thresholds for invertebrate pests in UK arable crops," *Crop Protection,* vol. 96, pp. 30-43, 2017/06/01/ 2017, doi: https://doi.org/10.1016/j.cropro.2017.01.009.

[8] J. Strassemeyer, D. Daehmlow, A. Dominic, S. Lorenz, and B. Golla, "SYNOPS-WEB, an online tool for environmental risk assessment to evaluate pesticide strategies on field level," *Crop protection,* vol. 97, pp. 28-44, 2017.

[9] F. LÓPEZ-GRANADOS, "Weed detection for site-specific weed management: mapping and real-time approaches," *Weed Research,* vol. 51, no. 1, pp. 1-11, 2011, doi: 10.1111/j.1365-3180.2010.00829.x.

[10] D. C. Slaughter, D. K. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture,* vol. 61, no. 1, pp. 63-78, 2008/04/01/ 2008, doi: https://doi.org/10.1016/j.compag.2007.05.008.

[11] M. J. Aitkenhead, I. A. Dalgetty, C. E. Mullins, A. J. S. McDonald, and N. J. C. Strachan, "Weed and crop discrimination using image analysis and artificial intelligence methods," *Computers and Electronics in Agriculture,* vol. 39, no. 3, pp. 157-171, 2003/08/01/ 2003, doi: https://doi.org/10.1016/S0168-1699(03)00076-0.

[12] X. P. Burgos-Artizzu, A. Ribeiro, M. Guijarro, and G. Pajares, "Real-time image processing for crop/weed discrimination in maize fields," *Computers and Electronics in Agriculture,* vol. 75, no. 2, pp. 337-346, 2011/02/01/ 2011, doi: https://doi.org/10.1016/j.compag.2010.12.011.

[13] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," *Expert Systems with Applications,* vol. 39, no. 8, pp. 7562-7573, 2012/06/15/ 2012, doi: https://doi.org/10.1016/j.eswa.2012.01.073.

[14] P. R. Eddy, A. M. Smith, B. D. Hill, D. R. Peddle, C. A. Coburn, and R. E. Blackshaw, "Weed and crop discrimination using hyperspectral image data and reduced bandsets," *Canadian Journal of Remote Sensing,* vol. 39, no. 6, pp. 481-490, 2014/01/01 2014, doi: 10.5589/m14-001.

[15] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, "Automatic crop detection under field conditions using the HSV colour space and morphological operations," *Computers and Electronics in Agriculture,* vol. 133, pp. 97-107, 2017/02/01/ 2017, doi: https://doi.org/10.1016/j.compag.2016.11.021.

[16] I. Filella and J. Penuelas, "The red edge position and shape as indicators of plant chlorophyll content, biomass and hydric status," *International Journal of Remote Sensing,* vol. 15, no. 7, pp. 1459-1470, 1994/05/10 1994, doi: 10.1080/01431169408954177.

[17] A.-I. de Castro, M. Jurado-Expósito, M.-T. Gómez-Casero, and F. López-Granados, "Applying neural networks to hyperspectral and multispectral field data for discrimination of cruciferous weeds in winter crops," *The Scientific World Journal,* vol. 2012, 2012.

[18] W. Deng, Y. Huang, C. Zhao, and X. Wang, "Discrimination of crop and weeds on visible and visible/near-infrared spectrums using support vector machine, artificial neural network and decision tree," *Sens. Transducers,* vol. 26, pp. 26-34, 2014.

[19] R. S. Fletcher and K. N. Reddy, "Random forest and leaf multispectral reflectance data to differentiate three soybean varieties from two pigweeds," *Computers and Electronics in Agriculture,* vol. 128, pp. 199-206, 2016/10/01/ 2016, doi: https://doi.org/10.1016/j.compag.2016.09.004.

[20] S. Raymond, P. Hilton, and R. Gabric, "Intelligent crop spraying: a prototype development," in *Proceedings: 1st International Conference in Sensing Technology–2005 Nov*, 2005, pp. 21-23.

[21] S. Askraba, A. Paap, K. Alameh, J. Rowe, and C. Miller, "Laser-Stabilized Real-Time Plant Discrimination Sensor for Precision Agriculture," *IEEE Sensors Journal,* vol. 16, no. 17, pp. 6680-6686, 2016, doi: 10.1109/JSEN.2016.2582908.

[22] M. S. Colgan, C. A. Baldeck, J.-B. Féret, and G. P. Asner, "Mapping Savanna Tree Species at Ecosystem Scales Using Support Vector Machine Classification and BRDF Correction on

Airborne Hyperspectral and LiDAR Data," *Remote Sensing,* vol. 4, no. 11, pp. 3462-3480, 2012. [Online]. Available: https://www.mdpi.com/2072-4292/4/11/3462.

[23] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," *Machine Learning,* vol. 46, no. 1, pp. 389-422, 2002/01/01 2002, doi: 10.1023/A:1012487302797.

[24] H. Hernault, H. Prendinger, D. A. Duverle, and M. Ishizuka, "HILDA: A discourse parser using support vector machine classification," *Dialogue and Discourse,* vol. 1, no. 3, pp. 1-33, 2010.

[25] Y. Ma and G. Guo, *Support vector machines applications*. Springer, 2014.

[26] X.-Y. Wang, T. Wang, and J. Bu, "Color image segmentation using pixel wise support vector machine classification," *Pattern Recognition,* vol. 44, no. 4, pp. 777-787, 2011/04/01/ 2011, doi: https://doi.org/10.1016/j.patcog.2010.08.008.

[27] W. S. Lee, V. Alchanatis, C. Yang, M. Hirafuji, D. Moshou, and C. Li, "Sensing technologies for precision specialty crop production," *Computers and Electronics in Agriculture,* vol. 74, no. 1, pp. 2-33, 2010/10/01/ 2010, doi: https://doi.org/10.1016/j.compag.2010.08.005.

[28] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* vol. 5, no. 4, pp. 115-133, 1943.

[29] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458,* 2015.

[30] H. S. Abdullahi, R. Sheriff, and F. Mahieddine, "Convolution neural network in precision agriculture for plant image recognition and classification," in *2017 Seventh International Conference on Innovative Computing Technology (Intech), Ieee, Londrés*, 2017, pp. 1-3.

[31] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 5, pp. 778-782, 2017.

[32] H. Yalcin and S. Razavi, "Plant classification using convolutional neural networks," in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2016: IEEE, pp. 1-5.

[33] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture,* vol. 147, pp. 70-90, 2018.

[34] A. Kapoor, S. I. Bhat, S. Shidnal, and A. Mehra, "Implementation of IoT (Internet of Things) and Image processing in smart agriculture," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2016: IEEE, pp. 21-26.

[35] D. Ball *et al.*, "Farm workers of the future: Vision-based robotics for broad-acre agriculture," *IEEE Robotics & Automation Magazine,* vol. 24, no. 3, pp. 97-107, 2017.
[36] M. Arya, K. Anjali, and D. Unni, "Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino," in *2018 International Conference on Power, Signals, Control and Computation (EPSCICON)*, 2018: IEEE, pp. 1-5.
[37] A. Li, "Color Image Segmentation Method and Its Application in Agriculture," *Revista de la Facultad de Agronomia de la Universidad del Zulia,* vol. 36, no. 3, 2019.
[38] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv,* 2016.
[39] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv,* 2014.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.
[44] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167,* 2016.
[45] E. Real *et al.*, "Large-scale evolution of image classifiers," *arXiv,* 2017.
[46] P. Verbancsics and J. Harguess, "Generative neuroevolution for deep learning," *arXiv,* 2013.
[47] P. Verbancsics and J. Harguess, "Image classification using generative neuro evolution for deep learning," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, 2015: IEEE, pp. 488-493.
[48] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578,* 2016.
[49] R. Goebel *et al.*, "Explainable AI: the new 42?," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 2018: Springer, pp. 295-303.
[50] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, "What do we need to build explainable AI systems for the medical domain?," *arXiv,* 2017.
[51] M. Mamourian, K. Milani Shirvan, R. Ellahi, and A. B. Rahimi, "Optimization of mixed convection heat transfer with entropy

generation in a wavy surface square lid-driven cavity by means of Taguchi approach," *Int. J. Heat Mass Transfer,* vol. 102, pp. 544-554, 2016/11/01/ 2016, doi: https://doi.org/10.1016/j.ijheatmasstransfer.2016.06.056.

[52] M. K. Balki, C. Sayin, and M. Sarıkaya, "Optimization of the operating parameters based on Taguchi method in an SI engine used pure gasoline, ethanol and methanol," *Fuel,* vol. 180, pp. 630-637, 2016/09/15/ 2016, doi: https://doi.org/10.1016/j.fuel.2016.04.098.

[53] W.-C. Chen, M.-H. Nguyen, W.-H. Chiu, T.-N. Chen, and P.-H. J. T. I. J. o. A. M. T. Tai, "Optimization of the plastic injection molding process using the Taguchi method, RSM, and hybrid GA-PSO," journal article vol. 83, no. 9, pp. 1873-1886, April 01 2016, doi: 10.1007/s00170-015-7683-0.

[54] A. Tsiolikas, D. Tsiamitros, K. Kitsakis, J. Kechagias, N. Mastorakis, and S. D. Kaminaris, "Optimization of neural network parameters using Taguchi Robust Design: Application in plasma arc cutting process," in *2017 Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, 24-27 Aug. 2017 2017, pp. 57-61, doi: 10.1109/MCSI.2017.19.

[55] M. Peker, "A new approach for automatic sleep scoring: Combining Taguchi based complex-valued neural network and complex wavelet transform," *Comput. Methods Programs Biomed.,* vol. 129, pp. 203-216, 2016/06/01/ 2016, doi: https://doi.org/10.1016/j.cmpb.2016.01.001.

[56] T. M. Patel and N. M. J. A. I. Bhatt, "Optimizing Neural Network Parameters Using Taguchi's Design of Experiments Approach: An Application for Equivalent Stress Prediction Model of Automobile Chassis," journal article vol. 1, no. 4, pp. 381-389, November 01 2018, doi: 10.1007/s42154-018-0045-5.

[57] M. A. Peters, "Deep learning, education and the final stage of automation," *Educational Philosophy and Theory,* vol. 50, no. 6-7, pp. 549-553, 2018/05/12 2018, doi: 10.1080/00131857.2017.1348928.

[58] A. Zeng *et al.*, "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May-3 June 2017 2017, pp. 1386-1383, doi: 10.1109/ICRA.2017.7989165.

[59] J.-P. Correa-Baena *et al.*, "Accelerating Materials Development via Automation, Machine Learning, and High-Performance Computing," *Joule,* vol. 2, no. 8, pp. 1410-1420, 2018/08/15/ 2018, doi: https://doi.org/10.1016/j.joule.2018.05.009.

[60] M. Bacco *et al.*, "Smart farming: Opportunities, challenges and technology enablers," in *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, 8-9 May 2018 2018, pp. 1-6, doi: 10.1109/IOT-TUSCANY.2018.8373043.

[61] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big Data for Remote Sensing: Challenges and

Opportunities," *Proceedings of the IEEE,* vol. 104, no. 11, pp. 2207-2219, 2016, doi: 10.1109/JPROC.2016.2598228.

[62] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of “big data” on cloud computing: Review and open research issues," *Information Systems,* vol. 47, pp. 98-115, 2015/01/01/ 2015, doi: https://doi.org/10.1016/j.is.2014.07.006.

[63] A. Kamilaris, A. Kartakoullis, and F. X. Prenafeta-Boldú, "A review on the practice of big data analysis in agriculture," *Computers and Electronics in Agriculture,* vol. 143, pp. 23-37, 2017/12/01/ 2017, doi: https://doi.org/10.1016/j.compag.2017.09.037.

[64] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon,* vol. 4, no. 11, p. e00938, 2018/11/01/ 2018, doi: https://doi.org/10.1016/j.heliyon.2018.e00938.

[65] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging,* vol. 9, no. 4, pp. 611-629, 2018.

[66] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*, 2016: IEEE, pp. 1-6.

[67] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 2017: IEEE, pp. 1-8.

[68] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2015: ACM, pp. 161-170.

[69] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149,* 2015.

[70] E. Real *et al.*, "Large-scale evolution of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017: JMLR. org, pp. 2902-2911.

[71] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, 2019, vol. 33, pp. 4780-4789.

[72] M. Bettoni, G. Urgese, Y. Kobayashi, E. Macii, and A. Acquaviva, "A convolutional neural network fully implemented on FPGA for embedded platforms," in *2017 New Generation of CAS (NGCAS)*, 2017: IEEE, pp. 49-52.

[73] J. J. Martínez, J. Garrigós, J. Toledo, and J. M. Ferrández, "An efficient and expandable hardware implementation of multilayer cellular neural networks," *Neurocomputing,* vol. 114, pp. 54-62, 2013.
[74] P. Symonds, A. Paap, K. Alameh, J. Rowe, and C. Miller, "A real-time plant discrimination system utilising discrete reflectance spectroscopy," *Computers and Electronics in Agriculture,* vol. 117, pp. 57-69, 2015.
[75] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.
[76] S. Askraba, A. Paap, K. Alameh, and J. Rowe, "Design of laser multi-beam generator for plant discrimination," in *High Capacity Optical Networks and Enabling Technologies (HONET), 2011*, 2011: IEEE, pp. 26-29.
[77] S. Askraba, "Optimization of an optoelectronics-based plant real-time discrimination sensor for precision agriculture," (in eng), *Journal of Lightwave Technology,* vol. 31, no. 5, p. 822, 2013.
[78] A. J. Paap, "Development of an optical sensor for real-time weed detection using laser based spectroscopy," 2014.
[79] B. Liu, Y. M. Yue, R. Li, W. J. Shen, and K. L. Wang, "Plant Leaf Chlorophyll Content Retrieval Based on a Field Imaging Spectroscopy System," (in English), *Sensors,* Article vol. 14, no. 10, pp. 19910-19925, Oct 2014, doi: 10.3390/s141019910.
[80] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters,* vol. 27, no. 8, pp. 861-874, 2006.
[81] S. Ioffe and C. J. a. p. a. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
[82] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806,* 2014.
[83] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*, 2009: ACM, pp. 609-616.
[84] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
[85] G. J. Q. R. Taguchi, New York, "Taguchi techniques for quality engineering," 1987.
[86] U. Grömping and H. Xu, "Generalized resolution for orthogonal arrays," *The Annals of Statistics,* vol. 42, no. 3, pp. 918-939, 2014.
[87] C.-S. Chen, J. Lu, and K.-K. Ma, *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei,*

Taiwan, November 20-24, 2016, Revised Selected Papers. Springer, 2017.

[88] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," *arXiv preprint arXiv:1705.10694,* 2017.

[89] S. Akbarzadeh, S. Ahderom, and K. Alameh, "A Statistical Approach to Provide Explainable Convolutional Neural Network Parameter Optimization," *International Journal of Computational Intelligence Systems,* 2019.

[90] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, no. 3, pp. 211-252, 2015.

[91] O. Bousquet, U. von Luxburg, and G. Rätsch, *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*. Springer, 2011.

[92] Y. Hechtlinger, P. Chakravarti, and J. Qin, "Convolutional neural networks generalization utilizing the data graph structure," 2016.