

Computervisie voor precisielandbouw

Jana Wieme

Studentennummer: 01201313

Promotoren: prof. dr. ir. Pieter Simoens, prof. dr. ir. Bart Dhoedt

Begeleiders: ing. Sam Leroux, dr. ir. Simon Cool (ILVO), ir. Ruben Van De Vijver (ILVO)

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2018-2019

Computervisie voor precisielandbouw

Jana Wieme

Studentennummer: 01201313

Promotoren: prof. dr. ir. Pieter Simoens, prof. dr. ir. Bart Dhoedt

Begeleiders: ing. Sam Leroux, dr. ir. Simon Cool (ILVO), ir. Ruben Van De Vijver (ILVO)

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2018-2019

Toelating tot bruikleen

De auteur geeft de toelating deze scriptie voor consultatie beschikbaar te stellen en delen van de scriptie te kopiëren voor persoonlijk gebruik.

Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze scriptie.

Jana Wieme, juni 2019

Voorwoord

Met deze masterproef besluit ik mijn opleiding als Master of Science in de industriële wetenschappen: informatica. Een opleiding vol behulpzame docenten en fijne medestudenten waar ik kennismaakte met een waaier van mogelijkheden.

Een jaar geleden koos ik voor een onderwerp in een sector waar ik enkele jaren ervoor nog nooit van gehoord had: precisielandbouw. Het gebruik van informatica in de precisielandbouw leek me een ontluikend onderzoeksgebied waarin nog vele uitdagingen te vinden zijn. Dit multidisciplinaire onderwerp deed een nieuwe wereld voor me open gaan. Ik kwam terecht in sector met mensen uit diverse vooropleidingen, waarbij het samenbrengen van al deze kennis tot nog een beter resultaat leidde. Ik zag dit onderwerp dan ook als een uitdaging om een meerwaarde te bieden met mijn opgedane kennis in de opleiding boordevol informatica.

In het bijzonder wil ik dr. ir. Simon Cool en ir. Ruben Van De Vijver van het ILVO hiervoor ten zeerste bedanken. Jullie enthousiasme en overtuigingskracht tijdens de voorstelling van dit onderwerp zorgde ervoor dat ik er helemaal voor wou gaan. Ook tijdens het werkjaar namen jullie mij vol enthousiasme mee in het verhaal van de precisielandbouw, stuurden jullie me bij waar nodig en voorzagen jullie mij van de nodige middelen om mijn masterproef te kunnen realiseren. Een aantal lijntjes tekst zijn eigenlijk niet voldoende om mijn dank aan jullie uit te drukken.

Daarnaast wil ik ook Jelle en Ruben bedanken voor hun bureau met mij te delen. Dankzij jullie kwam ik met veel plezier aan mijn masterproef werken. Ik keek telkens uit naar de interessante gesprekken en discussies zowel over mijn onderwerp als over andere projecten.

Ik ben het ILVO, en meer specifiek de afdeling agrotechniek, dan ook heel dankbaar om me de kans te geven deze masterproef uit te werken. Dankzij jullie besefte ik waar ik in de toekomst mee verder wil gaan. Ik kijk er dan ook naar uit om dit verhaal binnenkort verder te zetten.

Vervolgens wil ik mijn promotoren, prof. dr. ir. Pieter Simoens en prof. dr. ir. Bart Dhoedt bedanken voor hun kritische blik en feedback tijdens de tussentijdse besprekingen. Dankzij jullie raakte ik net dat stapje verder in het onderzoeken.

Ook wil ik ing. Sam Leroux bedanken voor alle begeleiding in mijn eerste onderzoekservaring. Jouw ervaring, inzichten en constructieve opmerkingen gidsten me doorheen deze masterproef. Bedankt Sam voor het geduld, het duwtje in de rug waar nodig en de zovele suggesties, dankzij jou heb ik mijn kennis uit de opleiding nog zoveel kunnen uitbreiden. Ook mijn dank voor jou kan ik niet zomaar in een tekstje uitdrukken.

Graag zou ik ook enkele vrienden willen bedanken. Jeroen, Cedric, Michael, bedankt voor de steun, motivatie en de feedback die jullie me gaven.

Evenzeer nog een woord van dank aan diegene rondom mij. Eerst en vooral mijn vriend Jef, bedankt, je was mijn steun en toeverlaat en stond altijd klaar om me tot rust te brengen. Mijn broers, Jelle en Jorg, bedankt voor de kritische blik en interessante discussies waarbij jullie onderzoeks- en informaticakennis ervoor zorgde dat ik weer een stapje verder geraakte. Ook mijn tante en nonkel, Kathleen en Filip, bedankt voor de steun en de snelle hulp voor de herstellingen wanneer mijn laptops het even voor bekeken lieten.

Tenslotte nog een speciaal woord van dank aan mijn ouders, Carine en Peter. Ze gaven me de kans om persoonlijk en academisch te groeien in goede omstandigheden. Ik wil hen oprecht bedanken voor de financiële en morele steun, het enorme geduld en het blijvende geloof in mij.

Bedankt allemaal om me de kennis aan te reiken, te leren gebruiken en me te motiveren ervoor te gaan. Om het in de woorden van J. W. von Goethe, Duits filosoof, wetenschapper en schrijver te besluiten:

“Knowing is not enough, we must apply. Willing is not enough, we must do.”

Jana Wieme, juni 2019

Computervisie voor precisielandbouw

door Jana WIEME

Scriptie ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Promotoren: Prof. Dr. Ir. P. Simoens, Prof. Dr. Ir. B. Dhoedt
Begeleiders: Ing. S. Leroux, Dr. Ir. S. Cool, Ir. R. Van De Vijver

Vakgroep Informatietechnologie
Voorzitter: Prof. Dr. Ir. B. Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Universiteit Gent
Academiejaar 2018-2019

Samenvatting

Door de toename van de wereldbevolking stijgt de druk op de voedselproductie en de impact op het milieu jaar na jaar. Deze toename verloopt echter niet gelijkmatig aan de toename van de hoeveelheid beschikbare grond voor voedselproductie, waardoor de productie op eenzelfde hoeveelheid oppervlakte drastisch omhoog moet, waarbij – indien mogelijk – de impact op het milieu niet toeneemt. De laatste jaren werd sterk geïnvesteerd in de verzameling van data binnen de landbouw en voedselproductie om onderzoek te kunnen doen naar systemen die hieraan tegemoetkomen. Dit valt binnen het domein van precisielandbouw.

Deze masterproef onderzoekt de mogelijkheden van deep learning binnen de computervisie-taken in het domein van de precisielandbouw. Het werk wordt opgesplitst in twee delen met elk hun eigen doelstelling. Een eerste deel heeft als doel het onderzoeken en ontwikkelen van de haalbaarheid van een automatisch supervised classificatiesysteem in twee diverse cases: herkenning van ritnaalden en de oriëntatiebepaling van bloemkolen.

Om bovenstaand systeem te realiseren, is er nood aan een grote hoeveelheid gelabelde data, wat een kost- en tijdrovende taak is. Het tweede deel van deze masterproef richt zich bijgevolg op het onderzoeken of het haalbaar is om deze bottleneck van menselijke interventie te reduceren. Daarom stelt deze masterproef een proof of concept voor van een alternatieve aanpak op basis van een convolutionele autoencoder en semi-supervised plantdetectie in overzichtsfoto's van velden gemaakt door drones.

Trefwoorden

Precisielandbouw, drones, computervisie, deep learning, autoencoder, semi-supervised learning, classificatie, objectdetectie

Computervision for precision agriculture

Jana Wieme

Supervisors: Pieter Simoens, Bart Dhoedt

Counsellors: Sam Leroux, Simon Cool, Ruben Van De Vijver

Abstract—The impact of population pressure on food production and the environment is increasing every year. The world’s population is growing fast, but the amount of available land for food production doesn’t follow. This implies that the production of food in each available area needs to increase drastically, and if possible, without increasing the impact on the environment. In order to find solutions for this global problem, research needs to be performed. For this purpose, substantial investments have been made in recent years to collect data. This research is part of precision agriculture, which aims to improve and develop agricultural systems to create more outputs (i.e. yield) with fewer inputs (i.e. fertilizers).

The goal of this thesis is to investigate the possibilities of deep learning within computer vision tasks concerning precision agriculture. Deep learning has emerged as a promising technique in various areas of science, and offers much potential to improve the current ways of precision agriculture. This thesis covers two different parts, each with their own objectives. The first part aims to investigate whether an automatic classification system is possible in two different cases: the recognition of wireworms and the orientation of cauliflowers.

To utilize this method, a large amount of labelled data by experts is required, which is an expensive and time-consuming task. To this end, the second part of this thesis focuses on examining whether it is feasible to reduce the bottleneck of human intervention. Accordingly, this thesis proposes a proof of concept of an alternative approach with a convolutional autoencoder and semi-supervised learning for plant detection in field photos taken by drones.

Keywords—Precision agriculture, precision crop farming, computervision, semi-supervised learning, deep learning, autoencoder, classification, objectdetection

I. INTRODUCTION AND OBJECTIVES

THE continuously increasing world population poses a challenge for agriculture: in order to be able to produce enough food, sufficient agricultural land is needed. However, the amount of available land for this food production is not increasing as fast as the population. The need for improvement in the food production pipeline can be considered as high. Precision agriculture tries to deal with this by developing systems that can produce more, with fewer inputs (i.e. herbicides and fertilizers), taking into account the environment. Consequently, the aim of precision agriculture is two-fold: economic and ecological. [1], [2], [3].

The recent technological evolution and rise of artificial intelligence in other domains, caught the attention of precision agriculture researchers. The use of artificial intelligence introduces a lot of new possibilities for the development of smart agricultural systems [1], [2], [4].

An important application of artificial intelligence is computer vision, which deals with making computers ‘seeing’ things, using images and videos as input data. The technological evolution made it possible to collect large amounts of data through the use of different cameras. The possibilities in terms of applications in which computer vision can be used in the context of precision farming are very wide [3]. Some examples are yield prediction, drought-stress assessment, disease detection, weed

detection, classification of different types of crops etc.

This thesis investigates the possibilities of deep learning within computer vision tasks concerning precision agriculture, divided in two parts. The first part (Section III) develops a supervised classification system based on convolutional neural networks in two different cases: the recognition of wireworms and the orientation of cauliflowers.

Even though the technological evolution led to data collection, the data needed by the aforementioned supervised classification systems must be labelled. Since labelling is an expensive and time-consuming task, current research is trying to find solutions for this bottleneck [5]. Besides, there is an enormous variability in nature: a lot of different plants, (visual) differences between plants of the same species, different stages of growth, background variations etc. [4], [6].

The second part of this thesis (Section IV) proposes a new approach for detecting crops in a semi-supervised way, focusing on minimizing the need of human interventions and making it a generally usable system for various applications.

II. RELATED WORK

Currently, the new approach (Section IV) is developed for detecting crops. However, this proof of concept needs to be further investigated and developed in order to create a useful application. For example, instead of detecting the crops, it can be trained to detect weeds. Weed detection is a hot topic in precision agriculture, and could be included in an agricultural robot. The robot will then be able to use herbicides only at the position of the detected weed. In this way, precision weed control can help to reduce the amount of herbicides, without reducing the yield and quality of the crops. The weakest link in those automated systems for weed detection, is the part of detection and identification of the weeds in the huge range of environmental conditions including the spectral similarity of weeds and crops [6], [7].

In recent years, several approaches have been developed for weed detection. These methods generally consist of four phases: data collection, data (pre)processing, the training of models and the evaluation of the method. In literature, there are many variants within these phases [8].

An important difference concerning data collection, is the fact that data can be captured with an Unmanned Ground Vehicle (UGV) or an Unmanned Aerial Vehicle (UAV). Using UGV makes it possible to create real time end-to-end applications, giving the robot the possibility to act immediately based on the information captured by the image. On the other hand, using UAV it is possible to capture larger areas in one images, so

the information about spatial relationship between plants can be used [9].

Another difference in this phase concerns the choice of the spectrum bands. Generally, the camera is capturing the visual spectrum of RGB (red-green-blue), but in some works they extend the spectrum with NIR (near-infrared).

In most methods, plants are separated from the background by preprocessing the image. When NIR is added to the spectrum, this step becomes easier because the difference between vegetation and background is clearly highlighted due to the high reflection of plants performing photosynthesis. Mostly this separation is followed by a clustering phase, clustering the vegetation detected pixels into blobs in order to extract patches containing singular crops or weeds. Grouping pixels generally makes a positive contribution: the network runs faster and it provides additional information [10], [11].

In literature, most methods are using supervised learning [9], [12]. As mentioned above, this learning method has the disadvantage that it requires a large and labelled dataset. Some methods are trying to reduce this bottleneck by using some sort of semi-supervised learning (labelling a subset of the images) [13], or completely using unsupervised learning [5].

Another difference in the training process concerns the use of machine learning techniques versus deep learning. Currently, most of the methods are still using traditional machine learning techniques [9], [13], but deep learning is gaining in popularity [1].

The contribution of this thesis is the development of a general approach for detecting crops with a limited need of labelled data. Little attempts have been made to develop a general and semi-supervised system for crop detection.

III. CLASSIFICATION CASE STUDIES

The first part of this thesis aims to develop a supervised classification system for two different cases: the recognition of wireworms and the orientation of cauliflowers. Both cases used Python 3.5 with open-source machine learning libraries TensorFlow [14] and Keras [15]. Models were trained on a NVIDIA GeForce GTX 960M GPU.

A. Recognition of wireworms

Because of the damage to crops by wireworms (click beetle larvae), the Flemish government started a project to map the distribution of different species for doing a classification. In this thesis an attempt is made to recognize and classify the three species of the *Agriotes* wireworms. This classification is currently being performed by experts who conduct a DNA test. The aim is to investigate whether the computer will be able to detect visual features that can lead towards an automatic classification or not.

For this purpose, a dataset of 1411 RGB images, divided into three categories was made available by the Instituut voor Landbouw-, Visserij- en Voedingsonderzoek (ILVO). The main disadvantage of this dataset is the unbalanced distribution. Therefore, data augmentation (rotation) is performed to get approximately the same number of images (800) for each of the three categories.

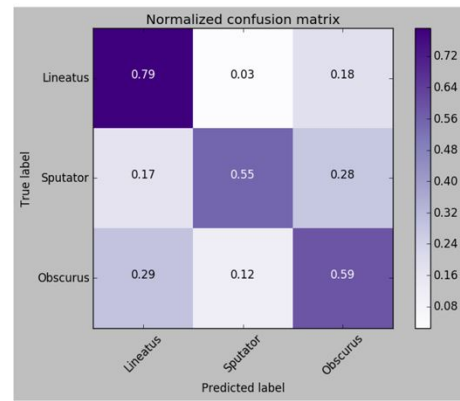


Fig. 1. Normalized confusion matrix for classification of wireworms.

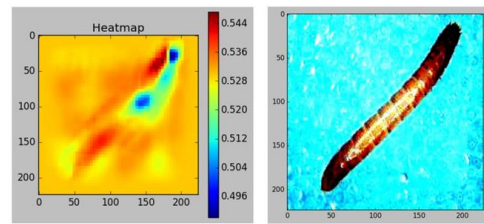


Fig. 2. Heat map of correct predicted *Agriotes lineatus*.

Subsequently, a deep convolutional neural network with four convolutional layers, alternated with max-pooling layers and ending in two fully connected layers is used to predict the classification. After the fully connected layer, a Softmax activation function is used to convert the results of the neural network into probabilities for each of the three classes [16]. After tuning the hyperparameters (i.e. learning rate, L2-penalty, dropout, number of epochs...) a final result of 64,4% accuracy was found. As can be seen in the normalized confusion matrix showed in figure 1, the model is able to correctly classify 79% of the lineatus, but the other two types are more difficult.

Despite the fact that experts recognize and classify based on DNA analysis, the above result is better than an arbitrary accuracy of 33%. The question may arise about which features the model is using to make a decision. This could be useful information for the experts. The technique used to visualize what the model is looking at, is the production of heat maps based on occlusion as written in the work of Zeiler et al. [17] In these heat maps (i.e. figure 2), blue pixels indicate the important parts and red the unimportant ones. The first impression of the heat maps shows that the model is generally looking at the wireworm, in particular at the head. Further analysis of these heat maps must be done in collaboration with experts in order to focus on more specific features.

B. Classification of cauliflowers

In the context of a feasibility study concerning the development of an automated system to transport cauliflowers to the right machine, a supervised classification model was needed in this second case study. The classification that needs to be made is based on the orientation of the cauliflower: normal position

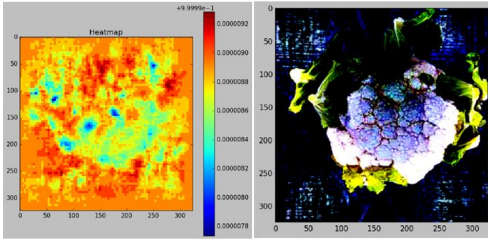


Fig. 3. Heat map of correctly oriented cauliflower.

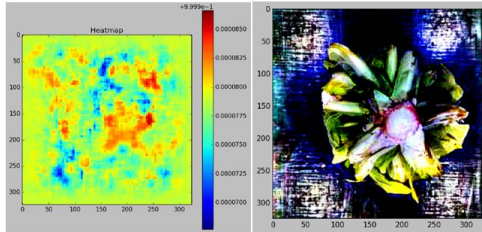


Fig. 4. Heat map of inversely oriented cauliflower.

with the cabbage visible, or inverse position with cabbage oriented to the ground.

Again, the data set was made available by ILVO, containing 449 images (in a controlled situation) with a good relationship between both classes. The deep convolutional neural network used in this case study, is completely analogous to the one in the other classification project.

After some trial and error for the hyperparameters, the final accuracy reached the value of 98,89%. As in the previous classification case study, heat maps were made to provide insights into how the model is classifies the figures. Figure 3 is showing a heat map for a correctly oriented cauliflower, while figure 4 displays an inversely orientation. From these heat maps, it seems that the model is paying attention to the visual cabbage structure (the amount of white visible cabbage), and not to the stem of the cauliflower.

IV. SEMI-SUPERVISED CROP DETECTION

A. Proposed method

As illustrated in the previous sections, the need for intelligent systems that can deal with different natural environments in the agriculture is increasing. This variation in species and environments are, after all, an important obstacle in the development of robust systems that can be used in the crop production. The second bottleneck in the development concerns the human intervention for labelling the captured images.

The second part of this thesis proposes a new approach for detecting crops in a semi-supervised way, focusing on minimizing the need of human interventions and making it a generally usable and transferable system.

The proposed method executes several steps as shown in figure 5. In the following subsections they will be explained step by step.

The method used Python 3.5 with the open-source deep learning platform PyTorch [21] and Tensorflow [14].

A.1 Data collection and preprocessing

The first step in the crop detection pipeline is the collection of useful data. As mentioned in Section II, this can be done in various ways. In this work, multiple RGB images taken by a UAV are stitched together to form an overview picture of a potato field and a small part of the environment. The only preprocessing done in this proof of concept is the normalization of the image to have a more symmetric data distribution. Further preprocessing is avoided to limit human intervention and to stimulate transferability.

A.2 Architecture: autoencoder

The decision to use a convolutional autoencoder as network architecture, is based on the fact that it has proven its usefulness in the concept of anomaly detection in other domains [18]. From this point of view, crops can be seen as anomalies in the large and diverse background of the field. The autoencoder takes during training each time a small part of the overview image, uses the convolutional layers to reduce the image to a feature vector and then uses this feature vector to do the inverse operations to reconstruct the image. The advantage here is the fact that this network is selecting interesting features by itself (unsupervised), because these features are needed to reconstruct the input image.

In most anomaly detection applications using autoencoders, the decision for an anomaly is based on the reconstruction error. In that system, the autoencoder is trained on a set of images in absence of anomalies, such that an anomaly results in a much higher reconstruction error during the test phase. In this proposed method, it is not possible to use that system because of the nature of the input image: it is containing both plant and background. Therefore, this approach uses the correlation between the various feature vectors. Further information about this is given in the next subsection.

The final network architecture, after some trial and error, is shown in figure 6. With this architecture, the system is able to properly reconstruct parts of the input image (image split into samples for training), such that the feature vector contains a relevant set of image features without the need for human interventions.

A.3 Semi-supervised learning

To avoid the expensive and time-consuming procedure of labelling the entire image, this method proposes a system of semi-supervised learning where only a small subset of samples (image parts) must be labelled. This subset contains positive (in this case plants) and negative samples (background). The challenge of this step concerns the fact that the variability of the original image must be covered as much as possible in the subset of labelled samples. In this proof of concept, the selection is made manually, but future work must improve this part of the approach.

After the subset is made, these labelled samples are sent through the encoder part of the autoencoder, which determines their feature vector and thus creates a set of feature vectors of

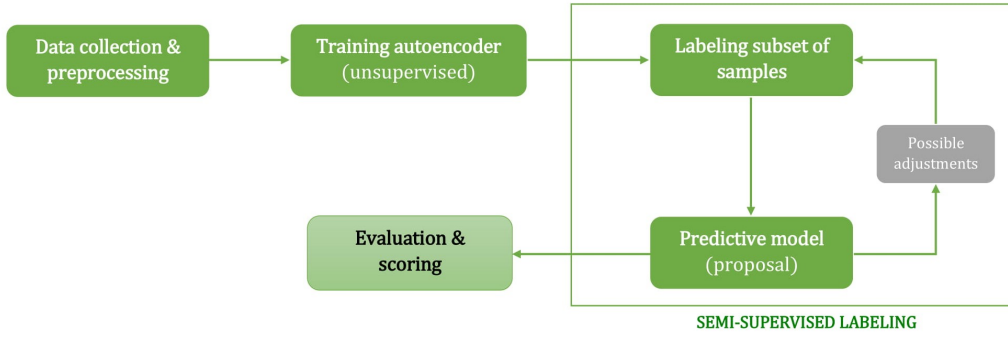


Fig. 5. Flowchart of the proposed method.

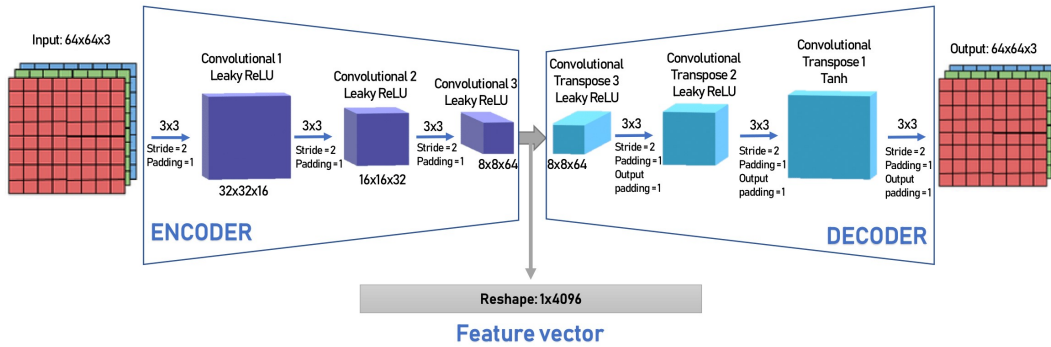


Fig. 6. Network architecture: convolutional autoencoder.

positive and negative samples. In the next step, the entire image is divided by a grid into boxes of the same size as the labelled samples (in this case 64x64 pixels). Each box is sent through the encoder to determine the feature vector as well.

The Canberra distance is used to determine whether a box belongs to the positive or negative class. This metric calculates the (dis)similarity between two feature vectors u and v as shown in equation 1. This distance has proved useful in the field of intrusion detection to detect the anomalous values [19], [20].

$$d(u, v) = \sum_{i=1}^k \frac{|u_i - v_i|}{|u_i| + |v_i|} \quad (1)$$

For each box of the original image, the degree of similarity is measured for all positive and negative samples. The minimum distance and thus the most similar sample, determines to which category the box belongs (positive: crop, negative: background), taking into account the entire set of labelled samples.

The idea behind this semi-supervised approach is to build an application where a farmer can use his expertise to label a subset of samples. Subsequently, a classification suggestion will be given for each box of the grid. The farmer can now make a fine tuning, for example by labelling some extra samples as positive or negative. This process can be repeated multiple times. Furthermore, the network itself doesn't need to be retrained, be-

cause this entire classification and labelling is happening after the training.

A.4 Evaluation of the model

To evaluate the model, it is necessary to have a ground truth. This ground truth is made by drawing bounding boxes around the plants in the original image, and afterwards calculating how much each box in the grid is covered by a bounding box. If more than 35% of a box in the grid is covered, than this box is labelled as positive.

Some critical remarks must be made regarding this labelling. First of all, the labelling is not done by an expert, although it was verified several times to avoid non-labelled crops. Moreover, plants are not rectangular and do not necessarily fit in a bounding box. This results in bounding boxes that also contain a part of background, which also introduce a small error in the accuracy of this evaluation. The threshold of 35%, decided by trial and error on the visual representation of the ground truth, plays an important role in this evaluation system.

Although, the ground truth is only needed for the research and development of the system, it is not important for the future application itself.

B. Results

To evaluate the proof of concept, different metrics were used: accuracy, precision, recall and F1-score. The final results are shown in table I. The high value of accuracy needs to be nuanced

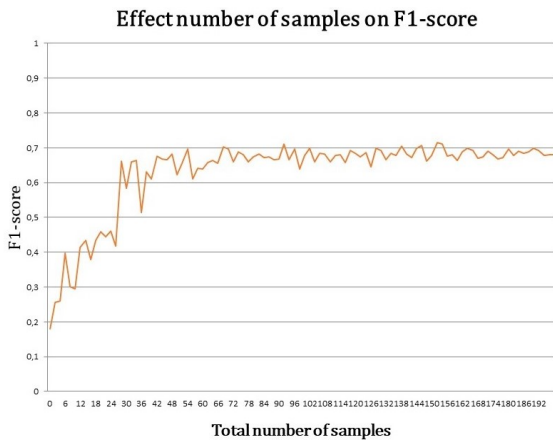


Fig. 7. Graphical representation of the effect of the number of labelled samples on the F1-score.

because of the unbalanced data set: even when labelling everything negative an accuracy of 90% will be reached because 90% of the grid boxes are negative. Therefore, precision (the number of positive labelled boxes in the prediction, being indeed positive in the ground truth), recall (the number of labelled positives in the ground truth that are detected) and their harmonic mean, the F1-score, are more valuable metrics [1].

TABLE I
RESULTS METRICS.

Metric	Value
Precision = $\frac{TP}{TP+FP}$	75,46%
Recall = $\frac{TP}{TP+FN}$	81,03%
F1-score	78,15%
Accuracy	96,03%

The F1-score of 78,15% is a promising result, especially taking into account the small error introduced by the ground truth. This value is slightly below the state of the art concerning detection systems, but it is difficult to compare the results of different approaches because of the wide range of possibilities concerning metrics, types of learning, applications, (non)-realistic data sets etc. Presumably further investigation of this proof of concept can improve this result and so the entire method towards a realistic model.

Since this proof of concept is based on the idea of limiting user intervention, an interesting question is the following: what is the effect of the number of labelled samples for the performance of the model. Therefore, the model described in the previous subsection is used with a varying number of labelled samples (from 1 of each type to 100 of each type). These samples were selected at random from a list of 100 available samples. This experiment was averaged over four runs to minimize potential outliers due to the random selection.

The result of this experiment is visualized in figure 7. At first, the performance increases when a larger number of samples is used. However, it converges to approximately 68%, when more than 35 labelled samples per species are employed. This plateau suggests that an extra effort in labelling the samples might not be useful.

V. CONCLUSION

This thesis investigated some possibilities of deep learning within computer vision tasks concerning precision agriculture.

The first part, Section III, started with an investigation of the possibilities of a classification system for wireworms, a harmful species for agricultural crops. Since this task attempts to classify wireworms visually, the results are in line with the expectations, especially considering that this poses challenges to experts relying on DNA analysis. The model was able to detect some features, but an automated classification is not yet possible. However, further investigation could be done, probably in collaboration with experts, to gain more insight in these features.

Another supervised classification system fits in the context of a feasibility study concerning the development of an automated system to transport cauliflowers to the right machine. Using a simple four-layered convolutional network, the developed model was able to make a classification with an accuracy of 98,89%. Based on this, real-world applications relying on this approach might become possible.

Current bottlenecks in the use of deep learning in computer vision in precision agriculture, are the needs for human intervention, as well as the enormous diversity in nature. The second part of this work, Section IV proposed a new approach for detecting crops in a semi-supervised way, trying to reduce the effects of the bottlenecks as much as possible. This resulted in a proof of concept of a flexible system, in which the user intervention is limited to the labelling of some samples. The resulting F1-score of 78,15% is a promising result. Besides, there is the fact that the effect from the number of samples on the F1-score converges. For future work, it will be interesting to extend this method by investigating some improvements.

Some possible improvements that could be investigated, are described in the related work (Section II). For example, one could think about a vegetation/background separation, blob forming techniques or the introduction of the NIR spectrum.

An important challenge in this new approach, concerns the selection of the samples that must be labeled. An interesting approach to handle this problem, could be the use of K-means clustering of all the feature vectors. In this way, centroids could be identified as relevant structures for the user that needs to label them. This could be implemented as an iterative approach, and as such the concept of active learning might be achieved.

ACKNOWLEDGMENTS

This work was performed as a collaboration between the Department of Information Technology from the University of Ghent (UGent) and ILVO, under supervision of ir. Ruben Van De Vijver (ILVO), dr. ir. Simon Cool (ILVO), ing. Sam Leroux (UGent), prof. dr. ir. Pieter Simoens (UGent) and prof. dr. ir. Bart Dhoedt (UGent). I want to thank them for the support and guidance throughout the realization of this master thesis.

REFERENCES

- [1] A. Kamilaris, F. X. Prenafeta-Bold, *Deep learning in agriculture: A survey* Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.

- <http://www.sciencedirect.com/science/article/pii/S0168169917308803>
- [2] S. Banu, *Precision Agriculture: Tomorrow's Technology for Today's Farmer*, Journal of Food Processing & Technology, vol. 6: 468, 2015.
 - [3] S. Cool, *Precisielandbouw: een samenvatting*, ILVO Agrotechniek, 2019.
 - [4] N. Zhang, M. Wang, N. Wang, *Precision agriculture-a worldwide overview*, Computers and Electronics in Agriculture, vol. 36, pp. 113 - 132, 2002.
 - [5] M. D. Bah, A. Hafiane, R. Canals, *Deep Learning with unsupervised data labeling for weeds detection on UAV images*, CoRR, vol. abs/1805.12395, 2018. <https://dblp.org/rec/bib/journals/corr/abs-1805-12395>
 - [6] D. Slaughter, D. Giles, D. Downey, *Autonomous robotic weed control systems: A review*, Computers and Electronics in Agriculture, vol. 61, no. 1, pp. 63 - 78, 2008 <http://www.sciencedirect.com/science/article/pii/S0168169907001688>
 - [7] P. Lottes, M. Hoferlin, S. Sander, M. Mter, P. Schulze, L. C. Stachniss, *An effective classification system for separating sugar beets and weeds for precision farming applications*, 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 5157-5163, 2016.
 - [8] E. Hamuda, M. Glavin, E. Jones, *A survey of image processing techniques for plant extraction and segmentation in the field*, Computers and Electronics in Agriculture, vol. 125, pp. 184 - 199, 2016. <http://www.sciencedirect.com/science/article/pii/S0168169916301557>
 - [9] P. Lottes, R. Khanna, J. Pfeifer, R. Siegart, C. Stachniss, *UAV-based crop and weed classification for smart farming*, 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3024-3031, 2017.
 - [10] A. Milioto, P. Lottes, C. Stachniss *Real-time Blob-wise Sugar Beets VS Weeds Classification For Monitoring Fields Using Convolutional Neural Networks*, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-2/W3, pp. 41-48, 2017. <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2-W3/41/2017/>
 - [11] M. Fawakherji, A. Youssef, D. Bloisi, A. Pretto, D. Nardi, *Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation* 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 146-152, 2019.
 - [12] C. Hung, Z. Xu, S. Sukkarieh, *Feature Learning Based Approach for Weed Classification Using High Resolution Aerial Images from a Digital Camera Mounted on a UAV*, Remote Sensing, vol. 6, pp. 12037-12054, 2014.
 - [13] M. Prez-Ortiz, J. M. Pea, P. A. Gutierrez, J. Torres-Sanchez, C. Hervs-Martnez, F. Lpez-Granados, *Selecting patterns and features for between-and within- crop-row weed mapping using UAV-imagery*, Expert Systems with Applications, vol. 47, pp. 85-94, 2016.
 - [14] Tensorflow, <https://www.tensorflow.org/>
 - [15] Keras, <https://keras.io/>
 - [16] Stanford University *CS231n: Convolutional Neural Networks for Visual Recognition*, <http://cs231n.stanford.edu/>
 - [17] M. D. Zeiler, R. Fergus, *Visualizing and Understanding Convolutional Networks*, CoRR, vol. abs/1311.2901, 2018. <https://dblp.org/rec/bib/journals/corr/ZeilerF13>
 - [18] R. Chalapathy, S. Chawla, *Deep Learning for Anomaly Detection: A Survey*, CoRR, vol. abs/1901.03407, 2019. <http://arxiv.org/abs/1901.03407>
 - [19] S. M. Emran, N. Ye, *Robustness of Canberra metric in computer intrusion detection*, Proceedings of the IEEE Workshop on Information Assurance and Security, 2001.
 - [20] A. Kumar, A. Abdelhadi, C. Clancy, *Novel anomaly detection and classification schemes for Machine-to-Machine uplink*, 2018 IEEE International Conference on Big Data (Big Data), pp. 1284-1289, 2018.
 - [21] PyTorch, <https://pytorch.org/>

Inhoudsopgave

Lijst van figuren	xiv
Lijst van tabellen	xvii
Lijst van afkortingen	xviii
1 Inleiding	1
1.1 Doelstelling	2
1.2 Opbouw	2
2 Computervisie	4
2.1 Van artificiële intelligentie tot computervisie	4
2.2 Artificiële neurale netwerken	6
2.2.1 Opbouw van een neuraal netwerk	6
2.2.2 Convolutionele neurale netwerken	10
2.2.3 Autoencoder (AE)	14
2.3 Soorten leren	15
2.3.1 Supervised vs. unsupervised learning	16
2.3.2 Semi-supervised learning	17
3 Literatuurstudie	18

3.1	Precisielandbouw	18
3.2	Computervisie in de precisielandbouw	22
3.2.1	Algemene problematiek: variabiliteit	22
3.2.2	Dataverzameling	23
3.2.3	Data (pre)-processing	25
3.2.4	Training van modellen	29
3.2.5	Evaluatie van het systeem	32
3.2.6	Overzicht	33
4	Classificatieproblemen in de precisielandbouw	36
4.1	Classificatie van ritnaalden	37
4.1.1	Ontwerp en implementatie	37
4.1.2	Resultaten	44
4.1.3	Besluit	48
4.2	Classificatie van bloemkolen op basis van de oriëntatie	48
4.2.1	Ontwerp en implementatie	49
4.2.2	Resultaten	50
4.2.3	Besluit	51
5	Semi-supervised detectie van planten	55
5.1	Ontwerp en implementatie	58
5.1.1	Dataverzameling en pre-processing	58
5.1.2	Netwerkarchitectuur: autoencoder	59
5.1.3	Semi-supervised labeling	61
5.1.4	Evaluatie van het model	64

5.1.5	Gebruikte technologieën	66
5.2	Resultaten	66
5.2.1	Netwerkarchitectuur	66
5.2.2	Evaluatie van de proof of concept	66
5.2.3	Evaluatie labeling subset	68
5.2.4	Bijgesneden afbeelding	72
5.2.5	Test proof of concept op nieuwe afbeelding	72
5.3	Besluit	73
6	Conclusie en perspectieven	75
6.1	Conclusie	75
6.2	Perspectieven	77
	Bibliografie	80
	Bijlagen	87

Lijst van figuren

2.1	Overzicht van artificiële intelligentie tot deep learning.	5
2.2	Schematische weergave van menselijk neuron (links) en de mathematische voorstelling (rechts). Afbeelding overgenomen uit [1].	7
2.3	Grafische weergave van mogelijke activatiefuncties.	9
2.4	Feed-forward versus recurrent neuraal netwerk. Afbeelding overgenomen uit [2].	9
2.5	Convolutionele laag: 32x32x3 inputafbeelding met 5x5x3 filter resulteert in featuremap van 28x28x1 door convolutie. Afbeelding overgenomen uit [3].	11
2.6	Visuele voorstelling van stride S=2. Afbeelding overgenomen uit [4].	12
2.7	Visuele weergave van filters. Afbeelding overgenomen uit [5].	12
2.8	Voorbeeld van pooling met filter 2x2 en stride 2. Afbeelding overgenomen uit [1]	13
2.9	Werking van een autoencoder. Afbeelding overgenomen uit [6].	15
2.10	Bias-variance voorbeelden: rechts underfitting, links overfitting. Afbeelding overgenomen uit [6].	17
2.11	Grafische weergave bias-variance trade-off. Afbeelding overgenomen uit [6].	17
3.1	Traditionele landbouw vs. precisielandbouw. Afbeelding overgenomen uit [7].	19
3.2	Cyclus van precisielandbouw. Afbeelding overgenomen uit [7].	20
3.3	Technologie in precisielandbouw. Afbeelding overgenomen uit [7].	21
3.4	Traditionele machine learning approach vs. deep learning approach. Afbeelding overgenomen uit [8].	32

4.1	Genormaliseerde confusiematrix na random forest classifier.	40
4.2	Weergave van verdeling van speciën, uitzetting 'distance between the pits' in functie van 'body width'.	41
4.3	Links: loss-functie - Rechts: genormaliseerde confusiematrix behorende bij CNN met originele afbeeldingen.	43
4.4	Visualisatie netwerkmodel classificatie van ritnaalden.	44
4.5	Genormaliseerde confusie matrix bij classificatie van ritnaalden.	45
4.6	Genormaliseerde confusie matrix one-VS-all classificatie van ritnaalden.	46
4.7	Genormaliseerde confusie matrix 'obscurus VS rest'.	46
4.8	Heatmap van een correct voorspelde lineatus.	47
4.9	Heatmap van een correct voorspelde sputator.	47
4.10	Heatmap van een obscurus, voorspeld als lineatus.	47
4.11	Geschaald en gecropt (verkleinde weergave).	49
4.12	Visualisatie netwerkmodel voor classificatie bloemkolen.	50
4.13	Links: verliesfunctie - Rechts: accuracy netwerk classificatie bloemkolen.	51
4.14	Confusiematrix classificatie van bloemkolen. Links: absolute - Rechts: relatieve.	52
4.15	Heatmap bloemkool 'kool boven' 1.	52
4.16	Heatmap bloemkool 'kool boven' 2.	53
4.17	Heatmap bloemkool 'kool boven' 3.	53
4.18	Heatmap bloemkool 'hart omhoog' 1.	53
4.19	Heatmap bloemkool 'hart omhoog' 2.	54
4.20	Heatmap bloemkool 'hart omhoog' 3.	54
5.1	Flowchart van de nieuwe methode.	57
5.2	Netwerkarchitectuur autoencoder.	60

5.3	Links: ground truth met drempel = 30%. Midden: ground truth met drempel = 35%. Rechts: ground truth met drempel = 40%.	65
5.4	Gelabelde punten (fuchsia) en bijhorend gelabeld vakje in raster van 64x64 pixels.	65
5.5	Links: voorbeeld reconstructie positieve sample. Rechts: voorbeeld reconstructie negatieve sample.	66
5.6	Links: gekozen positieve (groen) en negatieve (rood) sampels. Rechts: eindresultaat. Rode: FN, oranje: FN, groene: TP.	67
5.7	Grafische weergave van de verhouding tussen het aantal positieve en negatieve samples. Links in het geval van pos=neg-10 en rechts pos=neg+10.	69
5.8	Grafische weergave van de invloed van verhouding positief/negatief gelabelde samples op precision (linksboven), recall (rechtsboven), accuracy (linksonder) en F1-score (rechtsonder).	70
5.9	Grafische weergave van de invloed van een al dan niet gestuurde keuze van gelabelde samples op precision (linksboven), recall (rechtsboven), accuracy (linksonder) en F1-score (rechtsonder).	71
5.10	Grafische weergave van de invloed van het aantal gelabelde samples op de F1-score.	71
5.11	Bijgesneden afbeelding.	73
5.12	Resultaat na toepassen methode op veld met spinazieplanten.	74
1	Weergave van diverse gegevens in functie van elkaar omtrent classificatie van ritnaalden.	88
2	Overzichtsfoto aardappelveld.	89

Lijst van tabellen

3.1	Diverse metrieken [9].	32
3.2	Publicaties die gebruik maken van CV technieken voor (onkruid)detectie.	34
3.3	Publicaties die gebruik maken van CV technieken voor (onkruid)detectie 2.	35
4.1	Voorbeelddata uit Excel-bestand.	38
4.2	Resultaten van diverse classifiers op de gemeten karakteristieken.	39
4.3	Keuze voor enkele hyperparameters in netwerk voor classificatie van ritnaalden.	45
4.4	Keuze voor hyperparameters in netwerk voor classificatie van bloemkolen op basis van de oriëntatie.	50
5.1	Hyperparameters autoencoder.	61
5.2	Resultaten metrieken.	68

Lijst van afkortingen

AI	Artificiële intelligentie
AE	Autoencoder
ANN	Artificiële neurale netwerken
DL	Deep learning
CNN	Convolutioneel neuraal netwerk
CV	Computervisie
ExG	Excess Green Index
FN	False Negatives
FP	False Positives
GAP	Global Average Pooling
GNSS	Global Navigation Satellite System
ILVO	Instituut voor Landbouw-, Visserij- en Voedingsonderzoek
KNN	K-nearest-neighbour
ML	Machine learning
MLP	Multi-Layered Perceptron
MSE	Mean squared error
NDVI	Normalized Difference Vegetation Index
OBIA	Object-based image analysis
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Rood Groen Blauw
SGD	Stochastic gradient descent
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives
t-SNE	t-distributed Stochastic Neighbor Embedding
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle

1

Inleiding

Door de toenemende wereldbevolking stijgt de druk op voedselproductie en het milieu. De hoeveelheid grond geschikt voor voedselproductie daarentegen neemt niet zo sterk toe, waardoor er meer geproduceerd zal moeten worden op dezelfde oppervlakten, zonder dat de impact op het milieu toeneemt. In het proces van voedselproductie is er sprake van een grote verspilling (bv. gelijktijdig oogsten van al het fruit waardoor ook onrijpe vruchten geoogst worden), alsook van een sterke belasting op het milieu door gebruik van inputs zoals meststoffen en gewasbeschermingsmiddelen. Precisielandbouw probeert hier een oplossing aan te bieden: meer produceren met minder inputs om zo een efficiëntere productie te bekomen met minder impact op het milieu. Het doel van precisielandbouw is bijgevolg tweeledig: economisch en ecologisch [7, 9, 10].

Het Instituut voor Landbouw-, Visserij- en Voedingsonderzoek (ILVO), ontwikkelt en evalueert nieuwe mogelijkheden in het kader van precisielandbouw. Het is een internationaal bekend wetenschappelijk instituut van de Vlaamse overheid. Deze masterproef situeert zich op de afdeling agrotechniek, waar men focust op innovatieve en milieuvriendelijke landbouwsystemen. Dankzij de technologische revolutie afgelopen decennia, komt er immers een tijdperk aan waar ook in de landbouw slimme systemen op basis van artificiële intelligentie (AI) hun intrede doen.

Hiervoor wordt er voortdurend data verzameld, afkomstig van allerhande sensoren zoals bodemvochtsensoren, diverse camera's, pluviometers etc. Deze netwerken aan sensoren genereren een grote hoeveelheid data met een gemeenschappelijk doel: de efficiëntie van bepaalde technieken

in de landbouw optimaliseren.

Een belangrijke applicatie van artificiële intelligentie is computervisie (CV), waarbij gewerkt wordt met beeldmateriaal als input voor intelligente systemen, verzameld door diverse typische camera's. Enkele voorbeelden van toepassingen met computervisie voor precisielandbouw zijn [7]: classificatie van insecten, ziektedetectie bij planten, opkomstbepaling, plaats-specifiek onkruidbeheer ...

1.1 Doelstelling

Deze masterproef onderzoekt de mogelijkheden van van deep learning binnen de computervisie-taken in het domein van precisielandbouw. Het doel hierbij is enerzijds het ontwikkelen van softwarematige toepassingen in de precisielandbouw en anderzijds het onderzoeken van de mogelijkheid bepaalde bestaande computervisie-technieken te hercombineren tot een nieuwe efficiënte methode. Deze masterproef wordt opgedeeld in twee delen met elk hun eigen onderzoeksvragen:

1. Kan de classificatie van ritnaalden en de bepaling van de oriëntatie van bloemkolen op een geautomatiseerde manier gebeuren in plaats van manueel te classificeren?
2. In tal van huidige projecten wordt er nog veel tijd verloren met het labelen van de databases om efficiënte toepassingen te creëren. Is het mogelijk om met minder labeling alsnog zo'n toepassingen te creëren? Hoe kan dit gerealiseerd worden en wat is het effect op de performantie van het model? Wat zijn de gevolgen van een variërend aandeel gelabelde data?
3. Hoe kan er omgegaan worden met de enorme variatie van soorten gewassen en omgevingen bij de creatie van intelligente systemen binnen de precisielandbouw?

1.2 Opbouw

Dit werk begint in hoofdstuk 2 met een algemeen overzicht van de basisconcepten gaande van artificiële intelligentie tot computervisie. De literatuurstudie in hoofdstuk 3 geeft een overzicht van wat precisielandbouw is, aangevuld met diverse methodieken die gebruik maken van artificiële intelligentie in de precisielandbouw. toepassingen van artificiële intelligentie in de precisielandbouw, tot meer specifieke werken die aansluiten bij enkele van de deelvragen in deze masterproef.

In hoofdstuk 4 komt het eerste deel van de praktische realisaties in deze masterproef aan bod, behorende bij de eerste onderzoeksvraag. Er komen twee case studies aan bod: de herkenning

van ritnaalden in sectie 4.1 en classificatie van bloemkolen op basis van hun oriëntatie in sectie 4.2. Verder wordt in hoofdstuk 5 een proof of concept voorgesteld van een nieuwe aanpak, met oog op het detecteren van planten op overzichtsfoto's genomen met drones zonder de noodzaak aan een volledige labeling. Tot slot worden in hoofdstuk 6 besluiten geformuleerd over de diverse toepassingen van computervisie in de precisielandbouw die in dit werk aan bod kwamen, aangevuld met enkele interessante voorstellen voor een verdere uitbouw van de methode die in hoofdstuk 5 voorgesteld werd.

2

Computervisie

In dit hoofdstuk wordt een algemeen overzicht gegeven van enkele zaken met betrekking tot computervisie die verderop in dit werk aan bod zullen komen.

2.1 Van artificiële intelligentie tot computervisie

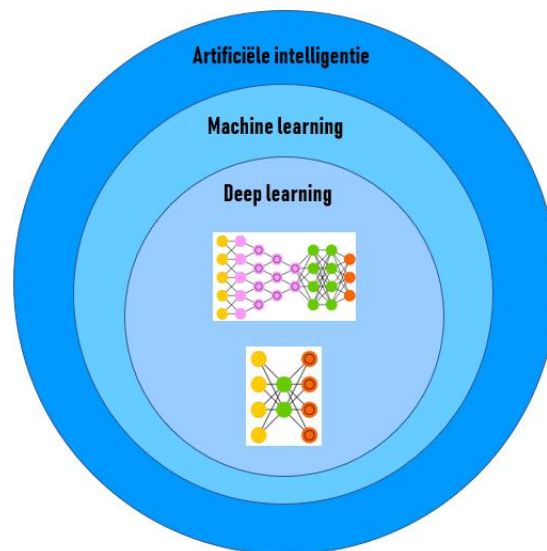
Reeds in de jaren '50 van de twintigste eeuw werd onderzoek gevoerd naar intelligente machines, waarbij de term 'artificiële intelligentie' ontstond. Op de vraag wat artificiële intelligentie (AI) eigenlijk is, werd door John McCarthy vervolgens geschreven: *"It is the science and engineering of making intelligent machines, especially intellegent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."* [11]. Kortom, AI is een wetenschap die zich bezighoudt met computerprogramma's of machines die denken en handelen op een intelligente, rationele manier [12].

Door de toename in rekencapaciteit en modelperformantie te laatste jaren stijgt het belang van AI in de onderzoekswereld, maar ook reeds in de bedrijfswereld. AI is dan ook niet meer weg te denken uit deze leefwereld en zit vervat in toepassingen zoals spraakassistentie, anti-spam software, zoekmachines, fraudedetectie, gezichtsherkenning, auto's uitgerust met lijnherkenning

... [13]

Eén van de meest gekende takken binnen de wetenschap van AI, betreft **machine learning (ML)**. Machine learning is het onderzoeksveld dat zich bezighoudt met algoritmen die data verzamelen, uit die data patronen probeert te herkennen en deze patronen vervolgens gebruikt om beslissingen te maken [6].

Waar AI probeert om intelligent gedrag te imiteren, gaat machine learning verder en focust op het leerproces. Bij machine learning probeert men te zoeken naar de sterktes en mogelijkheden van het programma of de computer om het complement te vormen aan menselijke intelligentie. Het programma of de computer tracht op die manier taken uit te voeren die buiten de mogelijkheden van de mens liggen [14, 15].



Figuur 2.1: Overzicht van artificiële intelligentie tot deep learning.

Ook machine learning kan opgesplitst worden op basis van het leermechanisme, waarbij **deep learning (DL)** een belangrijke subtak is. In figuur 2.1 wordt deze hiërarchische indeling weergegeven. Bij deep learning wordt er gebruik gemaakt van meer complexe modellen, waarbij de machine zichzelf algoritmen aanleert. Deze complexe modellen bestaan uit meerdere 'processing layers' [16].

Computervisie

Dankzij de sterke evolutie in de beschikbare hardware, evolueerde ook het onderzoek en gebruik van deep learning [17]. Hierdoor kende ook de **computervisie (CV)** de laatste jaren een spectaculaire vooruitgang. Computervisie kan gezien worden als een applicatie van AI waarbij men

gebruik maakt van verschillende technieken uit de machine learning (bv. K-means clustering, classifiers...). De laatste jaren werden aan computervisie ook technieken uit deep learning aan toegevoegd [11, 18].

In computervisie probeert het model informatie uit beelden te extraheren en transformeren om zo gewenste verbanden te leggen. Een machine ziet een afbeelding als een array van pixels in drie dimensies: de hoogte, de breedte, en de RGB(rood, groen, blauw)-waarden. Met behulp van artificiële neurale netwerken (ANN) gaat de machine vervolgens proberen om kenmerken uit de afbeelding te onttrekken om de gewenste verbanden te leggen. Dit proces bevat een zekere analogie met het menselijk biologisch systeem en de bijhorende (onbewuste) denkpatronen [19, 20].

Typische basisproblemen in de computervisie zijn [20]:

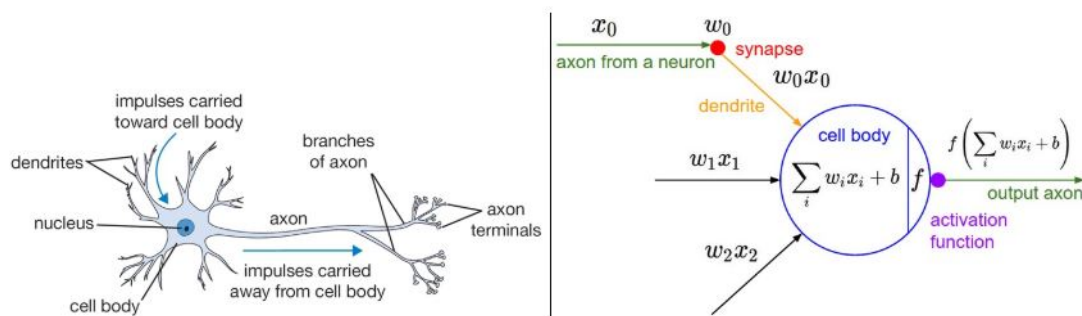
1. Herkenning van objecten: wat is het?
 - Classificatie: indeling in categorieën: wat?
 - Detectie: bepaling van het object: wat en waar?
 - Segmentatie: verdeling in verschillende gebieden/soorten: tot welke klasse behoort elke pixel?
2. Reconstructie van de (3D) omgeving: waar is het?
3. Bewegingsanalyse: waar gaat het naartoe?

2.2 Artificiële neurale netwerken

2.2.1 Opbouw van een neuraal netwerk

Het menselijke zenuwstelsel en brein liggen aan de basis van artificiële neurale netwerken. Een cel kan gezien worden als eenheid waaruit de mens is opgebouwd en kan verschillende functies hebben. Het menselijk brein bevat vele tientallen miljarden zenuwcellen of neuronen, verantwoordelijk voor het ontvangen en doorgeven van signalen. In figuur 2.2 staat aan de linkerzijde de grafische weergave van zo'n zenuwcel. Hierin wordt gevisualiseerd dat impulsen via dendriten binnenkomen in de celkern en het neuron verlaten via het axon wanneer ze een bepaalde drempel overschrijden. Dit axon vertakt en draagt het impuls vervolgens via de synapsen over aan de volgende neuron.

Naar analogie van dit model kan in 2.2 (rechts) de mathematische voorstelling gevonden worden, waarbij de impuls (x_0) uit een vorig neuron langsheen het dendriet (w_0) de cel binnenkomt



Figuur 2.2: Schematische weergave van menselijk neuron (links) en de mathematische voorstelling (rechts). Afbeelding overgenomen uit [1].

(w_0x_0) . De verschillende dendrieten hebben elk hun eigen gewicht: w_i . In de cel zelf komen verschillende impulsen (x_i -waarden) binnen, er ontstaat in de kern een gewogen som $\sum_i w_i x_i$.

In het meest eenvoudige geval spreekt men van een **perceptron** waarbij de output een binaire waarde heeft. De output bedraagt dus 0 of 1, afhankelijk of de gewogen som een bepaalde drempelwaarde bereikt of niet. Deze drempelwaarde kan gesimuleerd worden als bias (b), die ervoor zorgt dat de uiteindelijke gewogen som $(\sum_i w_i x_i + b)$ wordt. Deze bias is dus een maat voor hoe eenvoudig of moeilijk een perceptron zijn output 1 doorgeeft langsheen zijn axonen. Het idee is dat deze gewichten en biases in het trainingsproces bijgestuurd worden op basis van de error behorende bij de output om zo een gewenst resultaat te bekomen.

De drempel bepaalt dus of het neuron 'geactiveerd' wordt (1 als output). Dit kan wiskundig gezien worden als een stapfunctie, die ook wel de activatie functie f genoemd wordt. De wiskundige formule wordt weergegeven in 2.1. [1, 21, 22]

$$output = \begin{cases} 0 & \text{als } \sum_i w_i x_i + b \leq 0 \\ 1 & \text{als } \sum_i w_i x_i + b > 0 \end{cases} \quad (2.1)$$

Tegenwoordig wordt er meestal andere activatie functies gebruikt. Een moeilijkheid bij de activatiefunctie van een perceptron is dat zijn gewicht zelfs bij een kleine wijziging in gewicht of bias al kan flippen tussen 0 en 1, wat uiteraard het gedrag van het gehele netwerk wijzigt. Hierdoor is het lastig om de gewichten en biases gradueel te wijzigen om zo het gewenste gedrag te beïnvloeden. Beter zou zijn dat bij kleine wijzigingen in de gewichten of biases, er ook kleine wijzigingen in de outputs zijn. In een zoektocht naar nieuwe activatiefuncties met lagere sensitiviteit, zijn er tal van functies onderzocht met elk hun voor- en nadelen. Hieronder volgt een korte opsomming van de belangrijkste activatie functies die ook in deze masterproef verder gebruikt worden [1, 23, 24, 25, 26].

- **Sigmoid functie (logistic function):** Deze functie berekent een waarde tussen 0 en 1 uit een reëel getal volgens een niet-lineair verband (formule 2.2), om zo het hierboven aangehaalde probleem van het perceptron te vermijden. Het voordeel van dit verband is dat de afgeleide computationeel eenvoudig te berekenen is, wat bijdraagt aan een vlottere training van het netwerk. De training van het netwerk maakt gebruik van het backpropagation algoritme 2.2.2. Wegens recentelijk betere activatie functies wordt deze niet zo vaak meer gebruikt.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

- **Tangens hyperbolicus (tanh) functie:** Deze niet-lineaire functie zet een reëel getal om naar een waarde tussen -1 en 1 waardoor de output gecentreerd is rond het nulpunt. Tangens hyperbolicus kan uitgedrukt worden in functie van de sigmoid functie (zie formule 2.3). Doordat de uitvoerwaarden symmetrisch rond de oorsprong liggen, is de kans groter dat er waarden berekend worden die gemiddeld dichter bij 0 liggen. Dit maakt dat deze geschaalde functie verkozen wordt boven de sigmoid functie zelf [27].

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 = 2\sigma(2x) - 1 \quad (2.3)$$

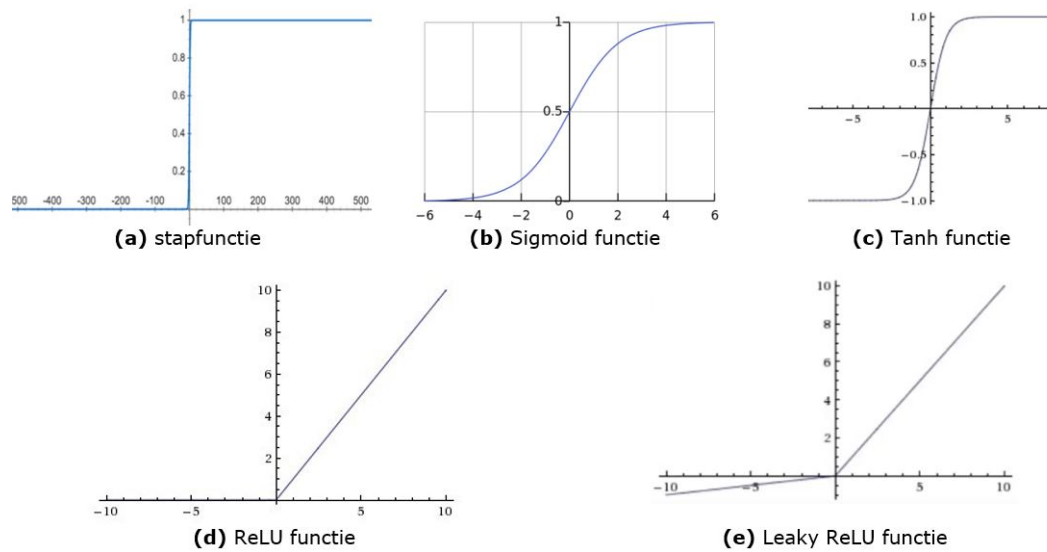
Beide functies hebben echter het nadeel dat ze satureren bij hoge inputwaarden op 0 of 1, waardoor de afgeleide naar 0 neigt en deze gedurende het ganse backpropagation algoritme meegenomen wordt waardoor de gewichten bijna niet geüpdatet worden. Het gevolg hiervan wordt het vanishing gradient probleem genoemd, waarbij het netwerk blijft hangen op een lokaal optimum omdat het niet meer verder geüpdatet wordt.

- **Rectified Linear Unit (ReLU):** Deze functie geeft een output voor positieve inputwaarden in de range $[0, \infty]$ volgens de vergelijking in 2.4. Een nadeel bij deze functie is dat alle negatieve inputwaarden onmiddellijk 0 worden, waardoor de trainingsmogelijkheden van het model (backpropagation algoritme) sterk beperkt worden: eens een neuron 0 is kan hij nooit nog geactiveerd worden. Een mogelijke oplossing hiervoor wordt in volgende activatie functie weergegeven.

$$f(x) = \max(0, x) \quad (2.4)$$

- **Leaky ReLU:** Het hierboven beschreven probleem bij de ReLU activatie functie, wordt bij de leaky variant vermeden door een kleine helling (α), bijvoorbeeld 0.01, in te voeren voor inputwaarden onder 0. Dit resulteert in een wiskundige functie beschreven in 2.5.

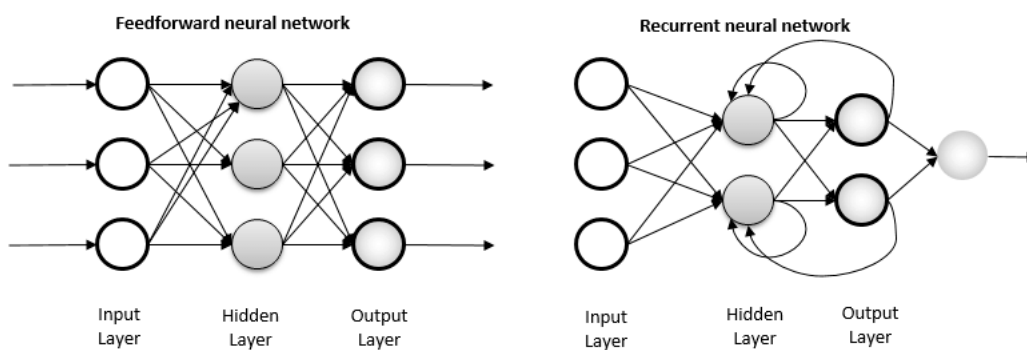
$$f(x) = \begin{cases} x & \text{als } x > 0 \\ \alpha x & \text{als } x \leq 0 \end{cases} \quad (2.5)$$



Figuur 2.3: Grafische weergave van mogelijke activatiefuncties.

Wanneer meerdere neuronen gecombineerd worden, ontstaat een neuraal netwerk. Die combinatie kan op verschillende manieren gebeuren wat leidt tot een bepaalde architectuur. In het algemeen kan de architectuur opgedeeld worden in twee soorten (zie ook figuur 2.4) [28]:

1. **Feed-forward netwerken:** alle verbindingen lopen in 1 richting zonder lussen, waardoor er geen intern geheugen mogelijk is.
2. **Recurrente netwerken:** lussen worden toegestaan waardoor een neuron zijn outputs eveneens kan gebruiken als inputs. Hierdoor ontstaat een afhankelijkheid van de vorige staat, wat impliceert dat er een korte termijn geheugen mogelijk is. Dit soort netwerken worden voornamelijk gebruikt voor tijdsreeksen.



Figuur 2.4: Feed-forward versus recurrent neuraal netwerk. Afbeelding overgenomen uit [2].

Een eenvoudig voorbeeld van de eerste soort is de Multilayer Perceptron (MLP). Zoals ook in afbeelding 2.4 aan de linkerkant getoond wordt, is de meest linkse laag de laag die de input ontvangt en de meest rechtse laag de output laag. Tussen deze lagen zitten verborgen lagen, hierin wordt de inkomende informatie verder getransformeerd met als doel als maar nuttigere kenmerken uit de data te extraheren. Alle neuronen zijn verbonden met alle neuronen uit de volgende laag, waardoor de feed-forward structuur ontstaat [29].

In deze masterproef komen enkel architecturen aan bod die behoren tot de eerste soort, deze worden verder toegelicht in volgende secties 2.2.2 en 2.2.3.

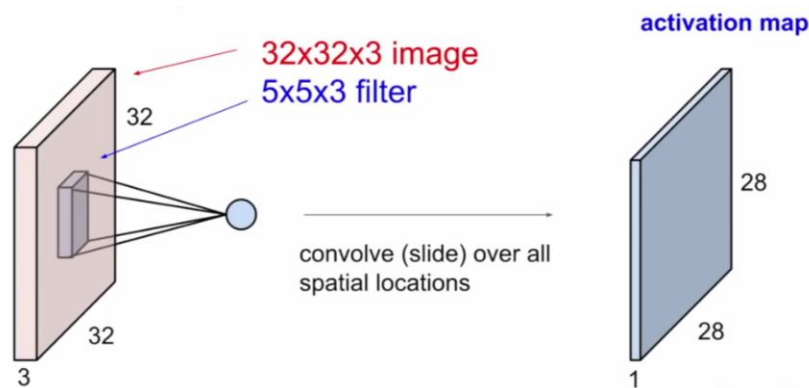
2.2.2 Convolutionele neurale netwerken

Wanneer de input van een neurale netwerk uit afbeeldingen bestaat, is een regulier netwerk zoals de MLP niet optimaal: afbeeldingen bestaan immers uit driedimensionale input (hoogte, breedte, diepte (RGB)) wat resulteert in enorme hoeveelheden input. Bovendien zit informatie in een afbeelding in 2D gespreid, wanneer dit met een lineair netwerk behandeld wordt, wordt het moeilijk om hiermee rekening te houden. Als een neuron een bepaald stuk informatie bevat, zal de omgeving rondom deze informatie namelijk ook vaak relevante informatie bevatten. Het is bijgevolg niet efficiënt om alle neuronen met elkaar te verbinden op een lineaire manier. Bovendien komen in afbeeldingen heel vaak dezelfde kenmerken terug (bv. randen, kleurovergangen ...) die dus hergebruikt kunnen worden. Een convolutioneel neurale netwerk (CNN) is hier wel voor geschikt.

Convolutionele laag

De convolutionele laag is de basislaag in het CNN. In deze laag zijn de neuronen niet verbonden met alle voorgaande neuronen of met de volledige afbeelding, maar wel met een lokaal gedeelte: het 'receptive field'. De grootte van dat veld wordt bepaald door de grootte van de bijhorende filter. Deze filter is een reeks van getallen, die telkens een bepaald gedeelte van het vorig oppervlak gaat gebruiken om de berekening mee uit te voeren. Om te vermijden dat er teveel parameters zouden zijn indien elk neuron een filter met getallen heeft, wordt eenzelfde filter gebruikt voor alle neuronen op eenzelfde diepte. Op die manier gaat de filter eigenlijk over de volledige input heen schuiven.

Belangrijk hierbij is dat de filter wel steeds de volledige diepte van de input meeneemt. Bij een RGB afbeelding zit de data vervat in drie dimensies (rode, groene en blauwe), dus zal de input diepte drie hebben. De verschillende dimensies worden bijgevolg asymmetrisch behandeld: breedte-hoogte wordt gedekt door een verschuivende filter met beperkte afmetingen (= lokaal), terwijl de diepte steeds volledig gedekt is door een filter van dezelfde diepte.

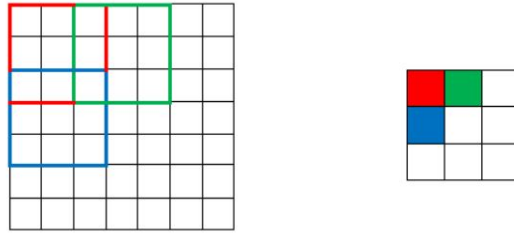


Figuur 2.5: Convolutionele laag: $32 \times 32 \times 3$ inputafbeelding met $5 \times 5 \times 3$ filter resulteert in feature-map van $28 \times 28 \times 1$ door convolutie. Afbeelding overgenomen uit [3].

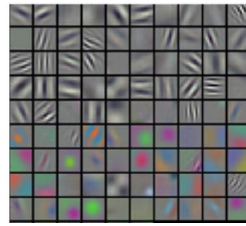
Met elementsgewijze vermenigvuldiging worden de waarden van de input (bv. de pixels van een afbeelding) telkens vermenigvuldigd met de waarden in de filter (= gewichten). In figuur 2.5 wordt bijvoorbeeld een filter van $5 \times 5 \times 3$ gebruikt over een afbeelding van $32 \times 32 \times 3$, waardoor er 75 vermenigvuldigingen zullen plaatsvinden. Vervolgens worden deze resultaten opgeteld waarbij een getal bekomen wordt die die bepaalde positie van de filter voorstelt, en dit gebeurt voor elke locatie. In het geval van figuur 2.5 geeft dit dus een array van $28 \times 28 \times 1$, vermits er $32 - 5 + 1 = 28$ mogelijke locaties voor de filter zijn. Deze array wordt de 'activation map' of ook 'feature map' genoemd. Deze matrixberekeningen voeren de wiskundige bewerking 'convolutie' uit, wat ook de naam van deze laag verklaart [1, 5, 26].

Bij dit hele systeem van convolutie komen enkele belangrijke hyperparameters aan bod die de ruimtelijke ordening van de outputlaag bepalen [1, 4, 26]:

1. **Filtergrootte (F)**: zoals hierboven reeds vermeld bepaalt dit het receptive field.
2. **Stride (S)**: deze waarde bepaalt de hoeveelheid pixels die de filter in één keer verschuift. In figuur 2.6 schuift de filter, in tegenstelling tot bij het vorige voorbeeld, op met twee pixels per keer en dus $S=2$.
3. **Padding (P)**: om de ruimtelijke indeling nog meer te kunnen bijsturen, kan er padding toegevoegd worden. Deze waarde geeft aan hoeveel rijen en kolommen met nullen er rondom de afbeelding worden toegevoegd. In bovenstaand voorbeeld was dit nul.
4. **Aantal filters (K)**: er kunnen meerdere filters toegevoegd worden die door eenzelfde groep neuronen gebruikt worden. Door het gebruik van meerdere filters, ontstaan er meerdere feature maps in de outputlaag. Deze parameter heeft dus een directe invloed op de diepte van de outputlaag. De naam feature map kan dan ook verklaard worden door het feit dat de output behorende bij een bepaalde filter naar een bepaald kenmerk of feature gaat



Figuur 2.6: Visuele voorstelling van stride $S=2$. Afbeelding overgenomen uit [4].



Figuur 2.7: Visuele weergave van filters. Afbeelding overgenomen uit [5].

kijken. Het gaat hierbij om kenmerken zoals een bepaalde kromming, kleur, blobs, edges etc. Dit wordt geïllustreerd in figuur 2.7.

De uiteindelijke grootte van de outputlaag ($W_2 \times H_2 \times D_2$), bij input van ($W_1 \times H_1 \times D_1$) kan met formules 2.6-2.8 bepaald worden [1].

$$W_2 = \frac{W_1 - F + 2P}{S} + 1 \quad (2.6)$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1 \quad (2.7)$$

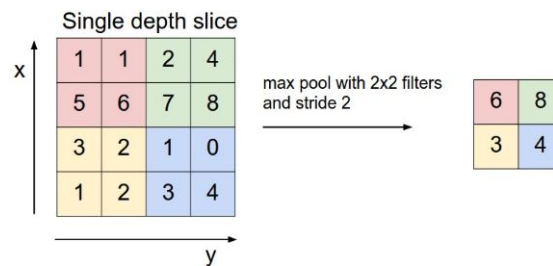
$$D_2 = K \quad (2.8)$$

Na zo'n convolutionele laag, wordt meestal gebruik gemaakt van een ReLU-laag, die zoals eerder vermeld een activatie functie uitvoert. Dit zorgt er ook voor dat het systeem, na het berekenen van lineaire convoluties niet-lineair wordt [4, 26].

Verder zijn er nog enkele andere lagen die regelmatig aan bod komen in een CNN. Enkele typische andere lagen in een CNN die in deze masterproef regelmatig aan bod komen, worden hieronder kort geschetst.

Max-pooling laag

Met behulp van een pooling laag, in dit geval max-pooling, wordt een downsample operatie uitgevoerd om de hoeveelheid parameters en berekeningen in het netwerk te verkleinen. Het verlies van informatie wordt verantwoord door het idee dat de exacte locatie van een eigenschap minder belangrijk is dan de eigenschap en zijn benaderde locatie. Het voordeel van een lager aantal parameters is dat overfitting beter controleerbaar is. Overfitting betekent immers dat het netwerk niet kan veralgemenen omdat het te sterk afgetraind is op getrainde data, waardoor voorspellingen over nieuwe data moeilijk zijn.



Figuur 2.8: Voorbeeld van pooling met filter 2x2 en stride 2. Afbeelding overgenomen uit [1]

Zo'n pooling laag heeft ook een filter (meestal 2x2) en een bijhorende stride, waarbij de input volgens de grootte van de stride gedownsamplend wordt in breedte en hoogte (niet qua diepte). In het geval van max-pooling gebeurt dit door telkens het maximum te nemen van de getallen die in het receptive field van de filter horen, zoals in figuur 2.8 getoond wordt. Het receptive field wordt bijgevolg vergroot [3, 4, 26].

Fully connected laag

Een fully connected laag komt meestal voor op het einde van een CNN. In deze laag worden alle neuronen volledig, paarsgewijs verbonden met de neuronen van de vorige laag, maar niet binnen eenzelfde laag.

Training van het netwerk

Na de opbouw van het netwerk uit verschillende lagen, volgt de trainingsfase. In deze fase krijgt het netwerk een hele hoop trainingsdata als input, waarbij het netwerk probeert te leren naar welke eigenschappen de filters moeten zoeken. Hiervoor moeten de gewichten dus telkens aangepast worden. Vooraf worden deze random geïnitieerd en tijdens de trainingsfase gebeurt die aanpassing via **back-propagation**. Dit gebeurt in drie fases, die uitgevoerd worden in één

trainingsfase: een epoch. Het aantal epochs is een parameter die ingesteld kan worden en varieert naargelang de toepassing [25].

1. **Feed-forward:** de input wordt doorheen het netwerk gestuurd, waarbij telkens de output van elk neuron wordt berekend tot in de laatste laag.
2. **Error gradient back-propagation algoritme:** in deze stap gaat het netwerk het resultaat uit de vorige fase gaan vergelijken met het beoogde resultaat (bijvoorbeeld door een label die aan de trainingsdata gegeven werd). Hiervoor wordt er gebruik gemaakt van een **verliesfunctie** die een maat is voor de inconsistentie tussen de beide resultaten. In het algoritme wordt er dan telkens de helling (gradient) van de error-functie bepaald op een recursieve manier. Dit gebeurt voor elk neuron, gaande van de laatste laag tot de eerste laag (back-propagation), telkens afhankelijk van de neuronen waarmee ze verbonden zijn.
3. **Update gewichten en bias:** op basis van de berekende error gradient worden de gewichten en bias van elke neuron bijgestuurd, gaande van de eerste naar de laatste laag. Dit is dus opnieuw een voorwaartse doorgang.

Een belangrijke hyperparameter die hierbij aan bod komt is de **learning rate**. Een hoge waarde hiervoor betekent dat er grotere stappen worden genomen bij de update van de gewichten, waardoor het model minder tijd nodig heeft om naar optimale gewichten te evolueren. Echter kan een te hoge waarde er ook voor zorgen dat de stappen te groot zijn, waardoor het optimum niet bereikt kan worden. Bij een te kleine learning rate daarentegen, is er het gevaar dat het netwerk blijft hangen in een lokaal optimum, omdat het niet in staat is om eruit te springen. Kortom, deze parameter moet met trial and error ingesteld worden tijdens de trainingsfase [25, 30].

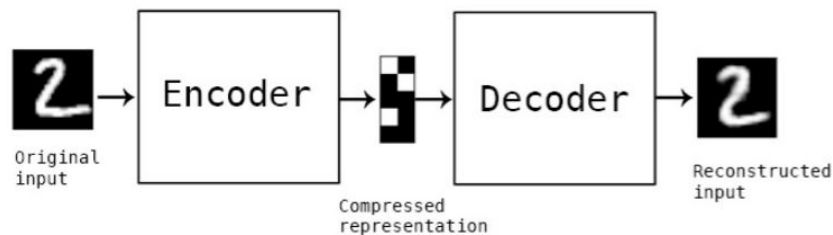
Testen van het netwerk

Nadat het netwerk getraind werd, dient gekeken te worden hoe goed het netwerk eigenlijk is. Hiervoor wordt er gebruik gemaakt van een hoeveelheid testdata die het netwerk nog niet gezien heeft. De beoordeling van hoe goed het netwerk is, hangt af van de toepassing en varieert ook doorheen deze masterproef.

2.2.3 Autoencoder (AE)

Een andere typische netwerkarchitectuur die in deze masterproef aan bod komt, is een autoencoder. Deze architectuur bestaat uit twee delen: een encoder en decoder. De originele input

x (hoog dimensionele vector) wordt door de encoder gecomprimeerd (door een bottleneck gestuurd met lage dimensie) tot een **latente representatie**, om vervolgens langsheen de decoder terug gereconstrueerd te worden. Een autoencoder probeert bijgevolg om een functie $h(x) \approx x$, de identiteitsfunctie, te bepalen [6, 31]. Dit wordt geïllustreerd in figuur 2.9. Er kan in de encoder-decoder structuur ook gebruik gemaakt worden van convolutionele lagen, waardoor de autoencoder ook een CNN wordt.



Figuur 2.9: Werking van een autoencoder. Afbeelding overgenomen uit [6].

Autoencoders kunnen gebruikt worden voor verscheidene toepassingen zoals classificatie, clustering, segmentatie en compressie. Een belangrijk voordeel is bovendien dat ze ook unsupervised kunnen leren, zie sectie 2.3. Een mogelijke toepassing is de pre-training van een neurale netwerk: het neurale netwerk wordt als encoder gebruikt en de gespiegelde versie als decoder. In deze eerste fase gaat de autoencoder pogen om bepaalde patronen uit de ongelabelde data te halen. Een tweede fase is dan het fine-tunen van het originele netwerk, met e gewichten van de autoencoder als startgewichten en supervised leren [6, 32].

2.3 Soorten leren

Leren houdt in dat een netwerk probeert om onderliggende verbanden te vinden tussen de verschillende data die tijdens het trainingsproces aan bod komt. Door de grote variatie in natuurlijke data, is het immers niet mogelijk om alle mogelijke situaties vooraf te kennen. Het model moet bijgevolg leren veralgemenen, leren om een verband te vinden om op die manier adequaat op nieuwe data te kunnen reageren. Concreet betekent dit dat de gewichten in het netwerk tijdens de training moeten convergeren naar een optimum die het onderliggend verband van de inputdata en labels het beste weergeeft. Het netwerk leert dus welke gewichten optimaal zijn in functie van de gegeven inputdata [6, 33].

In het gehele trainingsproces van neurale netwerken staat de manier waarop geleerd wordt dan ook centraal. Een belangrijke parameter hierbij betreft de **feedback** die beschikbaar is tijdens het leerproces.

2.3.1 Supervised vs. unsupervised learning

Bij **unsupervised leren** is er geen sprake van feedback en gaat het model proberen om patronen in de input te detecteren. Een typische toepassing van dit unsupervised leren is het detecteren van potentiële clusters. Dit unsupervised leren komt niet aan bod in deze masterproef.

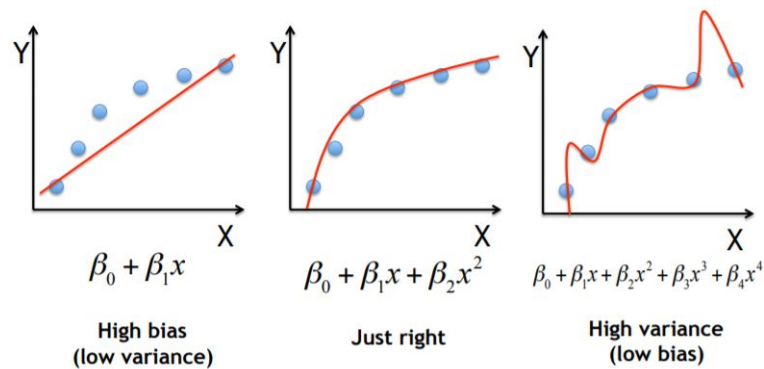
Bij **supervised leren** daarentegen is er net wel feedback voorzien. Hierbij komt data voor onder de vorm van input-output paren $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$. Er is dus een trainingsset met n voorbeelden, waarbij elke y -waarde eigenlijk gezien kan worden als bepaald door een onbekende functie $y = f(x)$. Op basis van deze gelabelde inputdata moet het model proberen een functie h (hypothese functie) te bepalen die de functie f zo goed mogelijk benaderd.

Vervolgens zal de correctheid van de hypothese functie gemeten worden met behulp van een andere hoeveelheid gelabelde inputdata: de testset. Wanneer $h(x) = y$, betekent dit dat de hypothese functie goed veralgemeend op nieuwe voorbeelden.

Een nadeel bij supervised leren is de noodzaak aan gelabelde data, een zogenaamde **ground truth**. Van deze ground truth wordt verondersteld dat de labeling correct gebeurde, en de bepaling van de hypothese functie is bijgevolg volledig hiervan afhankelijk. Hoe meer fouten (=ruis) in de labeling van de inputset, hoe minder accuraat het model zal zijn.

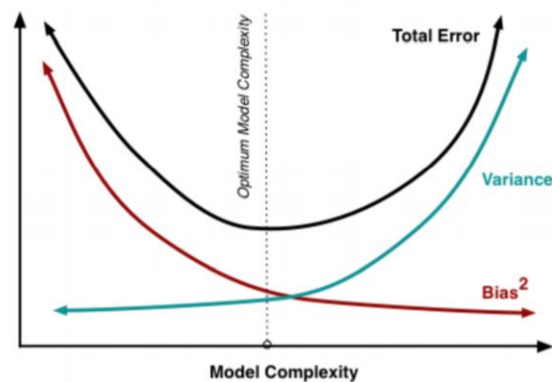
Het is echter niet alleen belangrijk dat de data correct gelabeld is, maar ook dat er een voldoende grote hoeveelheid gelabelde data beschikbaar is en dat deze data ook een zekere uniforme spreiding kent (bv. bij een classificatieprobleem een gelijkaardig aantal afbeeldingen behorende tot elke klasse). In het geval dat er slechts een beperkte hoeveelheid gelabelde data beschikbaar is, bestaat het gevaar dat het model de trainingsdata kan memoriseren, zonder hier een achterliggende structuur in te zoeken. Het gevolg is dat het model zal overfitten en slechte resultaten geeft met de testset. De inputdata heeft bijgevolg ook een invloed op de complexiteit van het model. Hoe complexer het model, hoe ingewikkelder de gezochte functie kan worden, maar ook hoe meer data nodig is om overfitting te vermijden [6, 28, 33, 34, 35].

Een belangrijk concept dat hiermee gepaard gaat, is de **bias-variance trade-off**, die weergeeft hoe goed het model kan veralgemenen. Bij deze trade-off staat de bias voor een constante error. De variantie geeft aan hoe sterk de error varieert tussen de verschillende datasets. In het geval van overfitting is er sprake van een hoge variantie en eerder lage bias, de variantie kan in dit geval gereduceerd worden door bijvoorbeeld meer trainingsdata te voorzien. Naast overfitting kan ook underfitting optreden, waarbij de bias net een hoge waarde aanneemt, maar de variantie laag is. In dat geval is het model dus niet in staat de trainingsset te vatten om hieruit een goede hypothese functie te bepalen, die bias moet gereduceerd worden door bijvoorbeeld langer te trainen of de complexiteit van het netwerk te verhogen. In figuur 2.10 wordt de underfitting links weergegeven, terwijl overfitting rechts staat. Het blijft echter een trade-off, vermits bij



Figuur 2.10: Bias-variance voorbeelden: rechts underfitting, links overfitting. Afbeelding overgenomen uit [6].

daling van het ene het andere vaak stijgt, dit wordt geïllustreerd in figuur 2.11 [6, 30].



Figuur 2.11: Grafische weergave bias-variance trade-off. Afbeelding overgenomen uit [6].

Typische toepassingen van dit supervised leren zijn classificatie, waarbij de y-waarden een eindige set van mogelijke labels zijn, en regressie, waarbij y een continue (numerieke) output voorstelt.

2.3.2 Semi-supervised learning

In de praktijk is het niet altijd evident om voldoende gelabelde data te hebben en volstaat het clusteren via unsupervised learning niet altijd voor de gewenste toepassingen. Zodoende ontstond er een mengvorm: **semi-supervised leren**. Bij semi-supervised leren wordt gebruik gemaakt van zowel een grote hoeveelheid ongelabelde data, als een kleiner aantal gelabelde voorbeelden. Het model gaat proberen om bepaalde patronen te zoeken in de de ongelabelde data, om vervolgens de gelabelde te gebruiken om het geleerde te optimaliseren. [6, 33, 36]

3

Literatuurstudie

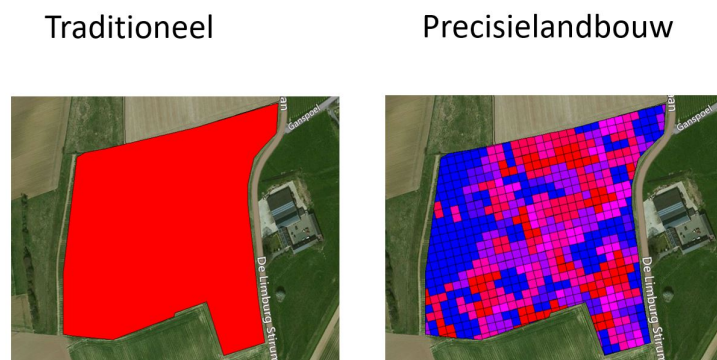
3.1 Precisielandbouw

Wanneer de vraag gesteld wordt wat precisielandbouw eigenlijk is, zijn er in de literatuur diverse definities terug te vinden. Enkele van deze definities: "**Precisielandbouw is ...**

- een vorm van landbouw die zich realiseert dat er **verschillen** zijn binnen percelen, en die probeert elke te onderscheiden eenheid grond op het **juiste tijdstip** de **optimale behandeling** te geven, gebruikmakend van geavanceerde **technologie**." (*D. Goense, 1988*) [7, 37]
- het telen van gewassen waarbij rekening wordt gehouden met de **specifieke behoeften** van het gewas, de weersomstandigheden en bodem op een **specifieke locatie**." (*W. Nugteren, 2009*) [7, 37]
- een vorm van landbouw waarbij teeltmaatregelen worden afgestemd op de **variatie in ruimte en tijd** van gewas- en bodemcondities binnen percelen. Het doel is een hoge productie van hoge kwaliteit te realiseren binnen vastgestelde grenswaarden voor milieu-indicatoren. Het maakt gebruik van **geavanceerde apparatuur en informatietechnologie**." (*J. Bouma, 1996*) [7, 37]

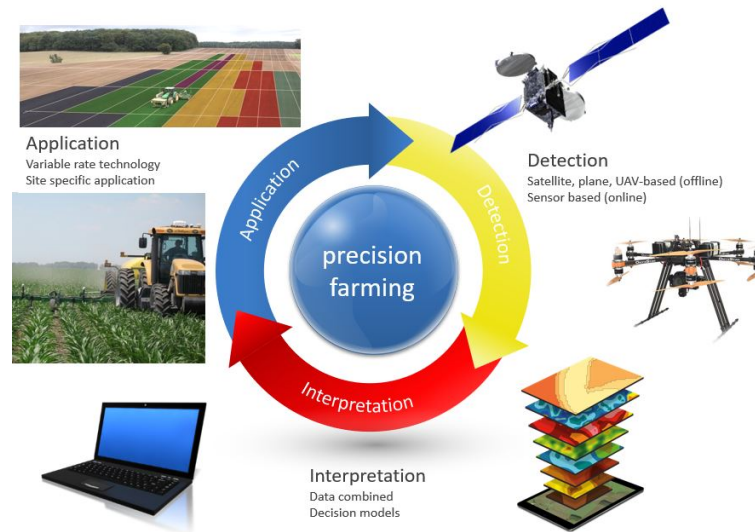
- defined as information and technology based farm management system to identify, analyze and manage variability within fields for optimum profitability, sustainability and protection of land resource.” (S. Banu, 2015) [10]

Samengevat kan gesteld worden dat precisielandbouw is: het op het **juiste moment** op de **juiste plek** toepassen van de **juiste teelmaatregel**, waarbij beslissingen genomen worden op basis van feiten (data) in plaats van op het gevoel. Het grote verschil met de klassieke landbouw is dat er niet per veld (grote rode oppervlakte in figuur 3.1), maar wel op kleinere schaal behoeftes (diverse kleine vakjes met de verschillende behoeften in verschillende kleuren in figuur 3.1) bepaald worden en hierop ingespeeld wordt, zoals visueel voorgesteld wordt in figuur 3.1 [7, 37, 38]. Precisielandbouw omvat meer dan enkel de plantaardige productiesystemen, ook dierlijke productie, visserij en bosbouw maken er deel van uit. In deze masterproef wordt gefocust op 'precision crop farming' en bijgevolg op alles wat te maken heeft met gewasproductie.



Figuur 3.1: Traditionele landbouw vs. precisielandbouw. Afbeelding overgenomen uit [7].

Precisielandbouw kan voorgesteld worden als een cyclus van detectie, interpretatie en applicatie zoals geïllustreerd in figuur 3.2. Enerzijds is er detectie of dataverzameling nodig om vervolgens bepaalde zaken te kunnen interpreteren en applicaties rond te bouwen. Anderzijds kan een bepaalde interpretatie of combinatie ook leiden naar interessante gebieden die vervolgens gedetecteerd kunnen worden [7]. Deze masterproef kadert in de fase van interpretatie: het combineren en analyseren van data om tot bepaalde beslissingen te komen.



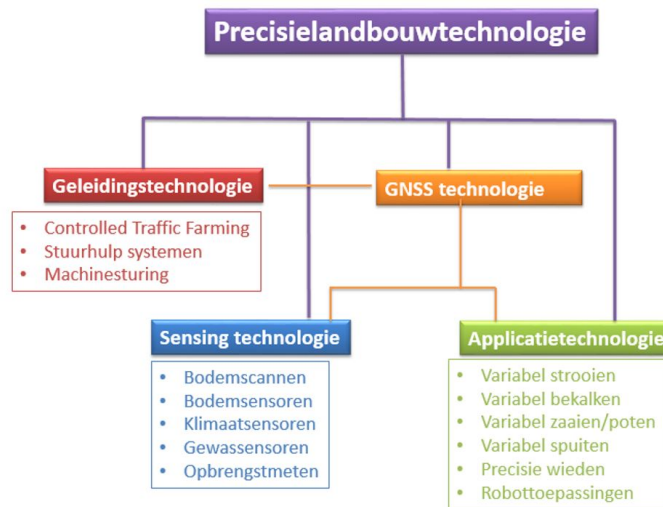
Figuur 3.2: Cyclus van precisielandbouw. Afbeelding overgenomen uit [7].

De mogelijkheden van precisielandbouw werden de laatste jaren zeer sterk uitgebreid dankzij een sterke evolutie in technologische mogelijkheden. Precisielandbouw omvat verschillende types technologieën die in onderlinge wisselwerking staan met elkaar zoals geïllustreerd in figuur 3.3. Het ontstaan van de precisielandbouw situeert zich reeds in de jaren '70 van de vorige eeuw dankzij het ontstaan van plaatsbepalingstechnologie met behulp van satellieten: Global Navigation Satellite System (GNSS) technologie.

Uit deze GNSS technologie ontstond de geleidingstechnologie, waarbij gefocust wordt op automatiseren van voertuigbesturing. Daarnaast is er nood aan sensortechnologie voor de collectie van data, in vele gevallen vanop afstand (remote sensing). Hiervoor zijn er heel wat verschillende types sensoren, zoals verschillende types camera's, bodemsensoren en klimaatsensoren beschikbaar, die in diverse toepassingen worden ingezet [7, 39].

Door deze evolutie in plaatsbepaling en sensoren wordt het mogelijk om de variabiliteit in kaart te brengen, waardoor er via beslissingsmodellen kleinere zones met bepaalde eigenschappen gevormd kunnen worden. Vervolgens kan de toepassingstechnologie hiervan gebruik maken om in de behoeften van elke zone afzonderlijk te voorzien. Een typische toepassing is de variable-rate technologie waarbij een product zoals mest of gewasbeschermingsmiddelen variabel gedoseerd wordt in het veld. De dosis wordt bijgevolg aangepast aan de plaats specifieke noden, bepaald door bovenstaande technologieën. [7, 10, 38].

In deze masterproef wordt onderzoek gedaan naar een mogelijk beslissingsmodel dat op basis van RGB afbeeldingen bekomen wordt met behulp van drones om de variabiliteit van gewassen in een veld in kaart te brengen.



Figuur 3.3: Technologie in precisielandbouw. Afbeelding overgenomen uit [7].

De schaal waarop de noden worden bepaald, varieert in precisie en kende ook reeds een evolutie. In precisielandbouw 1.0 werd gefocust op een nauwkeurige en uniforme behandeling van een perceel, met als typische toepassing stuurhulp bij tractoren. Precisielandbouw 2.0 gaat een stapje nauwkeuriger en werkt op gridlevel binnen een veld. Tenslotte is er ook precisielandbouw 3.0 waarbij de behandeling plaatsvindt op plantniveau, evoluerend naar bladniveau, om een behandeling per plant mogelijk te maken bijvoorbeeld met behulp van robotica.

De opkomst van artificiële intelligentie afgelopen jaren trok ook in de precisielandbouw de aandacht en zorgde voor mogelijkheden tot verdere automatisatie en optimalisatie van systemen. Deze evolutie komt in volgende sectie aan bod.

De laatste jaren nam het milieubewustzijn in de samenleving sterk toe, waardoor onderzoek naar precisielandbouw aan populariteit won. Om de nieuw beschikbare technologieën echter tot bij de kleine landbouwer te brengen zijn nog heel wat obstakels te overbruggen. In het algemeen is de kennis van de landbouwer over de mogelijkheden van precisielandbouw soms te beperkt wat ervoor zorgt dat, samen met de hoge kostprijzen, vele landbouwers nog niet investeren in deze mogelijkheden. Het is belangrijk om een goede sensibilisering te voorzien, zodanig dat toekomstige gebruikers het potentieel ervan leren kennen en op lange termijn de voordelen inzien: mogelijkheid om de oogst te optimaliseren met een duurzamere aanpak [7, 10, 40].

Daarbij aansluitend merkt Banu [10] op dat precisielandbouw in veel ontwikkelingslanden nog slechts een concept is. Ondersteuning, zowel strategisch als financieel, uit publieke en private sectoren is essentieel om ook deze landen mee te nemen in het verhaal van precisielandbouw. Er is immers veel potentieel om de landbouw in deze landen nog te optimaliseren, met positieve effecten op wereldniveau.

3.2 Computervisie in de precisielandbouw

De laatste jaren verschijnt er steeds vaker literatuur die het potentieel van artificiële intelligentie en meer specifiek computervisie in de snel evoluerende precisielandbouw aantoont. Het spectrum waarin AI binnen de precisielandbouw kan worden ingezet is zeer groot. Eerder onderzoek maakte voornamelijk gebruik van machine learning technieken, maar met de opkomst van deep learning ontstonden heel wat nieuwe mogelijkheden met veel ruimte voor onderzoek [9].

De mogelijkheden qua toepassingen waarin computervisie-technieken voor precisielandbouw kunnen worden ingezet zijn zeer divers. Enkele voorbeelden zijn plukrobots, ziekte detectie, voorspelling van de opkomst, droogte-stress bepaling, plant herkenning, insecten detectie in gewassen, classificaties van bepaalde types gewassen ...

Een andere toepassing die de laatste jaren veel aandacht kreeg, is het detecteren van planten en onkruid. Het detecteren van onkruid kan er namelijk voor zorgen dat de hoeveelheid gebruikte herbiciden (inputs) met een geautomatiseerd systeem verlaagd wordt, zonder de kwaliteit van de gewassen omlaag te halen. Het gebruik van herbiciden heeft immers negatieve gevolgen op het bestaande ecosysteem en zijn biodiversiteit. De zwakste schakel in de keten van zo'n geautomatiseerd systeem is echter de detectie en identificatie van het onkruid in de wijde range van omgevingscondities, inclusief de spectrale gelijkenissen tussen onkruid en planten [41, 42]. Een beschrijving van de problematiek die deze toepassing moeilijk maakt wordt gegeven in de volgende subsectie. Huidig onderzoek naar onkruiddetectie bevindt zich in een gevorderd stadium van detectie tussen rijen en semi-automatische methoden die gebruik maken van machine learning [36].

Het vervolg van deze sectie overloopt enkele methoden van computervisie in de precisielandbouw (met focus op onkruiddetectie). Er kan worden vastgesteld dat deze methoden in het algemeen bestaan uit vier fasen: dataverzameling, data (pre)-processing, training van modellen en het evalueren van het systeem. In de literatuur zijn er binnen deze fasen heel wat varianten te vinden, die hieronder verder toegelicht worden. De werken die hierbij centraal staan worden op het einde samengevat weergegeven in tabellen 3.2 en 3.3.

3.2.1 Algemene problematiek: variabiliteit

Het gebruik van computervisie in de precisielandbouw, meer specifiek in precision crop farming, is een uitdagende taak omwille van de enorme variatie in de natuur. Alleen al in de gewasproductie zijn er bijvoorbeeld ontelbaar veel plantensoorten. Plantensoorten onderling verschillen in meer of mindere mate, maar ook binnen een bepaalde soort zijn sterke variaties aanwezig, visuele variaties van vorm, kleur en grootte in allerlei mogelijke combinaties.

Bovendien doorloopt een plant diverse groeistadia. Een systeem geoptimaliseerd voor een vol-groeide plant, werkt daarom niet in de beginstadia van de groei. In de beginfase is de spectrale gelijkheid tussen diverse gewassen bijvoorbeeld heel groot [40]. Daarnaast is er ook een sterke variatie in de teeltomstandigheden (bodemsoorten, geografische regio, weersomstandigheden, aanwezigheid van insecten ...) Veel van de methoden in de literatuur zijn (nog) niet geschikt voor dit soort 'real world' condities [41].

Deze grote variatie vormt een obstakel om algemene systemen te ontwikkelen. Kortom, de ontwikkeling van intelligente en robuuste systemen in is een uitdagend en continu evoluerend onderzoeksgebied. Er worden bijgevolg diverse methoden voorgesteld in de literatuur voor verschillende toepassingen met elk hun sterktes en zwaktes.

Tenslotte kan opgemerkt worden dat ook de integratie van deze intelligente systemen in landbouwmachines en de overtuiging van het gebruik ervan, zie hierboven, ook een moeilijkheid is die in de toekomst overwonnen moet worden [15].

3.2.2 Dataverzameling

Unmanned Ground Vehicle (UGV) vs. Unmanned Aerial Vehicle (UAV)

In de diverse werken met computervisie in de precisielandbouw, kunnen twee categorieën onderscheiden worden wat betreft de manier waarop beelden verzameld worden. Daar waar Lottes et al. (2016,2018) [42, 43], Potena et al. [44], Milioto et al. [45] en Fawakherji et al. [46] gebruik maken van afbeeldingen genomen met robots (UGV) die in het veld opereren, wordt bij Hung et al. [47], Lottes et al [48], Milioto et al. [49] en Bah et al. [50] gebruik gemaakt van drones (UAV).

Zoals Lottes et al. (2017) [48] aanhalen, is het mogelijk om grotere gebieden vast te leggen op foto met behulp van UAV's. Tijdens de dataverwerking kan gebruik gemaakt worden van ruimtelijke relatie tussen planten bijvoorbeeld rijdetectie, vermits er een grotere hoeveelheid gewassen in één beeld gebracht kan worden. Bah et al. (2018) maken gebruik van rijdetectie in hun aanpak. Bovendien kan er ook informatie omtrent hoogtes worden verzameld voor de opbouw van hoogtemodellen, de foto's kunnen immers vanop diverse hoogten top-down genomen worden. Een nadeel is echter dat het gebruik van drones in alledaagse praktijk nog niet realistisch is: er is nog te weinig praktische en technische kennis, de reglementering hieromtrent is heel strikt en de infrastructuur hiervoor is beperkt. De evolutie naar een processing pipeline waarbij drones ingezet worden, moet in de toekomst verder uitgebouwd worden [36].

Daartegenover kan een UGV worden ingezet in een realtime end-to-end toepassingen: naast de camera voor het nemen van afbeeldingen en bijhorende detectiesysteem, kan de robot immers

uitgerust zijn met een selectief sproeisysteem of mechanische verwijdering. In dit geval zit de volledige pipeline van onkruiddetectie (onboard data-analyse) tot behandeling geïmplementeerd in de robot zodat realtime verwerking mogelijk wordt. Dit is bijvoorbeeld het geval bij Fawakherji et al. [46].

In dit werk gebruikt de voorgestelde methode afbeeldingen die gemaakt werden met behulp van drones vermits de methode gebruik maakt van overfitting op het totaalbeeld.

Sensoren voor diverse spectra

Naast de verschillende mogelijkheden om data te verzamelen, kan er ook een belangrijk onderscheid gemaakt worden wat betreft de spectra die de camera kan capteren. In veel werken [43, 45, 46, 47, 50, 51, 52] wordt gebruik gemaakt van het zichtbaar lichtspectrum en dus een RGB-camera. Deze camera's hebben als voordeel dat ze een goede prijs-kwaliteit verhouding hebben op basis van spatiële resolutie. Het nadeel van deze camera's is echter dat de spectrale resolutie (het aantal beschikbare banden) beperkt is [36].

Naast deze RGB-camera's kan er echter ook gebruik gemaakt worden van sensoren met meer spectrale banden: multi-spectraal of hyperspectraal. Bij multi-spectrale camera's wordt het RGB spectrum uitgebreid met de nabij-infrarood (NIR) band [36]. De NIR-band bevat extra spectrale informatie: de aanwezigheid van chlorofyl in planten zorgt voor de absorptie van een groot deel van het zichtbaar licht (energie nodig voor fotosynthese), terwijl het nabij-infrarood licht teruggekaatst wordt wat in afbeeldingen in de NIR-band zorgt voor een oplichting van de planten. Dit heeft als belangrijk voordeel dat (gezonde) planten in deze band eenvoudiger te scheiden zijn van niet-planten zoals de achtergrond. In de werken van onder meer Lottes et al. (2016) [42], Potena et al. [44] en Milioto et al. (2017) [49] wordt op deze manier een segmentatie uitgevoerd van vegetatie en achtergrond, wat de detectie van planten vereenvoudigt. Het belangrijkste nadeel is echter dat sensoren die NIR en RGB omvatten een veel hogere kostprijs kennen en de spatiële resolutie van de RGB-banden een stuk lager ligt.

Het verder uitbreiden van de spectrale banden leidt tot hyperspectrale camera's. Dit soort camera's bestaat uit een verzameling van sensoren die elk gevoelig zijn voor een bepaalde spectrale regio, waardoor een nog groter deel van het lichtspectrum omvat wordt. Het gebruik van hyperspectrale camera's in de precisielandbouw voor diverse toepassingen wordt nog volop onderzocht, ook binnen het ILVO. Het gebruik van hyperspectrale sensoren kan een hulp zijn om extra fysiologische en structurele karakteristieken van planten te onthullen die niet zichtbaar zijn in ons zichtbaar spectrum. Ook effecten van het weer, bijvoorbeeld de impact van vochtigheid en droogte, wordt beter zichtbaar in deze extra spectrale banden [38].

Daarnaast bestaan er ook nog infraroodcamera's (thermaal) die in staat zijn om beelden te

maken in het ver-infrarood gebied. Deze soort camera's kennen een lage spatiële resolutie en worden voornamelijk ingezet bij toepassingen omtrent temperaturen [36].

Kortom, een belangrijke afweging moet worden gemaakt qua kostprijs en spatiële resolutie voor het al dan niet gebruik van extra spectrale banden. Ook de toepassing waarvoor de sensoren geselecteerd worden speelt een belangrijke rol. In dit werk wordt gebruik gemaakt van RGB-beelden om te kijken of de ontwikkeling van een algemeen systeem mogelijk is zonder gebruik van NIR. In toekomstig werk kan onderzocht worden wat de impact is van de toevoeging van NIR op de robuustheid en accuraatheid van het systeem.

3.2.3 Data (pre)-processing

Alvorens de data door het trainingssysteem gestuurd wordt, kan ze bewerkt worden (pre-processing). In het algemeen heeft dit als doel om de diversiteit in de invoerdata te minimaliseren. Op die manier wordt bijvoorbeeld ruis verwijderd, de effecten van belichting geminimaliseerd of het contrast verbeterd [43, 53]. Zo toonde Slaughter et al. [41] bijvoorbeeld aan dat het effect van lichtinval op planten een grote invloed heeft op de nauwkeurigheid van het systeem. Het is dus van belang dat een goede pre-processing dit effect zoveel mogelijk minimaliseert.

Plant/achtergrond scheiding

Binnen de context van detectie, segmentatie en classificatie van planten is het gebruikelijk om de planten van de achtergrond te scheiden (vegetatie detectie) alvorens het systeem te trainen. In het werk van Hamuda et al. [53] wordt een overzicht gegeven van 'image processing techniques' met elk hun voor- en nadelen, alsook combinaties van technieken die reeds onderzocht werden. Deze worden ingedeeld in drie grote categorieën: kleurindex-, drempelwaarde- en learning-gebaseerde technieken. Hieronder volgen kort enkele veelgebruikte technieken die ook in literatuur rond onkruiddetectie aan bod kwamen (zie tabellen 3.2 en 3.3).

Eerst en vooral is er het gebruik van kleurindices waarbij kleuren van belangrijke regio's in de afbeelding geaccentueerd worden, terwijl de achtergrondkleuren verzwakt worden. Op die manier worden vele RGB-afbeeldingen omgezet naar een soort van alternatieve kleurruimte. Deze methoden zijn in het algemeen vrij eenvoudig wat betreft complexiteit, maar kennen als zwakte dat de nauwkeurigheid sterk samenhangt met de lichtinval. Bovendien is er nood aan een drempelwaarde om een effectieve segmentatie te bekomen.

Een index die in verscheidene werken ([48, 50, 54, 52]) aan bod komt is de **Excess Green Index (ExG)**. Deze index werd ontwikkeld door Woebbecke et al. [55] en maakt gebruik van de RGB-waarden. Deze RGB-waarden worden genormaliseerd ($R^* = \frac{R}{R_{max}}$, $G^* = \frac{G}{G_{max}}$, $B^* = \frac{B}{B_{max}}$)

en vervolgens omgezet tot chromatische coördinaten (r , g en b) met behulp van onderstaande formules:

$$r = \frac{R^*}{R^* + G^* + B^*}, g = \frac{G^*}{R^* + G^* + B^*}, b = \frac{B^*}{R^* + G^* + B^*} \quad (3.1)$$

Deze chromatische coördinaten worden vervolgens gebruikt om een soort van kleuruimte te bepalen met een gewijzigde tint: $2g - r - b$, de ExG. Deze index blijkt een goed contrast te voorzien tussen plant en bodem.

Een andere kleurindex, Normalized Difference Vegetation Index (NDVI) komt aan bod in de werken van onder meer Lottes et al. (2016) [42], Potena et al. [44] en Milioto et al. (2017) [49]. Deze index maakt gebruik van de eerder vermelde NIR-band waarin vegetatie het nabij-infrarood terugkaatst. Er ontstaat bijgevolg een verschil in reflectie tussen zichtbaar licht en nabij-infrarood licht. De grootte van het verschil hangt af van de belichting, waardoor dit genormaliseerd moet worden. Er wordt gebruik gemaakt van de NIR-afbeelding (I_{NDVI}) en de rode band van de RGB-afbeelding (I_R). De index wordt per pixel (i,j) bepaald met onderstaande formule:

$$I_{NDVI(i,j)} = \frac{I_{NDVI(i,j)} - I_{R(i,j)}}{I_{NDVI(i,j)} + I_{R(i,j)}} \quad (3.2)$$

Lottes et al. (2016) [42] maken gebruik van NDVI voor de plant/achtergrond segmentatie. In hun daaropvolgend werk [48] geven ze een aanvulling van hun methodiek zodat deze ook werkt met enkel RGB-beelden. Men toont aan dat de ExG ook een geschikte index biedt voor de identificatie van vegetatie o.b.v. RGB-afbeeldingen. Hierbij moet echter rekening gehouden worden dat in het geval van andere groene objecten die geen vegetatie zijn, deze foutief als vegetatie kunnen worden gedetecteerd.

Het gebruik van ExG of NDVI om bijvoorbeeld planten van onkruid te onderscheiden is dus niet evident, vermits deze op spectraal vlak (o.a. groene kleur) vaak sterk op elkaar lijken. Voor detectie van zieke planten, waar niet altijd meer aan fotosynthese wordt gedaan, geeft NDVI bijvoorbeeld ook een vertekend beeld. Het gebruik van kleurindices is bijgevolg sterk afhankelijk van de toepassing waarin ze gebruikt worden.

De eenvoud van deze kleurindex-gebaseerde methoden heeft zoals vermeld het nadeel dat de nauwkeurigheid afhankelijk is van de lichtinval. In het geval van veranderende lichtinval, maar ook bijvoorbeeld bodemgesteldheid, zelfs binnen een afbeelding, blijkt het gebruik van zo'n techniek niet voldoende robuust. In het werk van Potena et al. [44] tracht men dit te optimaliseren door NDVI te combineren met een eenvoudig CNN. Voor elke pixel die met behulp van NDVI geclassificeerd wordt als vegetatie, wordt het CNN gebruikt om dit te valideren.

Een tweede soort technieken zijn de drempelwaarde-gebaseerde technieken. Dit soort technieken worden voornamelijk gebruikt wanneer er sprake is van twee klassen, bijvoorbeeld plant of achtergrond. Bah et al. [50] maken gebruik van zo'n drempelwaarde na de bepaling van ExG om

een binair masker plant/achtergrond te bekomen. Een typische methode, die ook in [50] gebruikt wordt, is de Otsu's methode. Bij deze methode tracht het algoritme de variantie binnen de verschillende klassen te minimaliseren om een optimale drempel te bekomen. Ook Pérez-Ortiz et al. [51, 52] maken gebruik van Otsu na het gebruik van ExG, in dit geval om een pre-labeling (zie verder) te realiseren. Deze soort methoden hebben als belangrijk nadeel dat de drempel bijgestuurd moet worden in veranderde condities. De keuze van een goede drempel is dus belangrijk vermits de performantie hier sterk van afhangt [53].

Tenslotte zijn er de learning-gebaseerde technieken, waarbij gebruik gemaakt wordt van unsupervised of supervised machine learning methoden. Deze ML methoden maken bijvoorbeeld gebruik van kleurindices en pogen de segmentatie te verbeteren op vlak van variaties in onder meer belichting. Deze methoden zijn bijgevolg meer gesofisticeerd. Het grootste nadeel van deze technieken is dat de benodigde computertijd toeneemt, waardoor het vaak niet realistisch is om deze op dit ogenblik te gebruiken in realtime applicaties [53].

Een alternatief dat recentelijk door Fawakherji et al. [46] werd voorgesteld voor een pixel-gebaseerde scheiding tussen plant en achtergrond, betreft het gebruik van een semantisch segmentatie netwerk (aangepaste versie van UNet met encoder-decoder architectuur). Bij semantische segmentatie wordt elke pixel geëvalueerd, in dit geval dus een plant of achtergrond. Ook bij Lottes et al. (2018) [43] wordt gebruik gemaakt van zo'n semantische segmentatie, maar dan direct voor classificatie tussen onkruid, plant en achtergrond in plaats van als pre-processing techniek.

Kortom, in heel wat studies wordt gebruik gemaakt van image processing technieken. Het is niet omdat een bepaalde techniek in een bepaalde toepassing met bepaalde afbeeldingen goed functioneert, dat dit ook het geval is bij een ander type toepassing of afbeeldingen. In het algemeen blijven veranderingen in licht en schaduw, alsook een complexe achtergrond een uitdaging voor verder onderzoek.

Van pixel tot blob tot object

In verschillende werken vindt na het uitvoeren van een vegetatie detectie op pixelniveau een tweede bewerking plaats die bepaalde pixels gaat combineren tot vormen en objecten.

Een eerste soort groepering die regelmatig aan bod komt is een blob. Een blob is een verzameling pixels in een afbeelding die gelijkaardige eigenschappen hebben en zo een soort van regio vormt. Er wordt dus overgegaan van pixel-gebaseerde segmentatie naar blob-gebaseerde segmentatie. Deze techniek wordt in diverse toepassingen in de precisielandbouw gebruikt. Yamamoto et al. [56] gebruiken dit bijvoorbeeld voor de detectie van rijp fruit voor eventuele implementatie van een plukrobot, en bij Chen et al. [57] voor automatische tellingen van fruit.

De eigenschap waarop in de werken omtrent onkruiddetectie gegroepeerd wordt, is het al dan niet zijn van vegetatie. Op basis van de input-afbeelding en het binaire masker worden er stapsgewijs pixels samengevoegd die behoren tot belangrijke regio's, in dit geval dus pixels gelabeld als vegetatie. Op die manier ontstaan er verzamelingen aaneengesloten pixels met de gedeelde eigenschap vegetatie, de blobs. Vervolgens wordt een selectiekader rond de blob bepaald en worden er van de originele afbeeldingen patches gevormd als input voor verdere training. Dit systeem is terug te vinden in werken van onder meer Fawakherji et al. [46] en Milioto et al. [49].

Potena et al. [44] testen in hun methode zowel classificatie op basis van pixels als op basis van blobs. De blobs werden ook hier gevormd op basis van het onderscheid in vegetatie/achtergrond. Ze constateerden dat de versie met blobs niet alleen betere resultaten haalt (van 92% naar 98%), maar ook sneller gaat. De invoering van blobs kent bijgevolg een positieve invloed op het geheel.

In het werk van Bah et al. [50] wordt er gebruik gemaakt van segmentatie o.b.v. superpixels om een masker te bekomen rondom de rijen planten. De superpixels worden gevormd op basis van het simple linear iterative clustering algoritme, waarbij er lokale groepen van pixels gemaakt worden op basis van hun spectrale waarden en hun ruimtelijke nabijheid. Op die manier wordt de volledige afbeelding verdeeld in grotere regio's pixels [28, 58].

Pérez-Ortiz et al. [51, 52] maakt gebruik van Object-based image analysis (OBIA). Hierbij wordt een afbeelding verdeeld in verschillende objecten en worden deze objecten de eenheden voor verdere verwerking. Objecten kunnen gezien worden als regio's binnen een afbeelding die afgeleid worden op basis van gelijkenissen qua kenmerken in een of meer dimensies [52, 54]. Hiervoor wordt na de pre-labeling (bodem, plant of onkruid) van de pixels een clusteringsalgoritme gebruikt om objecten te vormen voor elke klasse. Het voordeel is dat objecten naast de spectrale eigenschappen ook informatie op hoger niveau bevatten. Bovendien levert het gebruik van objecten als eenheden (vereenvoudiging) een positieve bijdrage aan de computationele belasting.

Het samenvoegen van pixels tot op een zeker niveau levert in het algemeen een positieve bijdrage op in het systeem, vermits het algoritme op een hoger niveau extra informatie meekrijgt en de tijd voor verdere verwerking verlaagd wordt. Voor de samenvoeging zijn er diverse parameters, afhankelijk van de gebruikte methode, die bepaald moeten worden. Een voorbeeld hiervan is de waarde van compactheid bij superpixels, waarbij een hogere waarde de superpixel een regelmatigere vorm geeft. Wanneer bijgevolg gebruik gemaakt wordt van dit soort technieken, is het van belang om hier aandacht aan te spenderen.

Lottes et al. (2018) [43] maken daarentegen geen gebruik van groeperingen van pixels maar gaat extra informatie creëren door het gebruik van een encoder-decoder structuur voor visuele kenmerken met een extra parallelle decoder die instaat voor tijd-ruimte eigenschappen. In dit geval wordt dus gebruik gemaakt van pixel-gebaseerde segmentatie

3.2.4 Training van modellen

Eens de eventuele bewerkingen op de data zijn uitgevoerd, kan deze aan de modellen gegeven worden. Voor de opbouw en training van de gebruikte modellen, zijn er ook heel wat variaties mogelijk.

Supervised versus unsupervised leren

Zoals in hoofdstuk 2 reeds aan bod kwam, kan er een onderscheid gemaakt worden in soorten van leren. Ook in de literatuur rond precisielandbouw komen deze diverse vormen van leren aan bod. Supervised learning is terug te vinden in heel wat toepassingen zoals onkruiddetectie [42, 47, 48], ziekte-detectie [59], opbrengst voorspellingen [60], bepaling van rijpe vruchten [56], tellingen van vruchten [57] etc. In de literatuur is er een duidelijke meerderheid van methoden op basis van supervised learning terug te vinden.

Het grote nadeel is echter dat er bij supervised learning een voldoende grote, gevarieerde en gelabelde databank nodig is. Regelmatig wordt hiervoor beroep gedaan op eerder online geplaatste databanken, maar hier komt ook weer het probleem naar boven van de grote variabiliteit in de natuur [59]. Wanneer nieuwe ongelabelde data beschikbaar is voor een toepassing, kost het heel wat tijd om deze data volledig te labelen. Bovendien zijn sommige datasets zo complex dat er nood is aan een expert in het vakgebied voor het uitvoeren van de labeling. Het hebben van gelabelde data is een enorme bottleneck in de schaalbaarheid van toepassingen die gebruik maken van supervised learning. Het vermijden van deze labeling vormt dan ook een echt uitdaging in het verder onderzoek naar het gebruik van computersie in de precisielandbouw.

In enkele werken tracht men deze bottleneck te reduceren op diverse manieren. Pérez-Ortiz et al. [52] introduceren een nieuwe methode waarbij relevante patronen geselecteerd worden, die vervolgens door de gebruiker gelabeld moeten worden. Uit hun eerder gedane onderzoek [51] bleek immers dat er gemiddeld zo'n 100 patronen per klasse gelabeld moeten worden om goeie resultaten te bereiken. Om de taak van de gebruiker zoveel mogelijk te verlichten, wordt op basis van ExG en Otsu's thresholding een pre-labeling gedaan van de pixels, zodanig dat de gebruiker enkel de fout geclassificeerde objecten moet labelen. Dit kan gezien worden als een vorm van semi-supervised leren.

Ook Potena et al. [44] spendeerden aandacht aan het feit dat labeling een bottleneck is. Hierbij wordt een eenvoudig algoritme geïntroduceerd dat zorgt voor een automatische selectie van een deelverzameling met goeie dekking van de eigenschappen uit de originele dataset. Men start van een grote dataset en poogt iteratief een optimale deelverzameling te bepalen met maximale visuele entropie. Dit datareductie algoritme wordt 'unsupervised data summarization'

genoemd. Dit probleem kan eigenlijk geformuleerd worden als een speciale vorm van het knapzak probleem¹.

Een vervolg op het werk van Potena et al. zet nog een stap verder in de omzeiling van de bottleneck. Het werk van Cicco et al. [62] probeer namelijk om een synthetische dataset te genereren, waarbij de belangrijkste eigenschappen van de omgeving waarin men wenst te werken gerandomiseerd worden. Het gaat hierbij om eigenschappen zoals de soorten gewassen en onkruiden, het bodemtype en de lichtinval. Door de parameters van deze methode aan te passen, alsook een aantal voorbeelden van de werkelijke situatie toe te voegen, kan op een relatief eenvoudige manier een grote hoeveelheid data bekomen worden.

Miloto et al. (2018) [45] stellen een aanpak voor waarbij heel wat extra informatie meegegeven wordt aan het encoder-decoder model, om ervoor te zorgen dat het systeem met een minimale hoeveelheid data zich kan aanpassen aan een nieuwe situatie. Op die manier kan een model getraind worden met een gekende dataset en kan vervolgens met beperkte nieuwe data het model in een andere situatie gebruikt worden.

Naast deze aanpakken die trachten om de kost van het labelen zoveel mogelijk omlaag te halen, stellen Bah et al. [50] een unsupervised aanpak voor. Hierbij maakt men gebruik van het feit dat planten in veel gevallen ingezaaid worden in rijen. Door eerst een detectie van de rijen te doen (m.b.v. ExG, Otsu, superpixels etc), kan alle vegetatie die buiten deze rijen valt reeds geclassificeerd worden als onkruid. Er wordt vervolgens een masker gegenereerd dat op basis van intersectie tussen superpixels en de gedetecteerde lijnen de gewassen gaat markeren. Vervolgens worden deze gelabelde elementen gebruikt voor verdere classificatie van delen van de afbeelding met potentieel onkruid in de rij zelf. Een aangehaald nadeel is dat door het gebruik van ExG bepaalde delen van minder groene planten en onkruiden als bodem worden gezien. Het gebruik van multispectrale afbeeldingen kan hier mogelijk een verbetering bieden. Een ander nadeel van deze methode is ook dat er strikte noodzaak is aan het feit dat de planten in rijen gezaaid moeten worden.

In dit werk wordt een methode ontwikkeld in hoofdstuk 5 waar deze bottleneck van labeling zoveel mogelijk vermeden wordt door gebruik te maken van semi-supervised learning op een actieve manier. In hoofdstuk 4 omtrent classificatie waren gelabelde datasets beschikbaar en wordt gebruik gemaakt van supervised learning.

¹Het knapzak probleem is een np-compleet probleem. Er wordt een verzameling van n objecten gegeven met elk een afmeting en waarde, alsook een knapzak met een bepaalde capaciteit. De vraag is om objecten in de knapzak te plaatsen met zo'n groot mogelijke totale waarde, zonder de capaciteit van de knapzak te overschrijden [61]

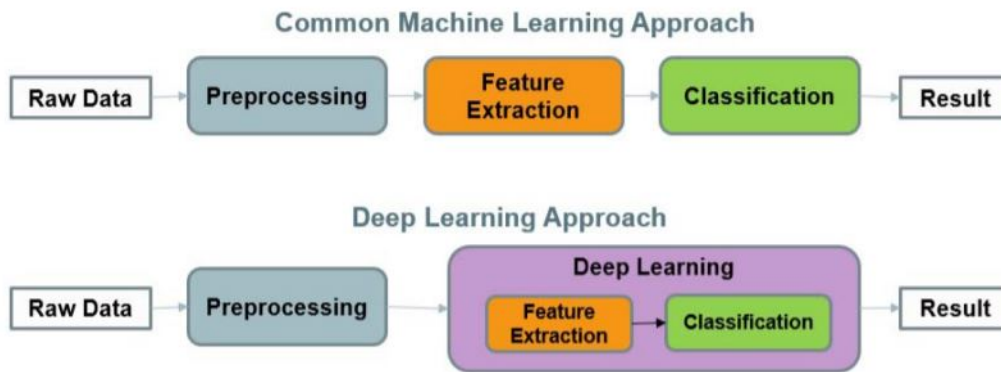
Machine learning versus deep learning

Uit huidige State of the art op vlak van computervisie-technieken in precisielandbouw, blijkt dat er voornamelijk gebruik gemaakt wordt van traditionele machine learning technieken zoals K-means, support vector machines, artificiële neurale netwerken, random forest etc. Ook in de diverse werken [15, 42, 48, 51, 52] uit deze literatuurstudie komen deze ML technieken regelmatig aan bod.

Naast deze ML technieken, ontstond een nieuw spectrum aan mogelijkheden door de ontwikkeling van deep learning. In veel andere domeinen werden reeds de voordelen van het gebruik van deep learning aangetoond, maar ook in de precisielandbouw kent deze recente technologie een groeiende populariteit. Het werk van Kamilaris et al. [9] geeft een overzicht van deep learning in precisielandbouw.

Uit de verscheidene werken die aan bod komen in Kamilaris et al. [9] blijkt dat CNN's sterk vertegenwoordigd zijn in de meest recente onderzoeken. Ook in recente werken van onder meer Milioto et al. [49, 45], Potena et al. [44], Bah et al. [50], Lottes et al. (2018) [43], Ferentinos et al. [59] en Fawakherji et al. [46] wordt duidelijk dat deep learning binnen de precisielandbouw een belangrijke opkomst kent de laatste jaren.

Een belangrijk verschil van deep learning ten opzichte van de traditionele ML, is het feit dat het ontwerp en de selectie van features niet meer handmatig hoeft te gebeuren. Bij het ontwerp van een algoritme is het immers meestal de bedoeling om bepaalde eigenschappen uit de data te halen die de variatie in de geobserveerde data kunnen omschrijven. Dit is echter niet zo eenvoudig, vermits er heel wat variatie mogelijk is. Deep learning tracht dit op te lossen door het gebruik van een feature hiërarchie. Na een eventuele pre-processing van de ruwe data gaat het deep learning model kenmerken van lagere niveau's (bv. krommen) gaan gebruiken in de training van hogere niveau's, om op die manier meer complexe kenmerken (bv. delen van het gezicht) te abstraheren [63, 9]. Kortom, bij een deep learning benadering gebeurt de feature extractie op een geautomatiseerd manier zoals geïllustreerd in figuur 3.4.



Figuur 3.4: Traditionele machine learning approach vs. deep learning approach. Afbeelding overgenomen uit [8].

Het grootste voordeel van deep learning is dus dat de menselijk interventie van feature engineering sterk gelimiteerd kan worden. Een nadeel is echter dat de modellen veel complexer worden en hierdoor een eerder langere tijd nodig hebben voor training. Het is echter wel zo dat dit soort problemen een hoge mate van parallelisatie mogelijk maken. Een ander nadeel werd hierboven reeds aangehaald en betreft de nood aan grote datasets met voldoende variatie [9].

Kamilaris et al. [9] haalt aan dat het erop lijkt dat deep learning de traditionele aanpakken stilaan overtreft, maar dat het erg moeilijk is om een vergelijkende studie te maken. Elk werk maakt immers gebruik van andere datasets, pre-processing technieken, parameters, modellen metrieken etc. Er kan echter wel geconcludeerd worden dat verder onderzoek naar deep learning interessante pistes kan opleveren voor de voortgang in de precisielandbouw.

3.2.5 Evaluatie van het systeem

Na de training van het model, volgt een evaluatie van het gehele systeem. Er zijn heel wat verschillende metrieken die de performantie van een systeem in kaart brengen. In de rest van dit werk zal gebruik gemaakt worden van enkele metrieken die in onderstaande tabel 3.1 worden opgelijst [9].

Tabel 3.1: Diverse metrieken [9].

Metriek	Beschrijving
Accuracy	Percentage van correcte voorspellingen
Precision	De fractie van geselecteerde items die correct geselecteerd werden
Recall	De fractie correcte items die geselecteerd werden
F1-score	Het harmonisch gemiddelde van precision en recall

De keuze van welke metriek er gebruikt kan worden hangt onder meer af van het doel van de toepassing. Wanneer getracht wordt een robot te ontwikkelen voor mechanische onkruidverwijdering, dient een hoge recall bereikt worden om ervoor te zorgen dat er veel onkruid verwijderd wordt.

Een belangrijke opmerking betreft dat een hoge accuracy niet meteen betekent dat het systeem goed werkt in het geval van een sterk ongebalanceerde dataset zoals bij onkruiddetectie. Hierbij ligt het aantal samples die onkruid vaak bevatten veel lager dan het aantal andere samples. Stel in het extreme geval dat het systeem alles als geen onkruid classificeert, kan alsnog een hoge accuracy bereikt worden omwille van deze grote verschillen qua aantallen.

Zoals eerder aangegeven is het niet eenvoudig om een vergelijking te maken tussen verschillende methodes in de literatuur door het gebruik van andere metrieken, modellen, datasets (real-life vs. specifieke afbeeldingen in gecontroleerde omgevingen), parameters etc. Bovendien hangen de resultaten ook heel sterk af van de soort toepassing. Bij classificatie van bijvoorbeeld bladeren, telling van fruit of plant herkenning worden in het algemeen goede resultaten behaald. Daartegenover liggen de resultaten vaak lager bij toepassingen die bijvoorbeeld een bodembedekking classificatie of onkruiddetectie [9].

Tenslotte nog een belangrijke opmerking, namelijk dat er regelmatig goede resultaten gepubliceerd worden, maar dat er geen aandacht werd gespendeerd aan de algemeenheid en flexibiliteit van het model. Er wordt vaak gebruikt gemaakt van dezelfde dataset voor training en testen, zonder dus rekening te houden met mogelijke veranderingen zoals lichtinval of bodemsoort. Het overdraagbaar maken en de veralgemening van het systeem zijn zoals eerder vermeld nochtans belangrijk voor het effectief inzetten van de toepassing in de praktijk [43, 9].

3.2.6 Overzicht

In de selectie van het gerelateerd werk werd de focus gelegd op (onkruid)detectie, vermits deze masterproef in eerste instantie als doel had om een supervised onkruiddetectiesysteem met autonome field robot te ontwikkelen. Omwille van het ontbreken van de nodige datasets met voldoende onkruid, kon dit niet uitgewerkt worden. Daarom wordt in deze masterproef toegespitst op classificatie en plantdetectie.

Een samenvatting van de methodieken voor detecties die in bovenstaande subsecties besproken werden, is terug te vinden in tabellen 3.2 en 3.3.

Tabel 3.2: Publicaties die gebruik maken van CV technieken voor (onkruid)detectie.

Paper	Samenvatting	Methodologie	Discussie
Hung et al. (2014)	Deze paper heeft als doel de classificatie van drie soorten onkruid in Australië (waterhyacint, tropical soda appel en serrated tussock) op basis van RGB-dronebeelden. Hiervoor wordt een learning-based algoritme voorgesteld die gebruik maakt van automatisch feature learning. De behaalde resultaten variëren tussen 70 en 90% F1 waarde, afhankelijk van diverse parameters.	<ol style="list-style-type: none"> 1. Sparse AE ter creatie van reeks afbeeldingsfilters o.b.v. CIFAR10 dataset 2. Convolutie afbeeldingsfilters met klassenaafbeeldingen voor feature vector bepaling 3. Texton gebaseerde aanpak van classificatie o.b.v. K-means clustering 4. Voorspelling door analoge feature extractie van input te vergelijken met texton gebaseerde voorstelling van de klasse 	<ul style="list-style-type: none"> + Vermijden menselijke interactie in feature selectie + Gebruik van reële RGB-afbeeldingen - Vereenvoudigingen in algoritme: <ul style="list-style-type: none"> • Voorafgaande selectie van patches i.p.v. volledige afbeelding → automatiseren door toevoeging vegetatie/achtergrond segmentering • Train en test dataset werden gebalanceerd
Lottes et al. (2016)	Deze paper beoogt detectie van suikerbieten en onkruid in een veld o.b.v. een combinatie tussen vegetatie detectie en DL met RGB- en NIR-afbeeldingen met mobiele robot.	<ol style="list-style-type: none"> 1. Vegetatie/achtergrond scheiding o.b.v. NIR-band m.b.v. de NYDI-index ter bepaling van een binair masker 2. Bepaling features (statistisch en vormelijk) van regio's die tot vegetatie behoren 3. Uitvoering RF-classificatie 4. Gebruik van informatie tussen nabijgelegen regio's m.b.v. Markov random fields 	<ul style="list-style-type: none"> + Mogelijkheid tot gebruik gekende spatiale informatie + Grootste deel onkruid correct geïdentificeerd, weinig false positives + Getest met diverse groeifasen - Manuele feature extractie - Lange runtime (4s per afbeelding) door vele features
(2017)	Mogelijkheid zonder gebruik van NIR o.b.v. droneafbeeldingen. Afhankelijk van de gebruikte index varieert de recall tussen 81 en 92%	Toevoeging geometrische features i.p.v. 4.	<ul style="list-style-type: none"> + RGB zonder NIR + Gebruik geometrische kenmerken voor plaatsbepalingen
Lottes et al. (2018)	Deze paper behandelt pixelwijze segmentatie van plant en onkruid voor robots m.b.v. fully convolutional netwerk inclusief sequentiële informatie (ruimtelijke ordening planten). De behaalde resultaten variëren en liggen algemeen rond of net boven de 90% F1 waarde.	Gebruik van AE voor visuele features, aangevuld met extra parallel pad die informatie meeneemt van de rijen	<ul style="list-style-type: none"> + Beter veralgemening voor ongeziene velden + Robuustheid in diverse groeifasen
Pérez-Ortiz et al. (2015, 2016)	Voor het maken van een overzichtskartaar van onkruid in zonnebloem en mais velden, gebruikt deze paper ML-methode met OBIA. De best behaalde resultaten liggen rond de 95% accuracy bij specifieke feature selectie.	<ol style="list-style-type: none"> 1. Gebruik van OBIA voor segmentatie tot objecten 2. Selectie van patronen voor labeling door gebruiker o.b.v. EXG met Otsu thresholding algoritme gevolgd door k-means clustering 3. Classificatie m.b.v. SVM, objectrepresentatie door histogrammen of data features 	<ul style="list-style-type: none"> + Menselijke interventie voor labeling beperkt tot correcties. - Grote verschillen tussen diverse gewassen - Features niet allen gewas-onafhankelijk - Parameter selectie is beperkt maar manueel

Tabel 3.3: Publicaties die gebruik maken van CV technieken voor (onkruid)detectie 2.

Potena et al. (2017)	In deze paper gaat men op basis van een multi-spectrale camera op field-robot een realtime plant- en onkruiddetectie uitvoeren gebruikmakend van 2 CNN's. De resultaten liggen tussen de 97-98% wat betreft nauwkeurigheid.	<ol style="list-style-type: none"> 1. CNN voor vegetatiedetectie bovenop de NVDI threshold 2. CNN (dieper) voor classificatie plant/onkruid (pixelgewijs) 3. Objectvoorspellingen maken o.b.v. gedetecteerde blobs en geclassificeerde pixels 	<ul style="list-style-type: none"> + Meer robuust systeem voor wijzigingen zoals licht en ondergrond dankzij dubbele threshold (NVDI-CNN) + Algoritme voor uitbreiding van de dataset om geen pixel per pixel labeling nodig te hebben
Mioto et al. (2017)	Op basis van RGB- en NIR-afbeeldingen afkomstig van een drone stelt men in deze paper een systeem voor van vegetatiedetectie gecombineerd met deep learning ter classificatie van suikerbieten en onkruid.	<ol style="list-style-type: none"> 1. Vegetatie/achtergrond scheiding o.b.v. NIR-band m.b.v. de NVDI-index ter bepaling van een binair masker 2. Bepaling blobs o.b.v. binair masker en creatie van image patch voor elke gedetecteerde blob 3. Supervised classificatie met CNN 	<ul style="list-style-type: none"> + Enkel input afkomstig uit afbeeldingen, zonder geometrische voorkennis + End-to-end aanpak zonder menselijke interventie + Eenvoudige aanpassing latere groeifasen door hertraining laatste laag - Afhankelijkheid van gelabelde databank
Mioto et al. (2018)	Realtime onkruiddetectie m.b.v. robot gebeurt in deze paper o.b.v. een CNN-gebaseerde semantische segmentatie	<ol style="list-style-type: none"> 1. Berekening diverse vegetatie indices en datarepresentaties als bijkomende informatie 2. Semantische segmentatie o.b.v. AE-architectuur 	<ul style="list-style-type: none"> + Opvangen probleem gelimiteerde dataset m.b.v. extra informatie uit de afbeelding + Beter veralgemening - Afhankelijkheid van gelabelde databank
Bah et al. (2018)	Deze paper voert een unsupervised labeling van data in om onkruid te detecteren in droneafbeeldingen. Hiervoor baseren ze zich op de detectie van rijen. De behaalde resultaten zijn vergelijkbaar aan een analoge gesuperviseerde methode.	<ol style="list-style-type: none"> 1. Automatische detectie van lijnen (ExG en Otsu threshold) waar planten staan om onkruid tussen rijen te detecteren met gebruik van superpixels 2. Gebruik van onkruid tussen rijen als database voor training CNN 3. Detectie van onkruid in de rijen o.b.v. getraind CNN 	<ul style="list-style-type: none"> + Unsupervised ter vermindering van dure labeling + Flexibel in nieuwe datasets - Nood geometrische eigenschap van planten op rijen - ExG levert beperkte achtergrond segmentatie, toekomstige optimalisatie door integratie NIR voor betere segmentatie als pre-processing
Fawakherji et al. (2019)	Deze paper beschrijft een DL-gebaseerde methode voor onkruid/plant classificatie op basis van RGB-afbeelding bekomen door een robot. De behaalde resultaten liggen rond de 90% nauwkeurigheid alsook precision.	<ol style="list-style-type: none"> 1. Pixel per pixel segmentatie m.b.v. AE tussen vegetatie en achtergrond 2. Gebruik van binair masker voor blobs van vegetatie te bekomen 3. CNN voor classificatie plant/onkruid 	<ul style="list-style-type: none"> + Onafhankelijk van de contextbruikbaar + Geen nood aan pixelwijs gelabelde data - Wel nood aan trainingsdata met gelabelde patches voor supervised classificatie

4

Classificatieproblemen in de precisielandbouw

Classificatieproblemen komen regelmatig aan bod binnen de precisielandbouw. Bij zo'n classificatieprobleem tracht men de afbeeldingen in te delen in afgebakende categorieën. De hoofddoelstelling van precisielandbouw is immers om op een kleinere schaal de behoeften te bepalen en hierdoor de behandeling van planten te optimaliseren. Dit gaat van een optimalisatie van de oogst, een meer duurzame werking met verlaagde hoeveelheid milieubelastende zaken, tot een efficiënter gebruik van werkuren. In dit hoofdstuk worden twee cases van classificatieproblemen besproken.

Beide cases werden uitgewerkt in Python 3.5, met behulp van verscheidene, open-source machine learning libraries zoals TensorFlow [64] en Keras [65]. De modellen werden getraind op een NVIDIA GeForce GTX 960M GPU.

Tot op de dag vandaag worden er echter bij veel van deze problemen niet zo'n efficiënte methodes gebruikt, waarbij de classificatie gebeurt door experts. Hierbij gaat veel kostbare tijd verloren. In deze masterproef werd nagegaan of geautomatiseerde classificatie mogelijk is bij twee projecten die op dat moment gaande waren op het ILVO: de herkenning van ritnaalden en de classificatie van bloemkolen op basis van hun oriëntatie.

4.1 Classificatie van ritnaalden

Ritnaalden zijn de larven van kevers uit de familie van kniptorren. Ze worden onderverdeeld in drie types: *Agriotes*-species, *Adrastus*-species en *Hemicrepidius*-species. In deze case gebeurt een classificatie van drie soorten binnen de *agriotes*-species.

Deze ritnaalden berokkenen schade aan gewassen zoals witloof, maïs en aardappelen. In 2015 gaf de Vlaamse overheid groen licht voor een meerjarig project om dit insect op een biologische manier te kunnen beheersen, met als eerste doelstelling het in kaart brengen van de verspreiding van de verschillende soorten. Daarna wou men op basis van die identificatie op zoek gaan naar een duurzame aanpak van dit probleem in de landbouw [66].

Het doel van deze case is de classificatie te automatiseren omdat op dit ogenblik experts de ritnaalden nog steeds handmatig classificeren, een tijdrovende opdracht. Een belangrijke kanttekening hierbij is dat deze classificatie niet visueel, maar op basis van DNA-onderzoek gebeurt.

4.1.1 Ontwerp en implementatie

Verkenning van data

Bij aanvang van dit project werd er een database ter beschikking gesteld met een bijhorende labeling en extra informatie (o.a. gemeten eigenschappen) in een Excel-bestand. Deze database bestond uit zo'n 1411 afbeeldingen, maar bleek niet goed gebalanceerd:

- 190 afbeeldingen van *Agriotes sputator*
- 408 afbeeldingen van *Agriotes obscurus*
- 813 afbeeldingen van *Agriotes lineatus*

Dat de classificatie door experts op basis van DNA-onderzoek gebeurt, was niet gekend bij aanvang van deze case. Dit plaatst de mogelijkheden van een geautomatiseerd systeem zuiver op basis van visuele kenmerken in een ander opzicht.

Een alternatieve database heeft afbeeldingen op het hoger niveau waarbij ook drie types (*Agriotes*, *Adrastus* en *Hemicrepidius*) onderscheiden kunnen worden. Het probleem hierbij is echter dat deze dataset zeer beperkt is qua omvang (< 500 afbeeldingen) en kwaliteit. Er werd beslist hier niet mee verder te werken en alsnog door te gaan met de originele database van afbeeldingen binnen de *Agriotes*. Er werd verder onderzocht of de computer in staat is om visuele kenmerken uit de diverse soorten te halen die kunnen leiden tot een classificatie.

In de eerste fase van het onderzoek naar mogelijk automatische classificatie van de drie speciën werd er gebruik gemaakt van machine learning technieken om inzicht te krijgen in de opgemeten data.

In het Excel-bestand dat de labeling bevat, zitten immers nog heel wat andere opgemeten karakteristieken. Dit zijn karakteristieken, zoals de lengte van het lijf of de kop of de afstand tussen de kaken, die mogelijks een bijdrage kunnen leveren aan hoe het visuele onderscheid gemaakt kan worden. Een voorbeeld van deze waarden wordt getoond in tabel 4.1. De tabel bevatte niet voor alle afbeeldingen gemeten alle karakteristieken, wat resulteerde in complete gegevens van 850 obscurus, 471 sputator en 1474 lineatus.

Tabel 4.1: Voorbeelddata uit Excel-bestand.

Species	Body length	Body width	Head length	Head width	Distance between mandibles
Obscurus	3031	325	315	278	122
Sputator	5277	570	337	374	134
Lineatus	9133	920	690	909	305

Species	Last abdomen length	Last abdomen width	Distance between the pits
Obscurus	328	287	186
Sputator	661	360	266
Lineatus	1318	901	603

Om te kijken of er bepaalde verbanden waren tussen een specie en zijn karakteristieken, werd gebruik gemaakt van classificatie technieken. In een eerste fase werd de data genormaliseerd met behulp van StandardScaler: van elke waarde werd het gemiddelde afgetrokken en dit werd vervolgens gedeeld door de standaardafwijking.

Daarna werden er verschillende soorten modellen (classifiers) getraind, die beogen om op basis van de inputgegevens een classificatie te maken van de speciën. Deze modellen werden daarna ook getest en beoordeeld op hun accuraatheid qua indeling in de verschillende klassen. De gebruikte classifiers zijn de volgende [67]:

- **Logistische regressie:** dit is een lineair classificatiemodel waarbij de kans op een uitkomst gemodelleerd wordt met behulp van een logistische functie. Zowel een one-vs-rest als een multinomiale regressie met L1 of L2 regularisatie werden uitgetest.
- **Naïve bayes:** dit is een probabilistische classifier die de 'naïeve' aanname doet dat er onafhankelijkheid is tussen elk paar van kenmerken en gebaseerd is op het theorema van Bayes. Er werd gebruik gemaakt van een Gaussiaanse versie, waarbij de waarschijnlijkheid van de kenmerken gaussiaans verondersteld wordt.

- **K-Nearest-Neighbour (KNN)**: dit is een classifier waarbij de klasse van object wordt bepaald door middel van de K dichtste instanties te vinden bij het object.
- **MLP classifier**: het multilayer perceptron model is een feed-forward netwerk die gebruik maakt van de stochastic gradient descent (SGD) als optimalisatiefunctie.
- **Random Forest (RF)**: dit is een ensemble methode waarbij gebruik gemaakt wordt van een collectie beslissingsbomen. Er wordt telkens een steekproef van de data genomen, en hiervoor wordt dan een beslissingsboom opgebouwd tot een collectie gevormd wordt. Meerdere beslissingsbomen leidt tot meerdere voorspellingen, die uitgemiddeld worden tot het resultaat.

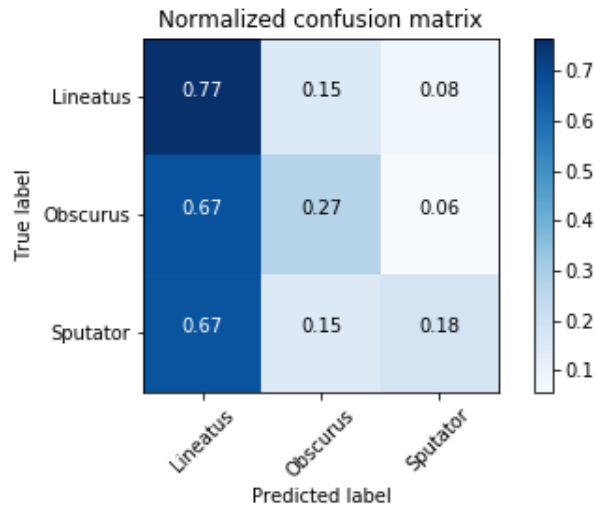
Naar aanleiding van Devi et al. [68] werd ook geprobeerd of een combinatie van enkele van deze voorgaande classifiers tot eventuele betere resultaten kon leiden. Bij ensemble learning wordt er een voorspelling gemaakt op basis van een aantal verscheidene modellen [28]. Random Forest is reeds een vorm van ensemble learning en in het werk van Paing en Choomcuay probeert men deze methode alsnog te optimaliseren. Er werden diverse combinaties van classifiers getraind en getest en de uiteindelijke voorspelling werd bepaald op basis van het gemiddelde van de afzonderlijke voorspellingen. In plaats van het gemiddelde te nemen, had de beslissing uit diverse combinaties ook beoordeeld kunnen worden door een extra model te trainen die de output van bovenstaande classifiers als input features krijgt. Omwille van de lage resultaten, zie ook hieronder, werd hier niet verder op ingegaan.

Tabel 4.2: Resultaten van diverse classifiers op de gemeten karakteristieken.

Classifier	Accuraatheid (%)
Multinomiale logistische regressie met L1 afstraffing	55.2%
Multinomiale logistische regressie met L2 afstraffing	55.2%
One-vs-rest logistische regressie met L2 afstraffing	55.4%
Gaussiaanse naïeve Bayes	39.8%
MLP classifier	54.9%
KNN (k=5) classifier	50.5%
Random forest classifier	51.5%
Combinatie: random forest + logistische regressie (L2-OvR)	52.6%
Combinatie: random forest + Gaussiaanse naïeve Bayes	49.5%
Combinatie: random forest + KNN	51.4%

De resultaten van de classifiers en enkele combinaties worden weergegeven in tabel 4.2. Deze blijken echter vrij slecht te zijn. Bij het van naderbij bekijken van de resultaten, blijkt dat de classifiers met de hoogste scores (o.a. one-vs-all logistische regressie) aan elke voorspelde waarde het label lineatus toekenden. Afhankelijk van de train-test splitsing ligt deze waarde iets hoger of lager, maar ook uit de verhouding in de totale dataset, zijn er 52% van de gegevens

lineatus. De genormaliseerde confusiematrix (random forest classifier) in figuur 4.1 bevestigt dit eveneens met een overwegend donkerblauwe linkerkolom bij de voorspelling van een lineatus. Een confusiematrix geeft de resultaten weer van het voorspelde label ten opzichte van het werkelijke label bij de elementen uit de testdataset.



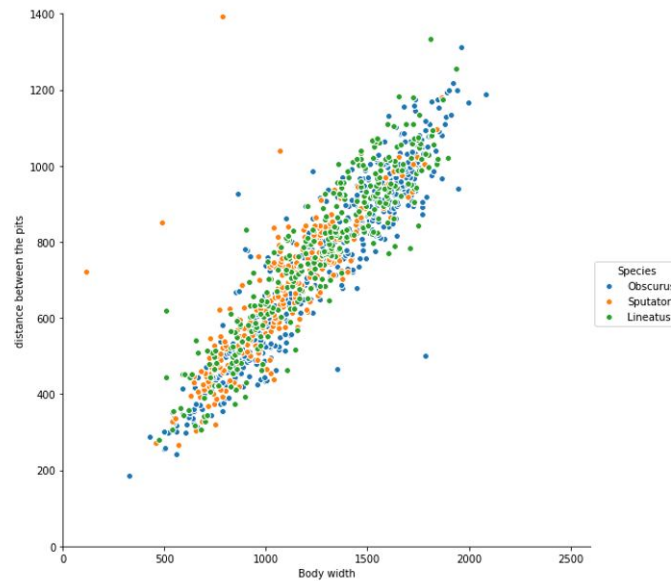
Figuur 4.1: Genormaliseerde confusiematrix na random forest classifier.

Dat de classificatie op basis van al deze gegevens niet zo evident is, blijkt ook uit de grafische weergave van de datapunten. In deze grafische weergave worden telkens twee gegevens ten opzichte van elkaar uitgezet en wordt elke instantie van een bepaalde specie in dezelfde kleur weergegeven. Een voorbeeld hiervan is terug te vinden in figuur 4.2, waarbij de 'distance between the pits' uitgedrukt wordt in functie van de 'body width', welke toch wel twee belangrijke visuele kenmerken zijn. Op deze figuur is het dan ook niet eenvoudig om de drie soorten speciën uit elkaar te halen, want er is een soort van wolk van datapunten, wat maakt dat ze ook niet zo eenvoudig te classificeren zijn. In bijlage A worden allerlei andere mogelijke combinaties van de verschillende gegevens in functie van elkaar getoond, waaruit blijkt dat dit verschijnsel van een wolk van datapunten in elke combinatie het geval is.

Er kan bijgevolg geconcludeerd worden dat er op basis van de uiterlijk gemeten kenmerken geen correcte classificatie mogelijk is.

Classificatie op basis van afbeeldingen: CNN

Na de verkenning van de beschikbare data (afbeeldingen en gegevens), wordt de mogelijkheid tot automatische classificatie van ritnaalden verder uitgewerkt met behulp van CNN's, een gebruikelijke computervisie-techniek zoals beschreven in 2.2.2. Vermits er gelabelde data beschikbaar is, wordt er bovendien gebruik gemaakt van supervised leren: er wordt een functie gezocht die



Figuur 4.2: Weergave van verdeling van speciën, uitzetting 'distance between the pits' in functie van 'body width'.

de input (afbeelding x) een label y toekent, waarbij de y -waarde behoort tot één van de drie speciën.

In eerste instantie wordt er gebruik gemaakt van de originele hoeveelheid afbeeldingen. Weliswaar in zwart-wit in eerste fase, later ook in RGB, alsook verkleind en bijgeknipt tot 200×200 afbeeldingen. De originele afbeeldingen waren immers te groot en te onregelmatig van vorm om als input te gebruiken. Bovendien werd de data genormaliseerd, zodanig dat de pixels een gelijkwaardige datadistributie kennen en het netwerk sneller kan convergeren. Het opgebouwde netwerk bevat twee convolutionele lagen, telkens gevolgd door een ReLU activatiefunctie en vervolgens een max-pooling laag. Na dit convolutionele gedeelte, wordt overgegaan op een fully connected laag met bijhorende ReLU en tenslotte een fully connected laag die de output weergeeft naar drie neuronen, drie klassen (speciën). Deze fully connected lagen fungeren als soort van classifiers die de high-level kenmerken uit de verscheidene convolutionele lagen gaan gebruiken om te leren classificeren. Na deze fully connected laag, wordt gebruik gemaakt van een **softmax activatiefunctie**. Deze functie wordt typisch gebruikt bij classificatieproblemen en zorgt ervoor dat de outputwaarden van het neurale netwerk worden omgezet in kansen die samen één zijn, zie ook formule 4.1. De output kan bijgevolg geïnterpreteerd worden als drie waarden die de kansen weergeven op de verschillende klassen. Op basis van de hoogste kans wordt dan de bijhorende klasse toegekend: **one hot encoding** zorgt voor een één bit bij de toegekende klasse en een nul

bit bij de andere klassen [1, 69].

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (4.1)$$

Na enkele pogingen en het afstemmen van enkele parameters (o.a. learning rate, initialisatie van gewichten, verschillende lagen), bleken alle netwerken zeer sterk te overfitten. Wanneer een netwerk overfitting kent, is dit zichtbaar in de weergave van de verliesfuncties voor training en validatie. De verliesfunctie is hier een cross-entropy of log loss functie, die als eigenschap heeft dat ze stijgt wanneer de voorspelde kans afwijkt van het effectieve label. Deze verliesfunctie wordt typisch gebruikt bij de softmax activatiefunctie.

Hier gaat de verliesfunctie behorende bij de trainingsfase sterk omlaag, het netwerk is in staat de trainingsset goed te herkennen. De verliesfunctie behorende bij de validatiefase gaat daarentegen niet omlaag, het netwerk kan niet veralgemenen. Dit is zichtbaar in figuur 4.3 aan de linkerzijde. Zoals in sectie 2.2.2 aan bod kwam, wordt er tijdens het trainen een loss functie gebruikt. In dit geval wordt gebruik gemaakt van de gemiddelde kwadratische fout of mean squared error (MSE) om de inconsistentie tussen de voorspelde klasse (\hat{y}) en de effectieve klasse (y) weer te geven. De MSE wordt berekend met formule 4.2, met N het aantal mogelijke klassen [1]:

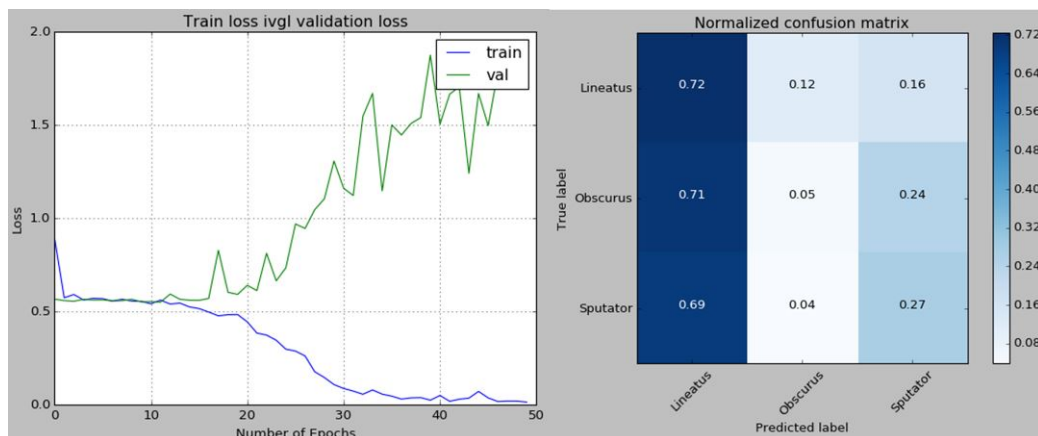
$$L = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2 \quad (4.2)$$

De verklaring hiervoor ligt voor een groot deel bij de beschikbare dataset: deze is te klein voor het netwerk met twee convolutionele lagen. Daarnaast blijkt uit de confusie matrix, figuur 4.3 rechts, dat het merendeel van de afbeeldingen gecategoriseerd wordt als lineatus, welke in vele gevallen onterecht is. Dit is te verklaren door de slechte verdeling van de dataset, er zijn immers vier maal zoveel afbeeldingen van lineatus als van sputator, en dubbel zoveel lineatus ten opzichte van obscurus.

Data augmentatie

Ten gevolge van bovenstaand probleem, een ongebalanceerde dataset, wordt een veelgebruikte techniek voor vermeerdering van data toegepast: data augmentatie. In dit geval werd gebruik gemaakt van rotaties: de afbeeldingen van de sputator werden driemaal geroteerd, die van de obscurus gespiegeld, met als gevolg dat er ongeveer 800 afbeeldingen van elke klasse zijn. Bovendien werd in de volgende fase gebruik gemaakt van de RGB afbeeldingen.

Naast deze basisbewerkingen om een gebalanceerde dataset te bekomen, werd het gebruik van een datagenerator kort uitgetest. Daarop worden batches van afbeeldingen gemaakt waarop realtime augmentatie (o.a. rotaties, verschuivingen, zoom) wordt toegepast. Dit leverde minder



Figuur 4.3: Links: loss-functie - Rechts: genormaliseerde confusiematrix behorende bij CNN met originele afbeeldingen.

dan 1% verbetering van het eindresultaat, maar was wel tijdsintensief, waardoor besloten werd hier niet mee verder te gaan.

Overfitting

Zoals in hoofdstuk 2 aan bod kwam, is er bij overfitting sprake van een lage bias en een hoge variantie. Om dit tegen te gaan zijn er verschillende mogelijkheden, waaronder het uitbreiden van de data, wat reeds gebeurde met data augmentatie, het gebruik van een andere netwerkarctitectuur, alsook de toevoeging van regularisatie.

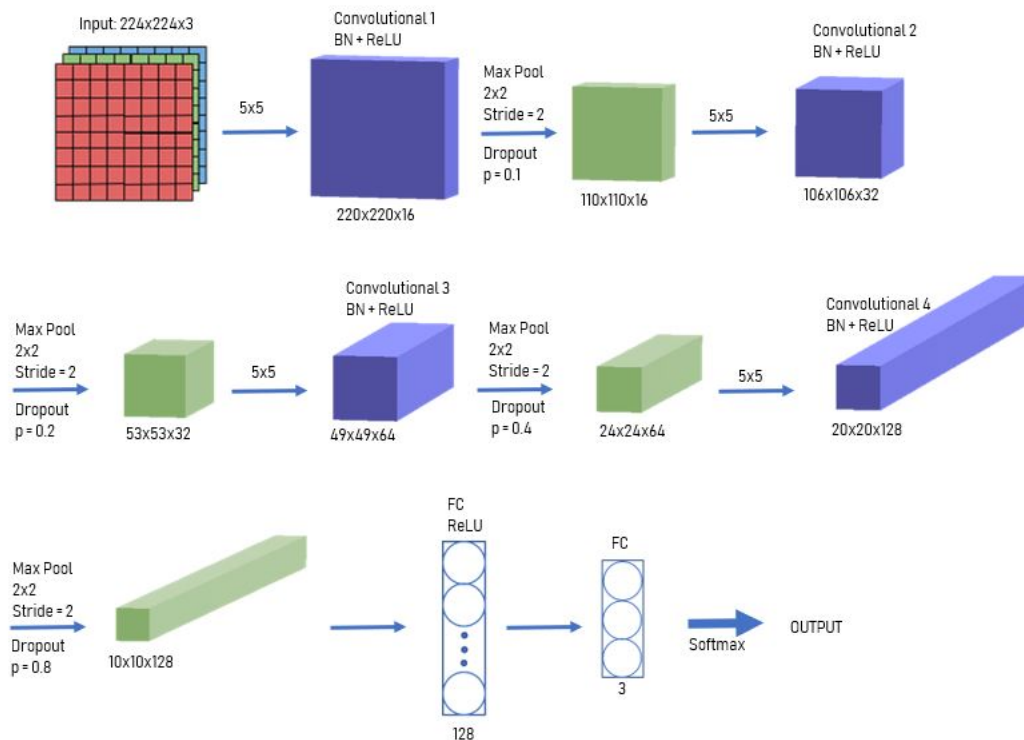
Om de sterke overfitting uit figuur 4.3 tegen te gaan, wordt de netwerkarctitectuur gewijzigd, worden er regularisatiemaatregelen geïntroduceerd, en worden de hyperparameters met trial and error bijgestuurd.

Regularisatie heeft als doel om de complexiteit van het netwerk te verminderen, om zo het feit dat het netwerk niet in staat is te veralgemenen tegen te gaan. Er bestaan verschillende soorten regularisatie, en in dit project werd enerzijds **dropout** en anderzijds **L2 regularisatie** gebruikt.

Dropout is, zoals de naam het aangeeft, het feit dat er random neuronen geëlimineerd worden tijdens de trainingsfase. De hoeveelheid neuronen die tijdelijk geëlimineerd worden, wordt bepaald door de hyperparameter p : de dropout rate. Vermits de neuronen willekeurig gekozen worden, zal het netwerk veranderen in elk epoch en kan het netwerk ook niet meer vertrouwen op een bepaalde node. Het gevolg hiervan is dat het netwerk zijn gewichten zal gaan verdelen, omdat de kans bestaat dat net die neuron met hoge gewichten eruitgehaald wordt. De gewichten worden kleiner, het netwerk wordt kleiner in aantal neuronen en dus wordt het geheel eenvoudiger.

L2 regularisatie heeft als hyperparameter λ . Deze parameter geeft aan hoe sterk hoge gewich-

ten afgestraft worden, waardoor hoge gewichten door het netwerk vermeden worden. Door de gewichten te verlagen, zal ook de gewogen som (zie 2.2) dalen, en de complexiteit van de activatiefunctie dus ook [70].



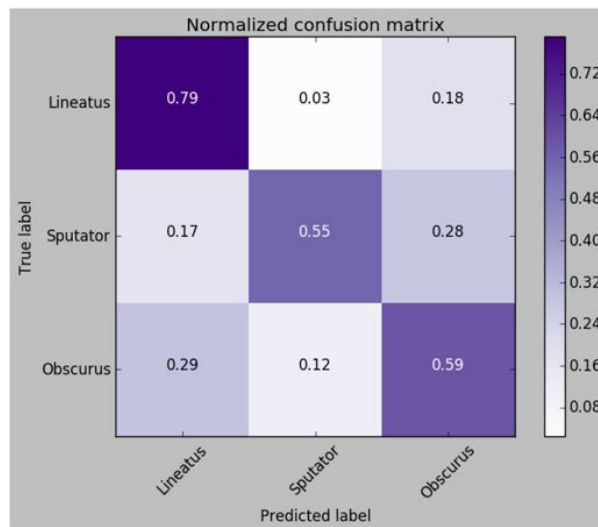
Figuur 4.4: Visualisatie netwerkmodel classificatie van ritnaalden.

4.1.2 Resultaten

Na toepassing van data augmentatie en het verfijnen van de hyperparameters om overfitting zoveel mogelijk te vermijden, wordt een nauwkeurigheid van 64,4% bereikt. De uiteindelijke keuze voor de parameters wordt weergegeven in tabel 4.3 De opbouw van het uiteindelijke netwerk wordt gevisualiseerd in figuur 4.4. In de bijhorende genormaliseerde confusie matrix, figuur 4.5 is te zien dat het netwerk in staat is om 79% van de lineatus correct te classificeren, maar dat de andere soorten moeilijker zijn.

Tabel 4.3: Keuze voor enkele hyperparameters in netwerk voor classificatie van ritnaalden.

Hyperparameter	Gekozen waarde
Aantal epochs	50
Batch size	16
Learning rate	0.001
Optimizer	Adam
L2-penalty	0.01



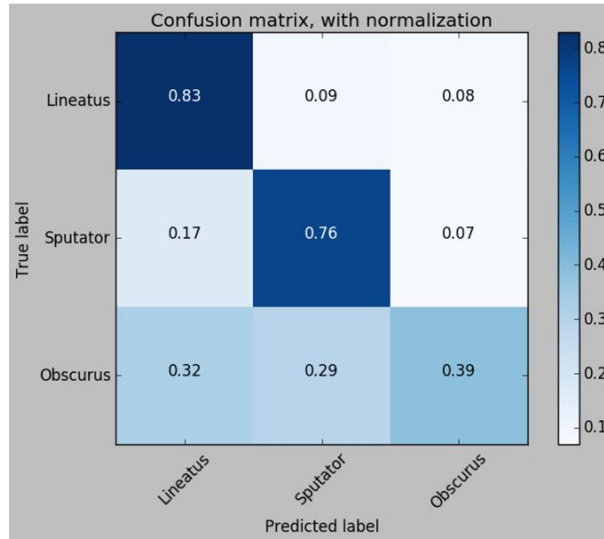
Figuur 4.5: Genormaliseerde confusie matrix bij classificatie van ritnaalden.

Alternatieve aanpak: One-VS-All classificatie

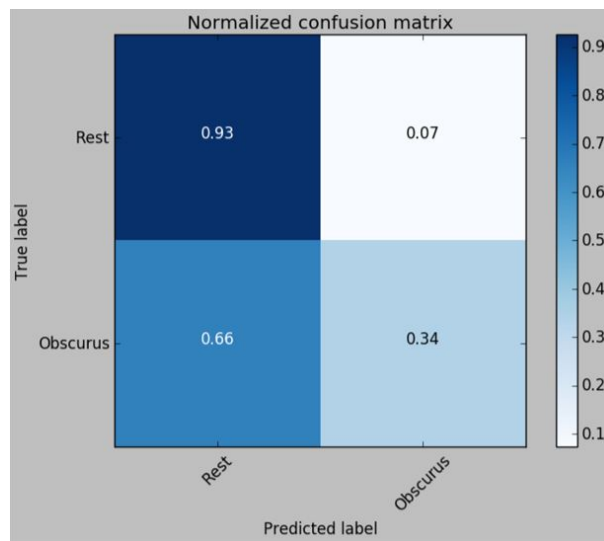
Ter aanvulling op bovenstaand netwerk, werd ook one-vs-all classificatie uitprobeerde. Bij dit soort classificatie, wordt het netwerk in dit geval driemaal getraind, eenmaal voor lineatus of 'rest', eenmaal voor sputator of 'rest' en eenmaal voor obscurus of 'rest'. In plaats van een drieweg output te bekomen, zoals in 4.4 in de laatste fully connected layer zichtbaar is, wordt nu eenwegs geclassificeerd. Na het trainen van de drie netwerken, zijn er bijgevolg drie waarden voor de kans dat het een bepaalde soort is ten opzichte van de rest. Vervolgens wordt er voor elke afbeelding bepaald welk van deze drie kansen het grootst is, en dat is dan de definitieve classificatie.

De nauwkeurigheid van dit systeem gaven een analoge nauwkeurigheid als bij de driewegs-classificatie, namelijk 64,2%. Uit de genormaliseerde confusie matrix, figuur 4.6, kan geconcludeerd worden dat het netwerk in dit geval in staat is om de lineatus en sputator te classificeren, maar dat de obscurus random verdeeld wordt over de drie klassen. Uit de confusiematrix beho-

rende bij het netwerk obscurus versus rest, figuur 4.7 bleek bovendien dat het netwerk vooral leerde wat de 'rest' was ten opzichte van de klasse zelf, mogelijks door de grotere hoeveelheid afbeeldingen in de restklasse.



Figuur 4.6: Genormaliseerde confusie matrix one-VS-all classificatie van ritnaalden.

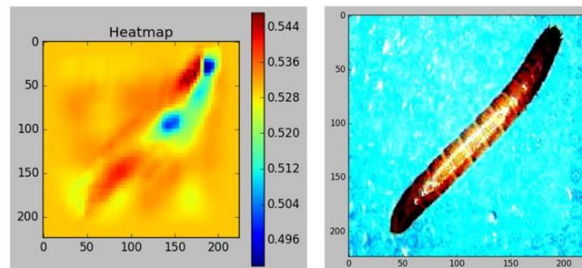


Figuur 4.7: Genormaliseerde confusie matrix 'obscurus VS rest'.

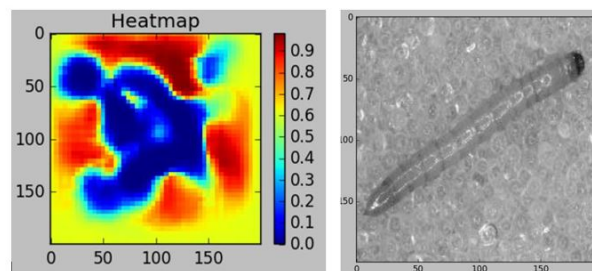
Visualisatietechnieken

Uit bovenstaande resultaten blijkt dat het netwerk alsnog in staat is om bepaalde zaken te classificeren, het haalt immers een nauwkeurigheid van 64% wat meer is dan de verwachte

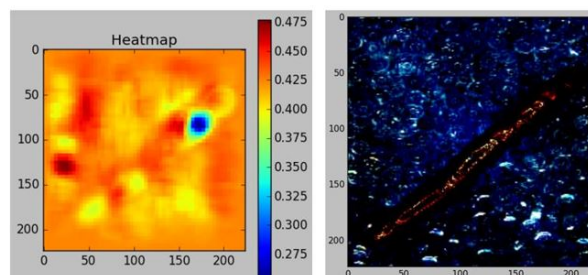
random 33%. De vraag die hierbij rees, is naar wat het model eigenlijk kijkt om de beslissing op te baseren en of niet random overfit wordt op een bepaalde keuze. Bovendien zou hier interessante informatie uitgehaald kunnen worden, namelijk de visuele kenmerken waarop een zeker niveau van classificatie mogelijk is.



Figuur 4.8: Heatmap van een correct voorspelde lineatus.



Figuur 4.9: Heatmap van een correct voorspelde sputator.



Figuur 4.10: Heatmap van een obscurus, voorspeld als lineatus.

Een interessante techniek om kenmerken te visualiseren, betreft het maken van heatmaps op basis van occlusie zoals beschreven in het werk van Zeiler en Fergus [71]. Het is immers een belangrijke vraag of het model effectief naar het object kijkt, of gewoon kijkt naar de achtergrond. Om dit te bereiken, wordt er systematisch een stukje van de afbeelding zwart gemaakt (occlusie) en vervolgens door het netwerk gestuurd. Wanneer het geoccludeerde stukje van de afbeelding door de computer als belangrijk werd geacht, zal het netwerk moeilijkheden ondervinden om correct te

voorspellen. Dit wordt vervolgens gevisualiseerd in een heatmap, waarbij de belangrijke stukken blauw worden gekleurd en de pixels die geoccludeerd werden maar geen invloed kenden op het resultaat rood.

Een interpretatie van enkele heatmaps levert wel op dat de computer voornamelijk naar het object zelf kijkt. In figuren 4.8-4.10 wordt telkens rechts de inputafbeelding (reeds na normalisatie) getoond en links de bijhorende heatmap. Bovendien wordt voornamelijk de kop in enkele heatmaps in het blauw voorgesteld. Echter is het onderscheid niet altijd goed te maken tussen de verschillende klassen, en lijkt ondanks de eerdere focus op de ritnaald zelf, er toch geen eenduidig verband te zijn.

Deze heatmaps zouden voor een betere interpretatie verder geanalyseerd moeten worden, weliswaar in nauwe samenhang met biologen om meer op bepaalde zaken te kunnen letten. Echter werd besloten om in deze masterproef hier niet verder op in te gaan, wegens te tijdrovend en het feit dat er nog andere onderzoeksprojecten mogelijk waren met andere, voor dit werk nuttigere, insteken.

4.1.3 Besluit

De opzet van dit onderzoek was het nagaan of er met behulp van computervisie een snellere manier van classificatie mogelijk is dan het huidige DNA onderzoek. Er kan geconcludeerd worden dat de computer in staat is om tot een zeker niveau kenmerken uit de afbeeldingen te halen, maar nog geen erg nauwkeurige classificatie kan maken. Uit de opgemeten uiterlijke karakteristieken in het Excel-bestand bleek reeds dat op basis van deze gegevens de classificatie moeilijk is. Ook de grootte van de dataset was een beperkende factor in het onderzoek. Bovendien kunnen experts in het vakgebied van ritnaalden geen classificatie maken op basis van visuele aspecten. Gegeven deze feiten, liggen de resultaten van nauwkeurigheid in de lijn der verwachtingen. Langduriger en intensiever onderzoek kan dus mogelijks betere resultaten opleveren, maar hierbij rijst de vraag of dit de moeite waard is.

4.2 Classificatie van bloemkolen op basis van de oriëntatie

Een tweede deelproject omtrent classificatie had een meer praktische doelstelling binnen het ILVO in kader van precisielandbouw. Het ging om de classificatie van de ligging van een bloemkool op een transportband in het kader van een studie rond automatisering van bloemkoolverwerking. Dit project maakte deel uit van een samenwerking met een ander bedrijf.

De opzet was het opbouwen van een geautomatiseerd systeem waarbij bloemkolen op een trans-

portband aangevoerd worden richting de onthartingsmachine met freeskop. Om deze bloemkolen te verwerken, is het van belang dat ze in correcte ligging aan de machine toekomen. De bloemkool dient met het hart beneden te liggen, en indien het hart naar boven ligt bijgevolg eerst gedraaid te worden. Er werd dus onderzocht of het haalbaar is te detecteren of de bloemkolen op de transportband correct liggen of niet.

4.2.1 Ontwerp en implementatie

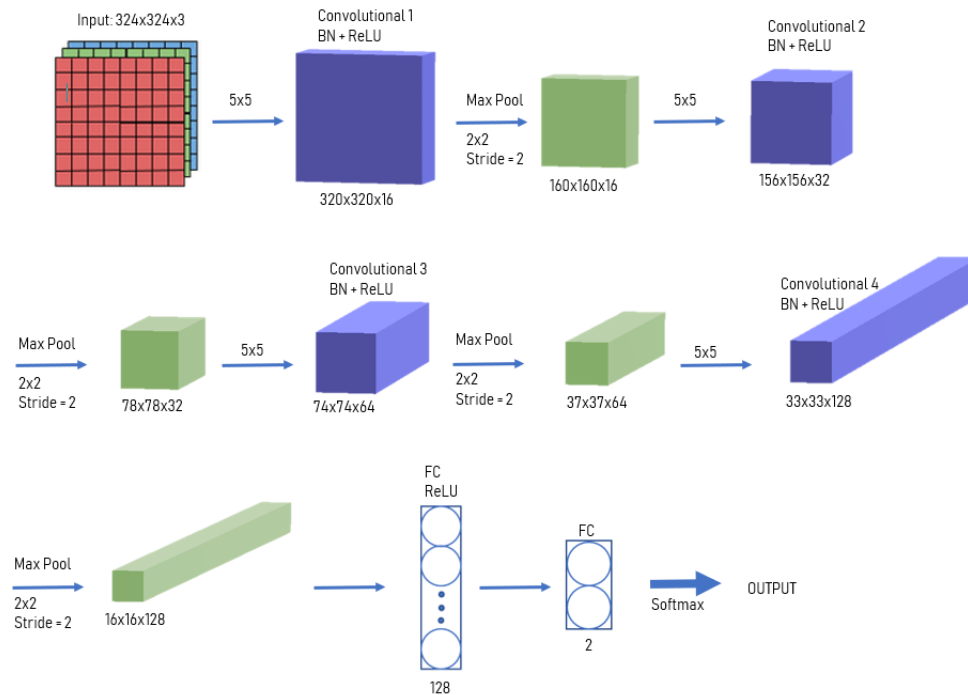
De beschikbare dataset werd speciaal voor dit project opgebouwd en bestond uit 449 foto's, waarbij er 239 met hart omhoog gelegen en 210 met hart naar beneden gelegen bloemkolen zijn. Vermits deze originele afbeeldingen een zeer hoge resolutie kenden (7952x5304 pixels), werden deze in de eerste fase sterk verkleind. Bovendien was het zo dat alle afbeeldingen op automatische manier genomen werden, waarbij de bloemkool duidelijk gecentreerd ligt door kalibratie van het toestel op vier zichtbare punten. Voor de eenvoud werd dan ook verdergegaan de afbeeldingen bij te knippen (crop) tot een vierkant. Dit resulteerde in een dataset met afbeeldingen zoals weergegeven in figuur 4.11



Figuur 4.11: Geschaald en gecropt (verkleinde weergave).

Na de eenvoudige bewerkingen op de afbeeldingen, werd er in de tweede fase een neurale netwerk opgebouwd zonder transfer learning. Transfer learning is een methode waarbij een eerder getraind model gebruikt wordt bij een ander, gerelateerd probleem. Er werd gekozen voor een convolutioneel netwerk bestaande uit vier blokken. De vier convolutional layers verdubbelen telkens qua filtergrootte en worden telkens gevolgd door een batch normalization en max pooling layer. Dit geheel wordt dan gevolgd door een dense layer met ReLU activation functie. De opbouw van het netwerk wordt gegeven in figuur 4.12.

De totale hoeveelheid foto's werd opgesplitst in een test-train verhouding van 20-80, en van die 80% trainingsafbeeldingen werd er nogmaals 20% afgesplitst voor de validation set.



Figuur 4.12: Visualisatie netwerkmodel voor classificatie bloemkolen.

Om tot dit eindnetwerk te komen, werden de hyperparameters en netwerkopbouw met behulp van trial and error bijgestuurd tot de waarden in tabel 4.4.

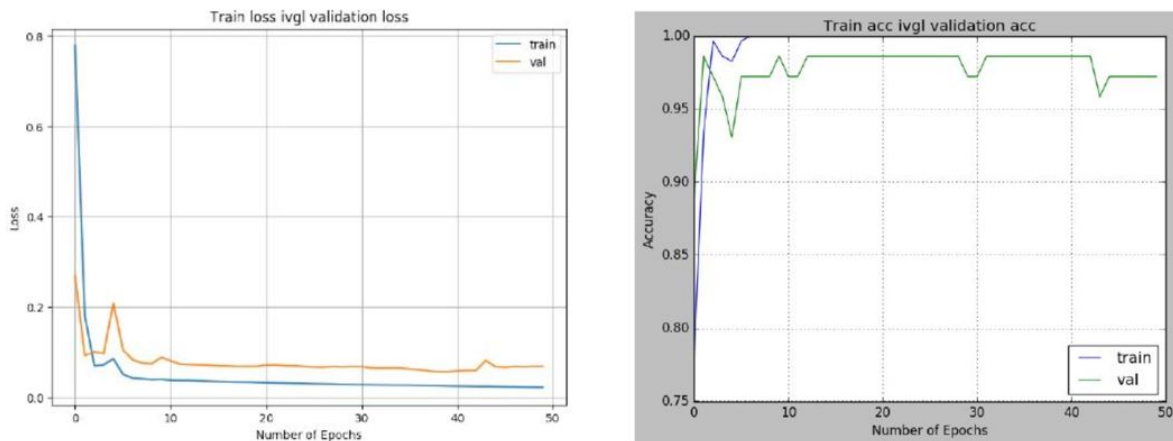
Tabel 4.4: Keuze voor hyperparameters in netwerk voor classificatie van bloemkolen op basis van de oriëntatie.

Hyperparameter	Gekozen waarde
Aantal epochs	50
Batch size	8
Learning rate	0.0001
Optimizer	Adam
L2-penalty	0.01

4.2.2 Resultaten

Tijdens de training kan gezien worden dat de verliesfunctie (MSE), alsook de nauwkeurigheid initieel sterk dalen/stijgen en vervolgens vrij constant blijven, weliswaar licht dalend. Dit wordt geïllustreerd in figuur 4.13.

Het uiteindelijke eindnetwerk zoals hierboven omschreven werd geoptimaliseerd tot een test-



Figuur 4.13: Links: verliesfunctie - Rechts: accuracy netwerk classificatie bloemkolen.

nauwkeurigheid van 98,89%.

Een opmerking hierbij is dat er ook enkele netwerken geprobeerd zijn met behulp van data augmentation, waarbij de afbeeldingen nog geroteerd en ingezoomd werden. Dit gaf echter exact dezelfde resultaten qua nauwkeurigheid en is dus geen meerwaarde in dit geval.

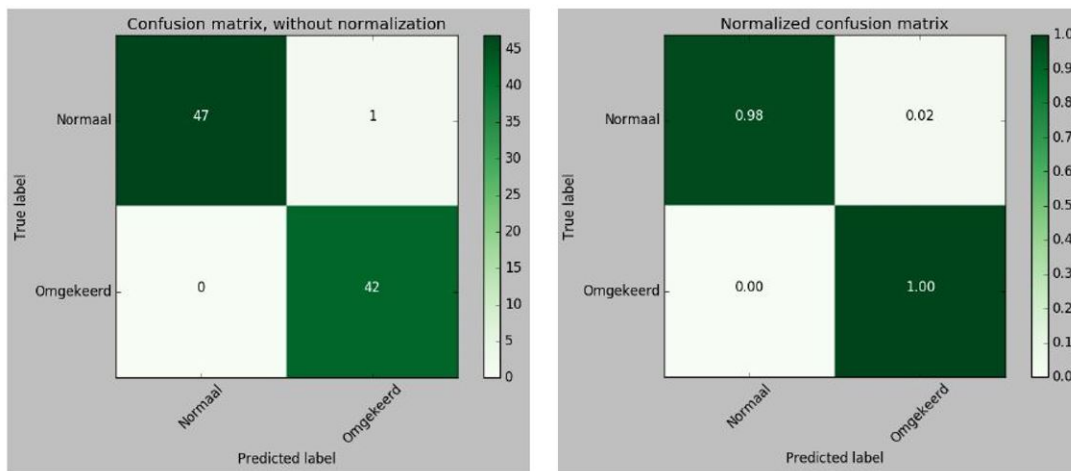
4.2.3 Besluit

Zoals in voorgaande paragraaf reeds vermeld werd, is de uiteindelijke nauwkeurigheid hoog, zijnde 98,89%. Bij visualisatie van de confusiematrix is het zichtbaar dat slechts één enkele afbeelding onterecht als omgekeerd werd ingeschat. De absolute voorstelling is terug te vinden in figuur 4.14 links en de relatieve voorstelling in 4.14 rechts. Dit toont aan dat een geautomatiseerd systeem voor de classificatie van de ligging van een bloemkool op de transportband met het voorgestelde systeem mogelijk is.

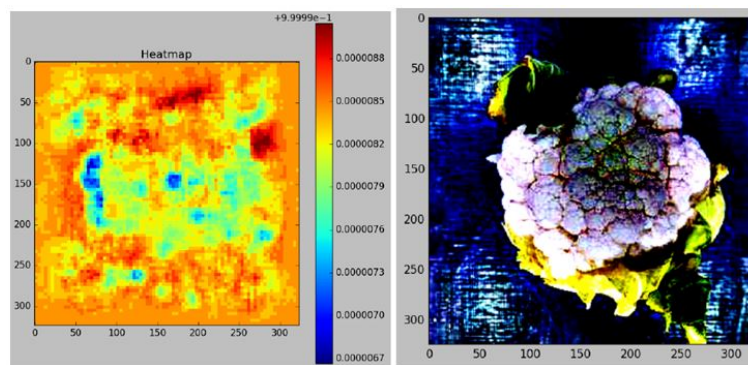
Een opmerking hierbij is dat dit systeem gebruik maakt van afbeeldingen in een gecontroleerde omgeving, wat de taak een stuk triviale maakt dan wanneer er een sterke variatie in de achtergrond is (bv. op het veld). In dit project rond automatisering van bloemkoolverwerking is dit echter de bedoeling vermits het gaat om classificatie van bloemkolen op een transportband.

Zoals in hoofdstuk 3 aan bod kwam, bestaan er diverse classificatiesystemen op basis van computervisie in de context van preciselandbouw. Het is echter moeilijk om deze resultaten te vergelijken aangezien deze sterk afhankelijk zijn van het gewas, de metrieken, de variatie in achtergrond, de beschikbare databank etc. Een eventuele vergelijking kan gemaakt worden met het onderzoek van Sabanci et al. [72] naar classificatie van granen (brood-durum), waar ook een binaire classificatie werd uitgevoerd op basis van foto's in gecontroleerde omgeving. De nauw-

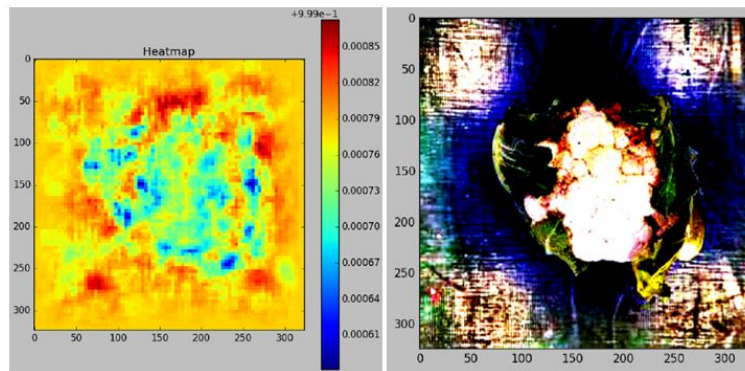
keurigheid ligt in dezelfde lijn (97.4-99.9%). Een belangrijk verschil is dat er gebruik gemaakt werd van machine learning met een ANN, waarbij heel wat feature engineering en dus menselijke interventie aan te pas kwam. Door het gebruik van deep learning (en bijgevolg automatische feature extractie) in de vooropgestelde methode voor classificatie van bloemkolen, wordt deze grootste bottleneck vermeden.



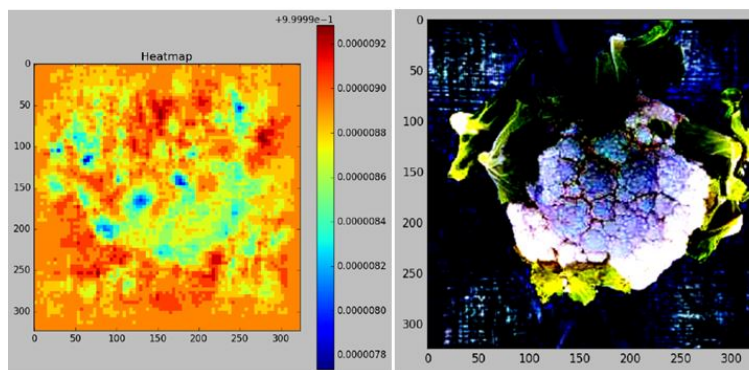
Figuur 4.14: Confusiematrix classificatie van bloemkolen. Links: absolute - Rechts: relatieve.



Figuur 4.15: Heatmap bloemkool 'kool boven' 1.

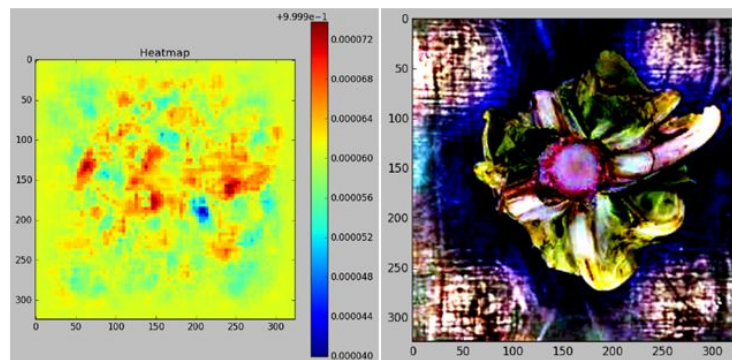


Figuur 4.16: Heatmap bloemkool 'kool boven' 2.

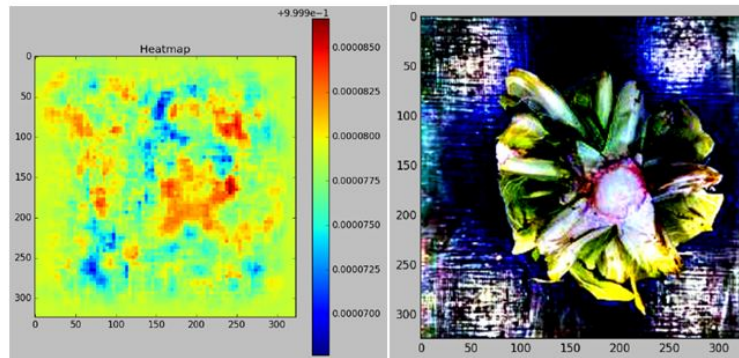


Figuur 4.17: Heatmap bloemkool 'kool boven' 3.

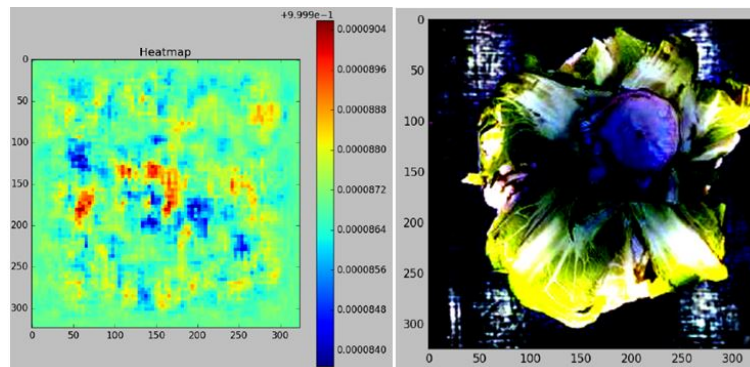
Voor de verdere interpretatie van de resultaten werden enkele heatmaps geproduceerd, analoog als bij voorgaand project (occlusie). De heatmap zelf geeft met behulp van de rode kleurtinten de minst belangrijke gedeelten weer, terwijl de blauwe tinten door het neurale net als meer interessant beschouwd worden. Enkele van deze heatmaps zijn terug te vinden in figuren 4.15-4.20, waarbij links de heatmap en rechts de originele foto na normalisatie getoond wordt.



Figuur 4.18: Heatmap bloemkool 'hart omhoog' 1.



Figuur 4.19: Heatmap bloemkool 'hart omhoog' 2.



Figuur 4.20: Heatmap bloemkool 'hart omhoog' 3.

Het netwerk lijkt meer belang te hechten aan de hoeveelheid wit (de kool) en bijhorende structuur op de de afbeelding van een correct liggende kool. Opmerkelijk is dat het netwerk geen sterke focus legt op een opvallende kenmerk van een omgekeerd liggende kool: het hart van de kool. Het netwerk kijkt daarentegen naar de omliggende pixels en lijkt op zoek te gaan naar de koolstructuur om op basis daarvan te bepalen in welke klasse de afbeelding ingedeeld moet worden.

5

Semi-supervised detectie van planten

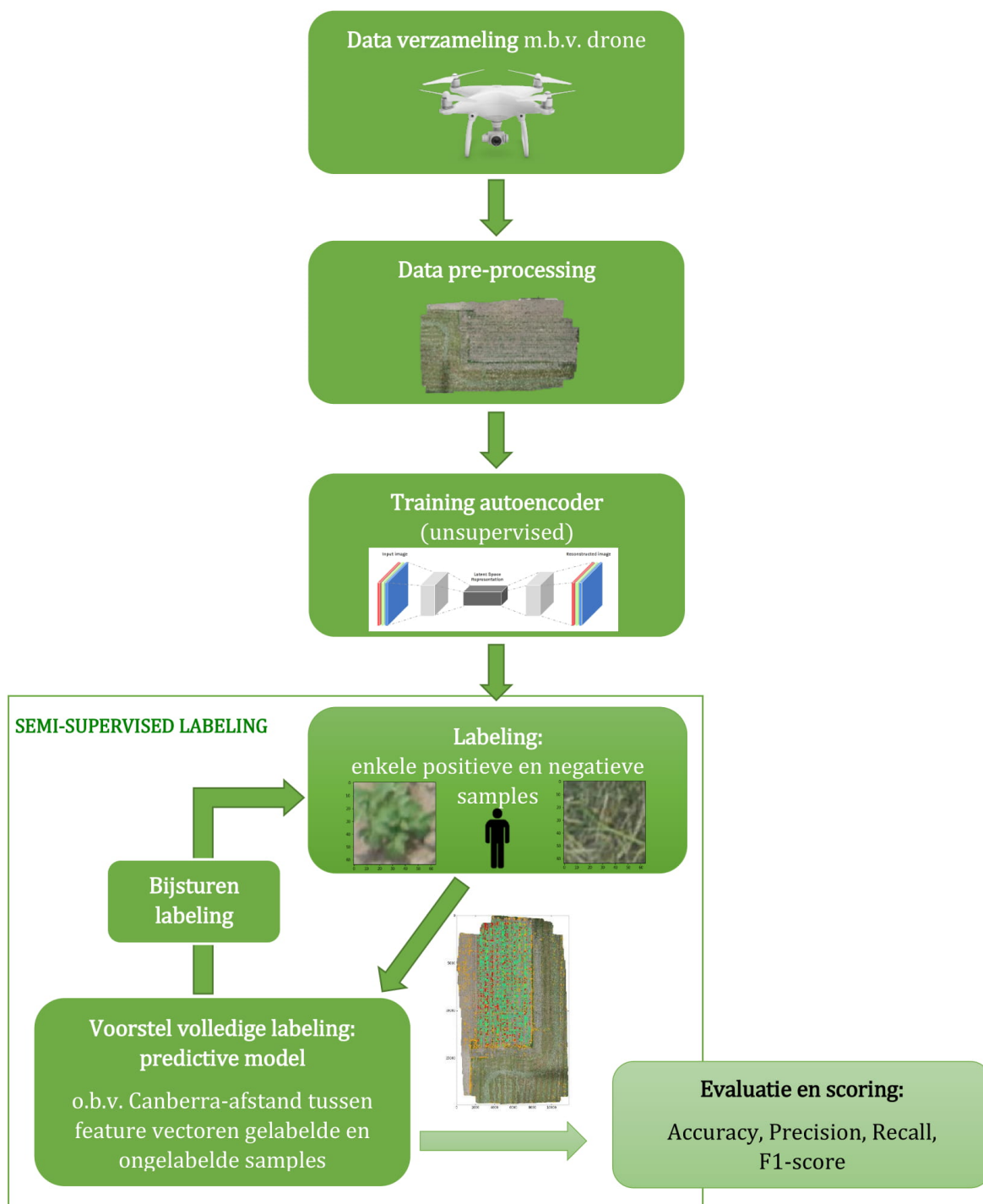
Zoals in hoofdstuk 3 aangehaald werd, stijgt de nood aan intelligente systemen die kunnen omgaan met de variabiliteit van de natuurlijke omgeving en teeltoomstandigheden. De variatie in soorten en omgevingen vormt immers een belangrijk obstakel in de ontwikkeling van robuuste systemen die functioneel kunnen worden ingezet in de gewasproductie. In dit hoofdstuk wordt de uitwerking van een proof of concept besproken, waarbij als doel vooropgesteld werd om een zo'n algemeen mogelijk systeem te creëren. De ontwikkeling van een algemene methode biedt namelijk de mogelijkheid om in de toekomst ingezet te worden in diverse toepassingen en in diverse omstandigheden.

Daarnaast kwam in hoofdstuk 3 ook aan bod dat het labelen van data een sterke bottleneck is. In tal van huidige projecten van computervisie in de precisielandbouw wordt hier veel tijd mee verloren. Dankzij instrumenten zoals UAV's en UGV's uitgerust met diverse sensoren, wordt data alsmaar vlotter beschikbaar. In schril contrast staat echter de benodigde tijd en bijhorende kost om die data klaar te maken (o.a. labelen) voor diverse toepassingen. De vraag die hierbij rijst is of het mogelijk is om de verzamelde data op een andere manier in te zetten voor de ontwikkeling van toepassingen, zonder de tijdrovende stap van volledige labeling te doorlopen. Gelabelde data is immers duur in vergelijking met ongelabelde. In deze proof of concept wordt gewerkt met een semi-supervised aanpak, waarbij getracht wordt om met zo weinig mogelijk labeling alsnog goede resultaten te bekomen.

Bovendien ontstond de laatste jaren op vlak van beeldverwerking een belangrijke evolutie: deep learning werd geïntroduceerd [9]. Het gebruik van deep learning kan ook bijdragen aan de algemene doelstelling van het limiteren van de menselijke interventies. In veel traditionele aanpakken van zo'n computervisie problemen wordt immers nog een handvol feature engineering gebruikt zoals geïllustreerd werd in hoofdstuk 3.

De vereisten die bijgevolg voorop gesteld werden bij de ontwikkeling van deze proof of concept zijn de overdraagbaarheid en flexibiliteit van het model in diverse contexten (o.a. diverse toepassingen, diverse teeltomstandigheden en gewassen) en de beperking van het aantal menselijke interventies op vlak van labeling en feature engineering.

De methode werd opgebouwd aan de hand van een overzichtsfoto van een aardappelveld, waarbij gefocust werd op de detectie van aardappelplanten in het geheel. Deze proof of concept kan in de toekomst echter verder uitgebouwd worden voor een divers aantal toepassingen: onkruiddetectie, detectie van rijp fruit voor een plukrobot, ziektedetectie bij planten, tellingen van het aantal opgekomen planten, voorspellingen van opbrengst etc.



Figuur 5.1: Flowchart van de nieuwe methode.

5.1 Ontwerp en implementatie

In deze sectie wordt de focus gelegd op het ontwerp en de implementatie van de architectuur van de nieuwe methode. Omwille van de beperkte tijd waarin dit onderzoek kaderde, werd besloten om de eenvoud van het model centraal te zetten. Daar waar tijd ruimte bracht, werden op sommige vlakken enkele diverse mogelijkheden geëxploreerd die bepaalde beslissingen in het ontwerpproces gestuurd hebben. Bovendien werd nagedacht over mogelijke optimalisaties die aan bod komen in Sectie 6.2.

De methode doorloopt verschillende stappen zoals geïllustreerd in de flowchart in figuur 5.1. De start van deze methode is het nemen van een hoge resolutie RGB-afbeelding met behulp van een drone, waarna de afbeelding doorheen een beeldanalyse pipeline gestuurd wordt, om tenslotte te eindigen met een (mogelijks bij te sturen) voorstel voor de indeling tussen het gewenste en dus gedetecteerde ten opzichte van de achtergrond. Het ontwerp en de ontwikkeling van de verschillende stappen uit deze pipeline worden toegelicht in volgende secties.

5.1.1 Dataverzameling en pre-processing

Alvorens een detectie van planten kan gebeuren, dient er data verzameld te worden. Zoals in sectie 3.2.2 werd aangehaald, bestaan er diverse methoden om data te verzamelen. In dit geval wordt gebruik gemaakt van beelden genomen met behulp van een UAV. Omdat het volledige veld niet als één beeld gefotografeerd kan worden met voldoende resolutie, wordt gebruik gemaakt van het softwareprogramma Agisoft [73] om de verschillende deelfoto's aan elkaar te plakken, zodoende een overzichtsfoto te krijgen.

De overzichtsfoto die bij de ontwikkeling van de methode gebruikt wordt, werd ter beschikking gesteld door het ILVO. Op deze afbeelding wordt een aardappelveld weergegeven, gefotografeerd tijdens de groeiperiode (mei 2018) van de plant waarbij een duidelijk onderscheid waar te nemen is tussen de diverse planten. De foto werd op A4-formaat toegevoegd in bijlage B.

Naast het aardappelveld centraal in de afbeelding, bevat de afbeelding ook diverse soorten ruis op vlak van natuur: een aanliggend bietenveld, een stuk grasveld, sporen van tractoren etc. Deze afbeelding is bijgevolg geen gecontroleerde, maar wel een realistische omgeving.

Alvorens de afbeelding aan de netwerkarchitectuur te geven, wordt deze genormaliseerd. Hiervoor wordt gebruik gemaakt van de beschikbare normalise-functie uit de Torchvision package [74], die een normalisatie uitvoert op basis van het gemiddelde en de standaarddeviatie. Dit zorgt ervoor dat de afbeelding in de range $[-1,1]$ geschaald wordt, zodanig dat er een meer symmetrische datadistributie ontstaat en het netwerk sneller kan convergeren.

Verdere pre-processing komt in deze proof of concept niet aan bod. Er wordt geen gebruik gemaakt van bijvoorbeeld de ruimtelijke ordening van de planten (inplanting in rijen [50]). Bij ziektedetectie staan de te detecteren planten bijvoorbeeld in dezelfde rijen als gezonde planten en bij een plukrobot is er geen sprake van rijen. Daarnaast wordt er geen gebruik gemaakt van plant/achtergrond scheiding en vaak bijhorende groepering van pixels tot grotere basiseenheden, zoals in de meeste werken uit de literatuurstudie aan bod kwam.

De pre-processing is bijgevolg tot een minimum gereduceerd, waarbij de focus lag op het vermijden van menselijke interventie en het stimuleren van de overdraagbaarheid van de methode. In de toekomst kan op dit gebied alsnog ingezet worden op een geautomatiseerd systeem voor pre-processing (bv. scheiding plant/achtergrond, groepering van pixels), wat het systeem mogelijk robuuster zal maken.

5.1.2 Netwerkarchitectuur: autoencoder

Wanneer buiten het domein van precisielandbouw gekeken wordt, bewees de autoencoder reeds zijn nut bij onder meer anomalie detectie [75]. In toepassingen omtrent anomalie detectie heeft men als doel om de afwijkende elementen op te sporen en is er bijgevolg sprake van een sterk ongebalanceerde dataset. Enkele voorbeelden hiervan zijn terug te vinden in het werk van Borghesi et al. [76], dat gebruik maakt van een autoencoder voor afwijkingen te detecteren in een high performance computing-systeem, Bhattad et al. [77] om afwijkende gezichten te herkennen en Baur et al. [78] voor afwijkingen in de hersenen op basis van MR-beelden.

Een veelvoorkomende manier waarop de autoencoder gebruikt wordt in anomalie detectie, is door het te trainen op een dataset zonder afwijkingen. Wanneer vervolgens in de testing een anomalie aan bod komt, zal de autoencoder veel meer moeite hebben met de reconstructie van die afbeelding. Dit resulteert in een veel hogere reconstructie error en op die manier kan de anomalie gedetecteerd worden [75].

Door de detectie van planten (en meer specifiek onkruid of zieke planten) te bekijken vanuit deze invalshoek, wordt in deze methode gebruik gemaakt van een autoencoder. In de toepassingen uit hoofdstuk 4 werd gebruik gemaakt van een supervised CNN, waarbij het netwerk een verzameling features van de afbeelding probeert te maken en deze features supervised worden bijgestuurd. In tegenstelling tot dit systeem gaat de autoencoder trachten om de feature vector (ook wel latente representatie genoemd) terug om te zetten naar een gereconstrueerde afbeelding, om van de verschillen tussen het originele en het gereconstrueerde beeld te leren.

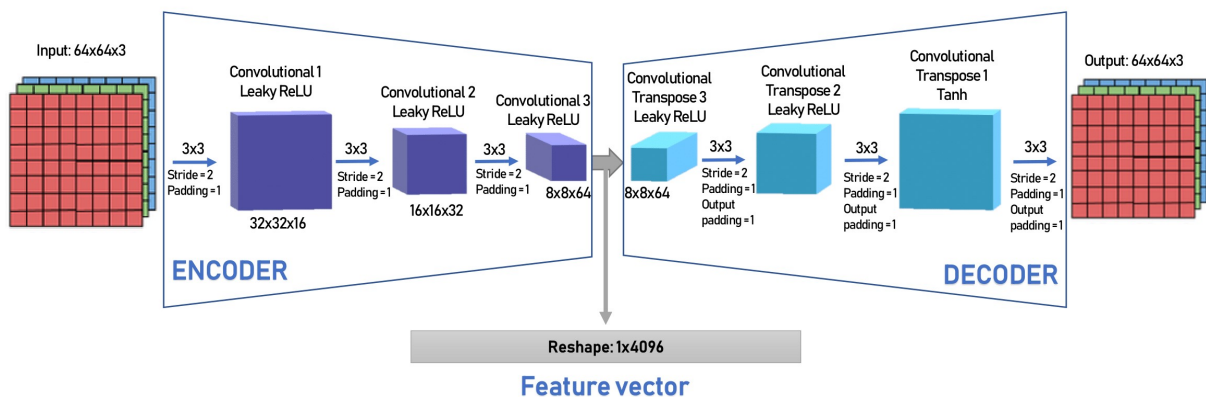
Bovendien gaat de autoencoder zelf interessante features gaan selecteren, wat veel efficiënter is dan de handmatige bepaling van features zoals in veel werken uit de literatuurstudie in hoofdstuk 3 het geval bleek te zijn. Op die manier kan een autoencoder zaken leren met minimale menselijke

interventie. Dit wordt 'representation learning' genoemd [63].

De data waarop de autoencoder in dit geval getraind wordt, namelijk de overzichtsfoto, bevat echter zowel de normale achtergrond als de te detecteren planten. De planten worden verder benoemd als positief, vermits het doel is om deze te detecteren, terwijl de achtergrond negatief genoemd wordt. Omwille van deze input wordt geen gebruik gemaakt van de hierboven vernoemde techniek met reconstructie error. Daarentegen maakt deze methode gebruik van de correlatie tussen de diverse latente representaties of feature vectoren. De autoencoder wordt unsupervised getraind op basis van een random selectie van stukjes met grootte 64x64 pixels uit de afbeelding. Deze stukjes worden samples genoemd en worden bij de verwerking op willekeurige wijze horizontaal en verticaal gespiegeld om op die manier de training dataset uit te breiden (meer variatie).

De keuze voor de afmeting van 64 pixels is bepaald op basis van een trade-off tussen de gemiddelde grote van de aardappelplant en de diverse elementen op de achtergrond (o.a. andere soorten planten). Een te kleine afmeting zou niet volstaan om een gemiddelde plant te omvatten en een te grote afmeting omvat het risico dat de plant en een groot stuk achtergrond in eenzelfde venster vallen.

De gebruikte autoencoder wordt geïllustreerd in figuur 5.2 en bestaat uit een uit drie convoluntionele lagen in de encoder, gevolgd door een gespiegeld model in de decoder. Er worden geen fully connected lagen gebruikt, want deze bleken een minder resultaat op te leveren.



Figuur 5.2: Netwerkarchitectuur autoencoder.

Zoals in de figuur zichtbaar is, bestaat de latente representatie uit een 8x8x64 structuur, die verderop gebruikt wordt als feature vector, omgezet naar een 1D structuur van 4096 waarden. Als alternatief werd global average pooling (GAP) uitgetoetst. Hierbij wordt een dimensie reductie uitgevoerd, waarbij de 8x8x64 structuur gereduceerd wordt naar een 1x1x64. Elke 8x8 feature map werd hierbij tot één nummer herleid door het gemiddelde te nemen over al deze waarden. De uiteindelijke feature vector bleek te weinig informatie te bevatten om een grondige

scheiding tussen positieve en negatieve samples uit te voeren. Dit kan verklaard worden door het feit dat in heel wat samples zowel aarde als plant aan bod komen, waardoor het uitmiddelen van de waarden een sterk gelijkmatige distributie levert. Er werd hier dan ook niet mee verdergegaan.

Enkele parameters van de training van het uiteindelijke model worden weergegeven in tabel 5.1. Deze waarden werden bepaald op basis van een trial and error en verkenning van het systeem.

Tabel 5.1: Hyperparameters autoencoder.

Hyperparameter	waarde
Aantal epochs	60
Adam optimizer met L2 regularisatie	1e-5
Learning rate	0.001
Variatie learning rate	x0,1 elke 20 epochs
Loss functie	MSE

5.1.3 Semi-supervised labeling

Om de tijd- en kostenrovende procedure van het volledig labelen van de afbeelding te vermijden, wordt gebruik gemaakt van een beperkte subset gelabelde deeltjes van de afbeelding. De delen van de afbeelding die alsnog gelabeld worden, worden niet per pixel gelabeld, maar wel per sample zoals in bovenstaande sectie de afbeelding ook per sample verwerkt werd.

De totale afbeelding bevat een grote variabiliteit, de geselecteerde subset dient bijgevolg samples te bevatten die deze zoveel mogelijk dekt en bijgevolg naar een maximale visuele entropie streeft. De selectie van zo'n representatieve set van samples is bijgevolg een uitdagend probleem. Bij het ontwerp van deze proof of concept wordt de bepaling van de subset handmatig bijgestuurd op basis van trial and error.

Eens de gelabelde samples bepaald zijn, worden deze door de getrainde encoder gestuurd ter bepaling van de bijhorende feature vector. Zo ontstaat een verzameling feature vectoren behorende bij de positieve samples, alsook een verzameling behorende bij de negatieve samples.

Daarna wordt de totale afbeelding, verdeeld in een raster met vakjes van 64x64 pixels, vakje per vakje door de getrainde encoder gestuurd om de feature vector (4096x1) te bepalen.

Om te bepalen of het vakje tot de positieve of negatieve klasse behoort, wordt gebruik gemaakt van de Canberra afstand (Bill Williams en Godfrey Lance, 1966) [79]. Deze multidimensionale afstand geeft de mate van (on)gelijkheid tussen twee groepen data, in dit geval tussen feature vectoren u en v . De formule die hiervoor gebruikt wordt, wordt weergegeven in 5.1. De Canberra afstand werd reeds in diverse werken [80, 81] in de context van intrusion detection gebruikt om

de afwijkende waarden te detecteren.

$$d(u, v) = \sum_{i=1}^k \frac{|u_i - v_i|}{|u_i| + |v_i|} \quad (5.1)$$

Voor elk vakje uit het raster wordt bijgevolg de mate van gelijkheid met alle positieve en alle negatieve samples bepaald, om vervolgens de minimale waarde en dus grootste gelijkheid te bepalen. Op basis hiervan wordt het vakje vervolgens geclassificeerd als positief (in dit geval aardappelplant) of negatief (geen aardappelplant).

Het doel van deze semi-supervised aanpak is dus om in de toekomst gebruik te maken van de expertise van de landbouwer die van deze methode gebruik maakt. De labeling van enkele samples kan immers gebeuren door de gebruiker van de toepassing, waarbij de landbouwer dus gevraagd wordt om enkele 'positieve samples' (= dingen waarin hij geïnteresseerd is) en enkele 'negatieve samples' (= dingen waar hij niet in geïnteresseerd is) aan te duiden op de foto. Deze samples worden vervolgens gebruikt als gelabelde dataset voor de bepaling van de categorie (positief/negatief) van alle andere samples in de afbeelding. Vervolgens wordt een resultaat getoond van de classificatie, waarbij de landbouwer de kans krijgt dit resultaat bij te sturen door enkele extra samples toe te voegen aan de subset. Dit proces kan doorgaan tot de landbouwer het resultaat bevredigend vindt of er geen verbetering meer plaatsvindt. Het netwerk zelf hoeft niet telkens opnieuw getraind te worden, vermits de classificatie volledig gebeurt na dit trainingsproces.

In de context van ziektedetectie kan de landbouwer bijvoorbeeld gebruik maken van enkele gekende zieke planten om op die manier eventuele andere zieke planten te detecteren in zijn veld.

Enkele alternatieven

Selectie van de gelabelde samples: Zoals aangehaald is het belangrijk om een subset te labelen die voldoende representatief is. Op dit ogenblik blijkt de trial and error bijgestuurde selectie van samples tot de beste resultaten te leiden. Hiervoor werden zo'n 100 samples per categorie gelabeld, waarbij op basis van een random generator alsook visuele evaluatie tussendoor tot de uiteindelijke subset gekomen werd.

Een andere idee die tijdens het onderzoek naar boven kwam, betreft het gebruik van de reconstructie error. De volledige afbeelding wordt hierbij vakje per vakje door de autoencoder gestuurd. De vakjes die de grootste reconstructie error kennen zijn bijgevolg diegene waar het netwerk het meest moeite mee heeft. Op basis hiervan werd een heatmap gemaakt met kleuring

naar stijgend belang. Het labelen van een bepaald aantal meest belangrijk vakjes zou echter niet altijd een goed resultaat opleveren, vermits bepaalde onverwachte elementen (bv. een gieter) in de afbeelding naar boven zullen komen, terwijl deze niet relevant zijn voor de labeling. Deze methode wordt bijgevolg niet gebruikt.

In de toekomst wordt gestreefd naar een methode waarbij de gebruiker enkele samples te zien krijgt waarbij gevraagd wordt deze te labelen, in tegenstelling tot nu, waarbij de gebruiker zelf samples moet kiezen.

Bepaling categorie voor alle vakjes: Om de afstand tussen twee eendimensionale vectoren te bepalen, kunnen diverse soorten afstanden gebruikt worden. Oorspronkelijk werd gebruik gemaakt van de Euclidische afstand tussen twee vectoren in plaats van de Canberra afstand. Bij de Euclidische afstand wordt de afstand bepaald tussen twee punten (in rechte lijn), waarbij de punten in een meerdimensionale (in dit geval 4096) ruimte geplaatst worden. De Euclidische afstand hecht evenveel belang aan elke dimensie, wat bij deze methode resulteerde in mindere resultaten dan de gewogen Canberra afstand. De resultaten lagen gemiddeld zo'n 5% lager wat betreft F1-waarde (zie verder). Ook de Bray-Curtis, Minowski en City Block afstanden werden uitgetoetst, maar gaven mindere resultaten dan de gekozen Canberra afstand.

Als alternatief voor de bepaling op basis van een afstand tussen twee vectoren, kan de indeling gebeuren met behulp van een classifier. Een classifier (o.a. Random Forest, MLP, KNN, Naïeve Bayes) wordt hierbij getraind op basis van een set gelabelde data ingedeeld in twee categorieën (positief-negatief). Vervolgens worden de feature vectoren van alle vakjes uit de afbeelding gebruikt als testdataset. De categorie voor elk vakje wordt bijgevolg bepaald met behulp van deze classifier. Dit resulteerde echter niet in goede resultaten, wat ergens te verwachten was door de erg kleine hoeveelheid gelabelde trainingsdata. Ook het idee van hier semi-supervised (self-training) verder te werken met de classifier werd uitgetoetst, maar leverde geen goede resultaten. Hierbij worden na het trainen en testen op basis van de gelabelde dataset, de meest betrouwbare testresultaten toegevoegd aan de trainingsdataset om vervolgens opnieuw te trainen. Het gevaar bij deze methode is immers dat fouten die in het begin worden meegenomen telkens versterkt worden.

Tenslotte werd ook getracht om de meerdimensionale data in de feature vectoren grafisch voor te stellen met behulp van Principal Component Analysis (PCA) en t-distributed Stochastic Neighbor Embedding (t-SNE), twee technieken die de dimensies trachten te reduceren. PCA is een wiskundige techniek die gebruik maakt van de correlatie tussen de dimensies en tracht met een minimaal aantal variabelen een maximale variatie te behouden van de originele datadistributie. t-SNE daarentegen is een probabilistische methode die tracht om de originele data te representeren met minder dimensies en de divergentie tussen beide representaties te minimaliseren [82]. Deze beide technieken werden gebruikt om de gelabelde samples voor te stellen. Er werden echter geen goede resultaten bekomen om de 4096-dimensionale feature vectoren voor te stellen

in de 3D visuele ruimte. De reductie van 4096 naar 3 dimensies is in dit geval een te grote stap.

Indien een visuele voorstelling in een 4096-dimensionale ruimte mogelijk zou zijn, wordt verwacht dat de positieve samples een cluster zullen vormen, terwijl de negatieve samples eerder gespreid liggen omwille van de diversiteit. De keuze voor het gebruik van minimale afstanden tot de samples (dichtstbijzijnde buur) in plaats van een gemiddelde afstand wordt hiermee aangetoond. Een gemiddelde waarde van de cluster negatieve samples zou de grote diversiteit in achtergrondelementen immers uitmiddelen. De gebruikte aanpak om met behulp van de minimale afstand tot één van de gelabelde samples hun feature vector is in deze aanpak bovendien mogelijk omdat er geen te grote aantallen gelabelde samples zijn.

5.1.4 Evaluatie van het model

Om het model te evalueren is het noodzakelijk een zekere referentie te hebben van wat effectief aardappelplanten zijn en wat niet. Er wordt hiervoor gebruik gemaakt van een 'ground truth', een 64x64 raster dat over de afbeelding past en voor elk vakje aangeeft of het een aardappelplant is of niet.

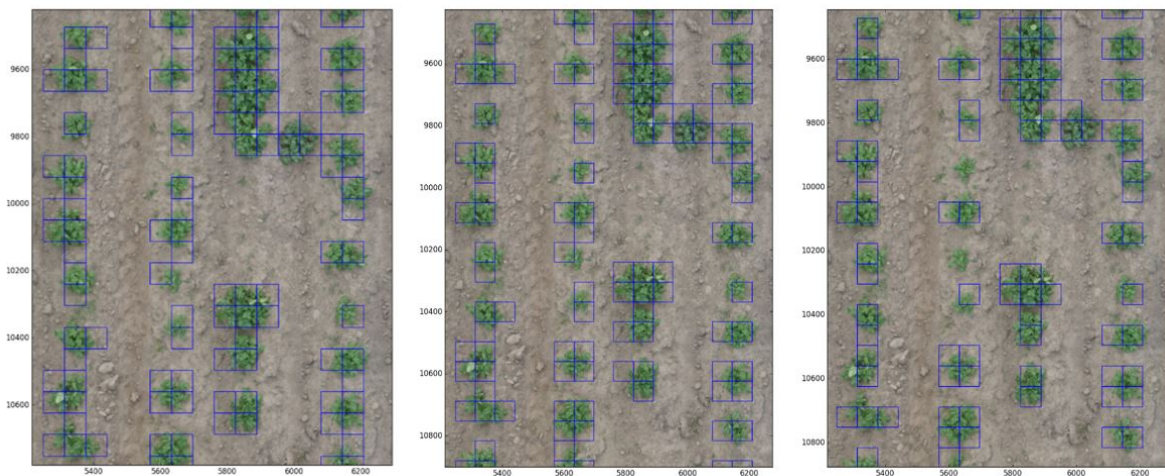
Op de originele afbeelding worden er eerst en vooral selectiekaders getekend rondom de aardappelplanten. Vervolgens worden de vakjes van het 64x64 raster geselecteerd waarmee de selectiekaders overlappen. Indien meer dan 35% van het vakje in het raster bedekt wordt door een (deel van een) selectiekader, wordt het bijhorende vakje aangeduid als aardappelplant.

Over de kwaliteit van deze ground truth zijn twee belangrijke opmerkingen te maken. Enerzijds werden de selectiekaders niet getekend door een expert, waardoor er mogelijk bepaalde moeilijke zichtbare planten niet volledig gelabeld werden. Hiervoor werd wel een veelvoudige manuele controle van de totale labeling uitgevoerd.

Daarnaast zijn planten niet zomaar rechthoekig, waardoor binnen het selectiekader soms ook een stuk achtergrond bevat. Dit kan resulteren in vakjes die als plant worden gelabeld, maar eigenlijk voor het grootste deel uit achtergrond bestaan.

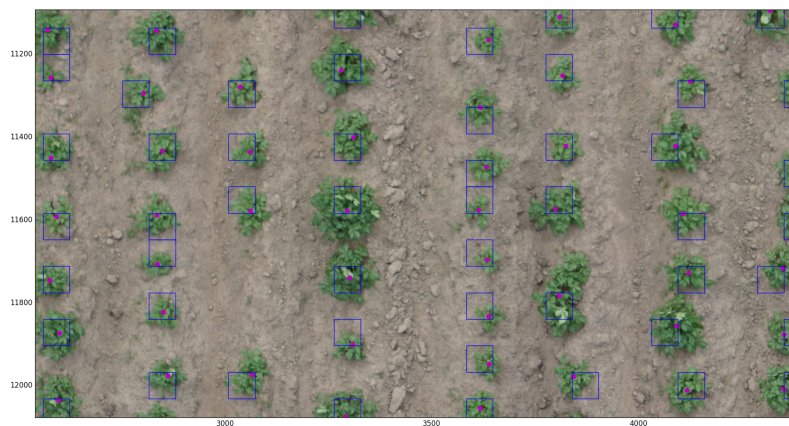
De drempelwaarde van 35% speelt hierin een belangrijke rol. Uit trial and error en visuele evaluatie van de ground truth, wordt de drempel op 35% ingezet. Een hogere drempel resulteert in het niet detecteren van eerder kleine aardappelplanten, zoals rechts in figuur 5.3 getoond wordt. Deze lagere drempel gaat daarentegen sommige vakjes met vrij veel achtergrond en een klein stukje plant als plant labelen, zoals links in figuur 5.3 zichtbaar is. De vakjes die als plant gecategoriseerd worden, zijn hier met blauwe rechthoekjes getekend.

Een alternatieve labeling die bij aanvang van de ontwikkeling van de methode gebruikt werd, was een labeling waarbij elke plant met een punt gemarkeerd werd. Het nadeel hiervan is dat er per



Figuur 5.3: Links: ground truth met drempel = 30%. Midden: ground truth met drempel = 35%. Rechts: ground truth met drempel = 40%.

plant vervolgens maar één vakje in het raster als plant gelabeld werd, wat een grote foutmarge met zich meebracht. Dit wordt geïllustreerd in figuur 5.4



Figuur 5.4: Gelabelde punten (fuchsia) en bijhorend gelabeld vakje in raster van 64x64 pixels.

Aangezien het maken van deze ground truth enkel nodig is voor het onderzoek naar de kwaliteit van de ontwikkelde methode, dus niet voor het gebruik van de methode zelf en de beperkte tijd van dit onderzoek, werd besloten om met deze foutmarge verder te werken en hier geen verdere tijd aan te spenderen.

De gebruikte metrieken ter evaluatie van het model, zijn de in hoofdstuk 3 aangehaalde accuracy, precision, recall en F1-score.

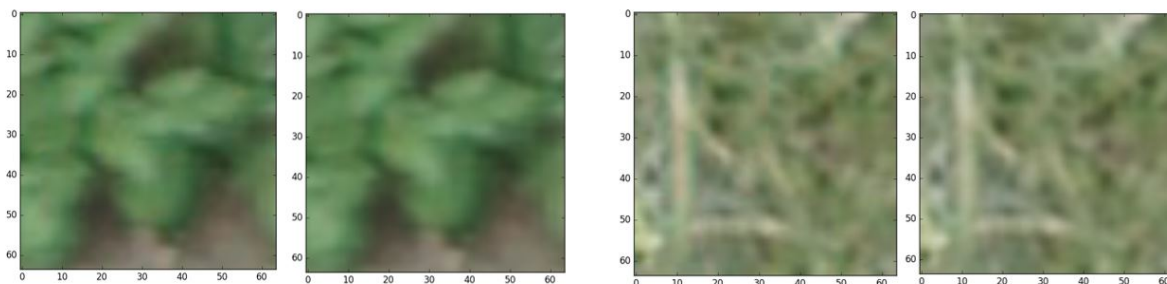
5.1.5 Gebruikte technologieën

De architectuur die in deze sectie beschreven werd, werd geïmplementeerd in Python 3.5. Er werd gebruik gemaakt van PyTorch [83], een open-source deep learning platform, aangevuld met enkele functionaliteiten uit de TensorFlow [64], een ander open-source platform voor machine learning. De modellen werden getraind op een NVIDIA GeForce GTX 960M GPU.

5.2 Resultaten

5.2.1 Netwerkarchitectuur

Op niveau van de autoencoder kan het resultaat van de werking getoond worden aan de hand van enkele reconstructies. Het doel van dit gedeelte is immers om een stukje van de afbeelding correct te kunnen reconstrueren, waardoor na de encoder een latente representatie ontstaat met gegevens die het stukje van de afbeelding representeren. In figuur 5.5 worden twee voorbeelden gegeven van deze reconstructies, met links in het voorbeeld de originele sample en rechts de reconstructie. Er kan vastgesteld worden dat de autoencoder in staat is om de afbeeldingen correct te reconstrueren na een reductie tot 4096 features.



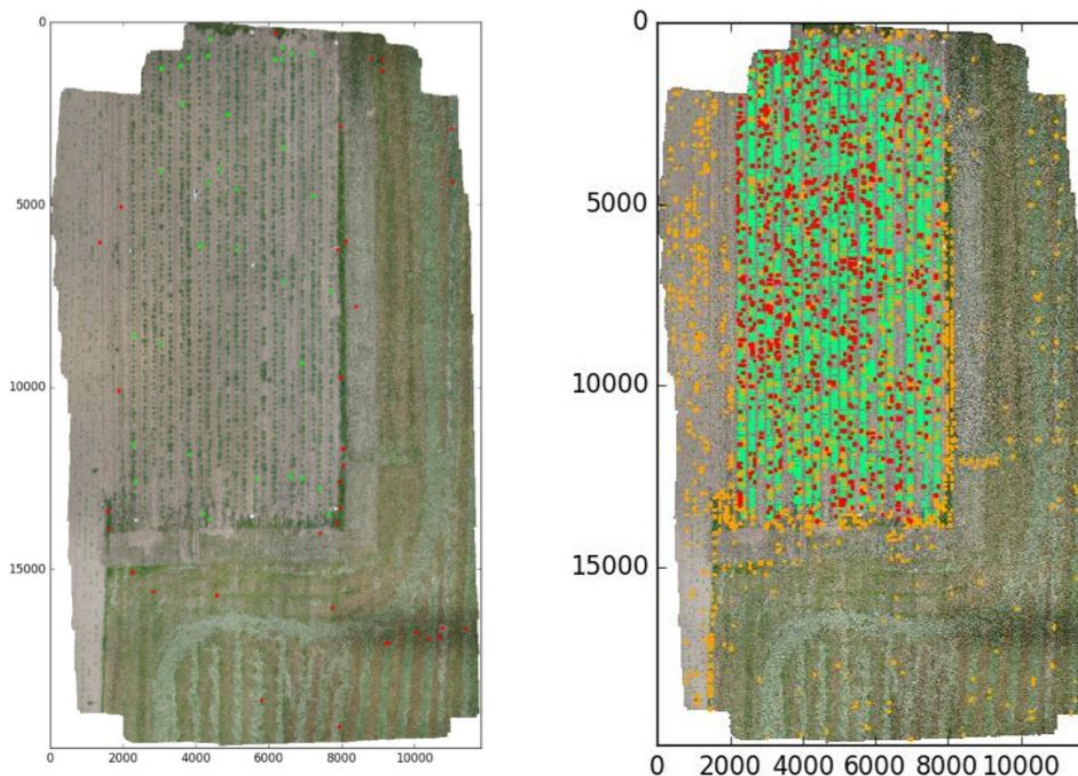
Figuur 5.5: Links: voorbeeld reconstructie positieve sample. Rechts: voorbeeld reconstructie negatieve sample.

5.2.2 Evaluatie van de proof of concept

De resultaten voor diverse metrieken bij de effectieve uitwerking van de proof of concept van de hierboven beschreven methode worden weergegeven in tabel 5.2. De vrij hoge accuracy moet hierbij genuanceerd worden omwille van de ongebalanceerde dataset. Zoals aangehaald in hoofdstuk 3 zorgt dit ervoor dat zelfs in het extreme geval dat alles negatief gelabeld wordt, er nog steeds een accuracy van 90,2% bereikt wordt (de ground truth bevat 56730 samples waarvan er

51202 negatief en 5528 positief zijn). Daarom wordt in verdere evaluatie voornamelijk gefocust op de F1-score die een harmonische uitmiddeling is van de precision en recall.

Deze resultaten werden bekomen bij het gebruik van 35 samples per soort, waarvan de keuze met behulp van trial and error bijgestuurd werd. Links in figuur 5.6 worden de gekozen samples voor labeling weergegeven op de totale afbeelding, waarbij de groene vakjes de positieve samples voorstellen en de rode vakjes de negatieve. Rechts in deze figuur staat een voorstelling van het eindresultaat, waarbij de groene vakjes de 'true positives (TP)' (juist gedetecteerde planten), de rode vakjes de 'false negatives (FN)' (niet gedetecteerde planten) en oranje vakjes de 'false positives (FP)' (niet plant die gedetecteerd werd als plant) voorstellen. Alle overige niet gekleurde vakjes zijn de 'true negatives (TN)'. Diverse FN zijn afkomstig door de vermelde fout in de labeling. De aanwezige false positives moeten in toekomstig onderzoek verder gereduceerd worden.



Figuur 5.6: Links: gekozen positieve (groen) en negatieve (rood) samples. Rechts: eindresultaat. Rode: FN, oranje: FN, groene: TP.

Deze resultaten tonen aan dat de methode 96,03% van de 64x64 vakjes uit het raster correct gaat benoemen als plant of geen plant. De waarde voor de precision geeft aan dat 75,46% van de als plant benoemde vakjes ook effectief planten zijn. Het totaal aantal vakjes die als plant gedefinieerd worden, wordt bepaald door de ground truth. De recall illustreert dat 81,03% van

Tabel 5.2: Resultaten metrieken.

Metriek	waarde
Precision = $\frac{TP}{TP+FP}$	75,46%
Recall = $\frac{TP}{TP+FN}$	81,03%
F1-score	78,15%
Accuracy	96,03%

de planten uit de ground truth ook effectief correct geïdentificeerd worden. De F1-score geeft zoals vermeld het harmonisch gemiddelde weer tussen de precision en recall, vermits het zowel belangrijk is om zoveel mogelijk planten te detecteren, als om niet teveel valse detecties te hebben. Deze waarde houdt bijgevolg rekening met zowel de FP als de FN, daar waar precision en recall slechts op één van beide focussen. In deze situatie geeft de F1-score een beter beeld van de juistheid van het systeem omwille van de ongebalanceerde verhouding tussen planten en achtergrond. Zoals eerder aangehaald dient hierbij rekening gehouden te worden dat er reeds een zekere foutmarge in de ground truth sloop door de manier van labelen.

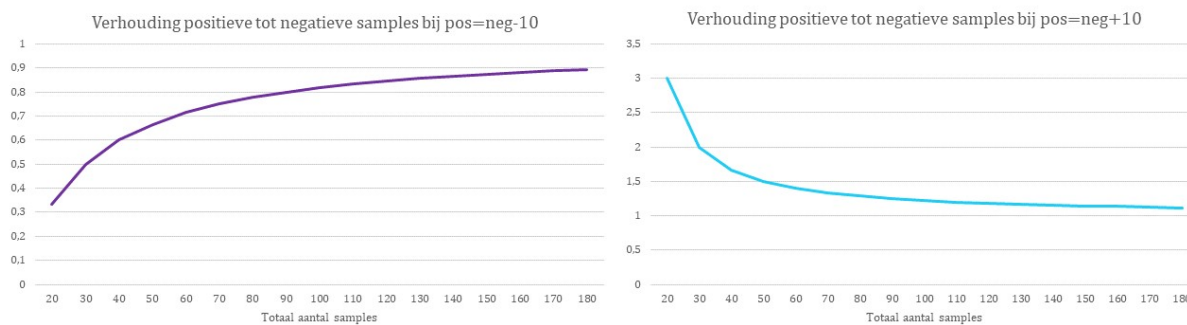
5.2.3 Evaluatie labeling subset

De bepaling van het aantal samples, de invloed van de gestuurde keuze van samples alsook de verhoudingen tussen het nodige aantal positieve en negatieve samples zijn interessante gegevens die door de toepassing van de methode bepaald kunnen worden.

Verhouding positieve/negatieve aantallen gelabelde samples

Een eerste verband dat onderzocht werd, is de verhouding tussen het aantal positief en negatief gelabelde samples voor een optimaal resultaat. In figuur 5.8 geeft de oranje lijn de resultaten van een gelijk aantal samples van deze categorie, de blauwe lijn telkens 10 meer positieve dan negatieve samples en de paarse lijn 10 minder positieve dan negatieve (elkaars inverse). Door het gebruik van een constant verschil qua aantal, evolueert de onderlinge verhouding van het aantal positieve en negatieve samples zoals geïllustreerd in figuur 5.7.

Wanneer er meer positieve dan negatieve samples aanwezig zijn (blauw) zal bij een kleine hoeveelheid samples de precision laag en recall hoog liggen. Dit is te verklaren door het feit dat er bij een beperkt totaal aantal samples slechts een klein aandeel van negatieve samples aan bod komt. In het geval van bijvoorbeeld 20 samples in totaal, zijn er 15 positieve en slechts 5 negatieve samples. Hierdoor wordt slechts een heel klein aandeel van het complete negatieve spectrum gedekt en kent het systeem vooral het positieve spectrum. De kans dat de dichtstbijzijnde buur



Figuur 5.7: Grafische weergave van de verhouding tussen het aantal positieve en negatieve samples. Links in het geval van $\text{pos}=\text{neg}-10$ en rechts $\text{pos}=\text{neg}+10$.

van een vakje zijn feature vector tot het positieve spectrum behoort, stijgt.

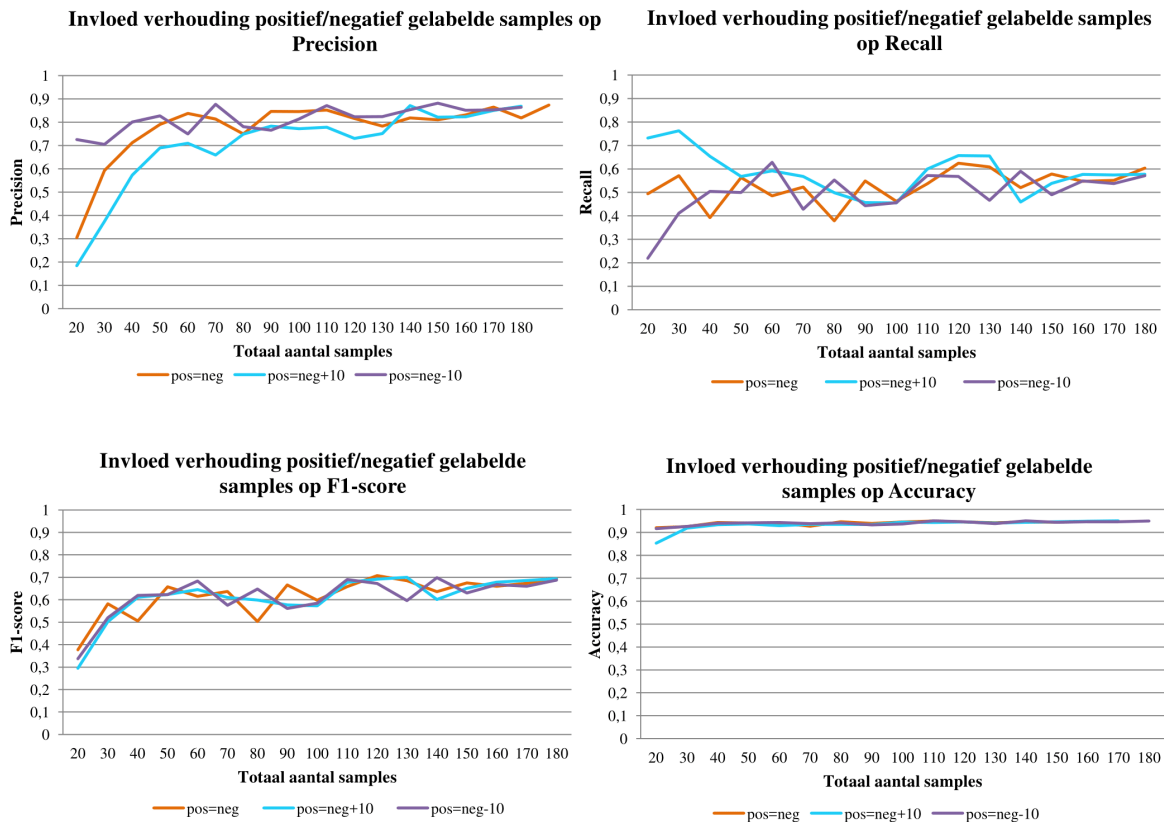
Dit uit zich in een hoog aantal FP en beperkt aantal FN. In de formule voor recall (zie tabel 5.2) is het aantal FN een belangrijke parameter in de noemer, wat de hogere recall verklaart. Analoog staat het aantal FP in de noemer bij de formule voor precision, wat dan weer de lage waarde verklaart. Naarmate de verhouding convergeert naar 1, convergeren ook de precision en recall naar de lijn met een gelijk aantal samples. De iets lagere accuracy in dit geval kan verklaard worden door de ongebalanceerde dataset zoals hierboven uitgelegd.

Het omgekeerde effect doet zich voor in de omgekeerde situatie (paarse lijn), waarbij het aantal negatieve samples in de verhouding overheerst zoals zichtbaar links in figuur 5.7.

Vermits de F1-score een harmonisch gemiddelde is van precision en recall, is het effect qua verschil in aantallen hier niet zo zeer merkbaar en lopen de drie curves vrij gelijk. Afhankelijk dus van de gewenste metriek kan bijgevolg geopteerd worden voor een hoger aantal positieve of negatieve samples mee te geven indien er slechts een kleine hoeveelheid samples gelabeld kunnen worden.

Sturing van keuze gelabelde samples

Om na te gaan wat het effect is van de keuze van samples om te labelen, wordt een vergelijking gemaakt tussen een random keuze van labels (uit 100 beschikbare labels per soort) en een gestuurde keuze. Voor deze gestuurde keuze wordt gebruik gemaakt van een set van 50 samples (25 positieve en 25 negatieve) die actief geselecteerd werden op basis van trial and error omtrent de dekking van de variabiliteit. In plaats van random uit de 100 samples te kiezen, werd eerst uit de geselecteerde set geselecteerd. Eens de grens van 50 samples voorbij, wordt de gestuurde set aangevuld met random waarden uit diezelfde overige beschikbare 100 labels. Dit resulteert



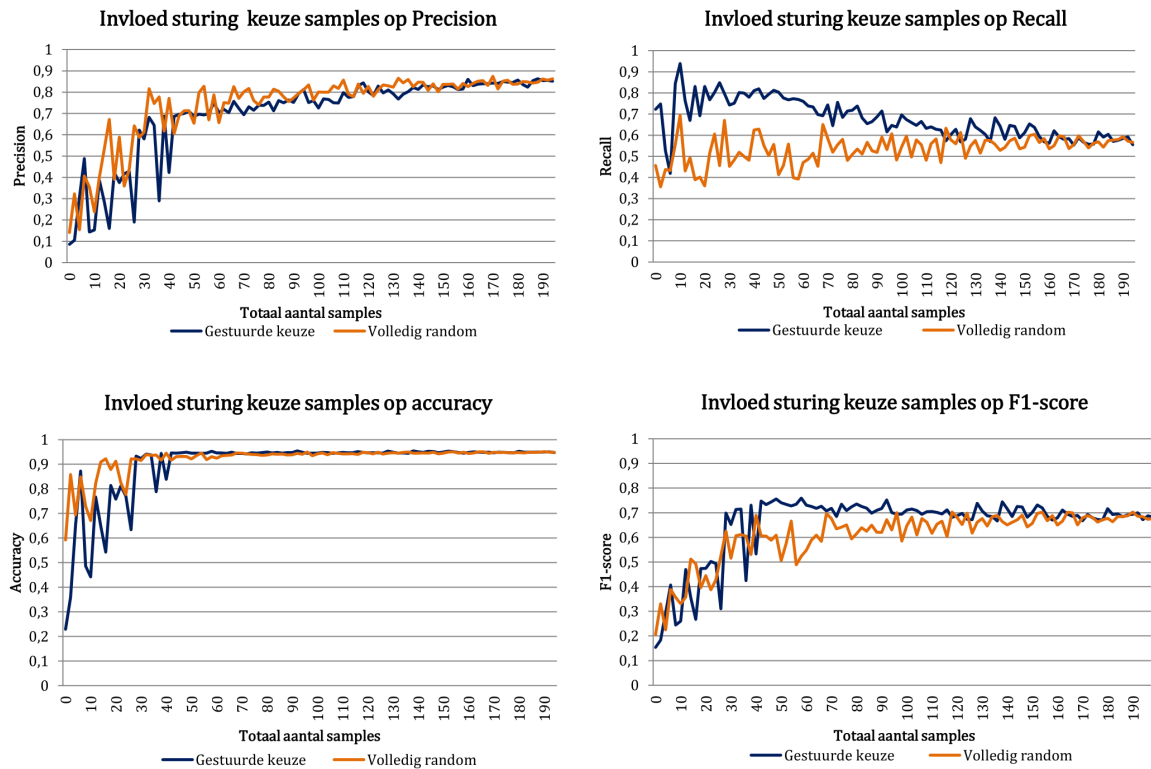
Figuur 5.8: Grafische weergave van de invloed van verhouding positief/negatief gelabelde samples op precision (linksboven), recall (rechtsboven), accuracy (linksonder) en F1-score (rechtsonder).

in de convergentie van de grafieken in figuur 5.9.

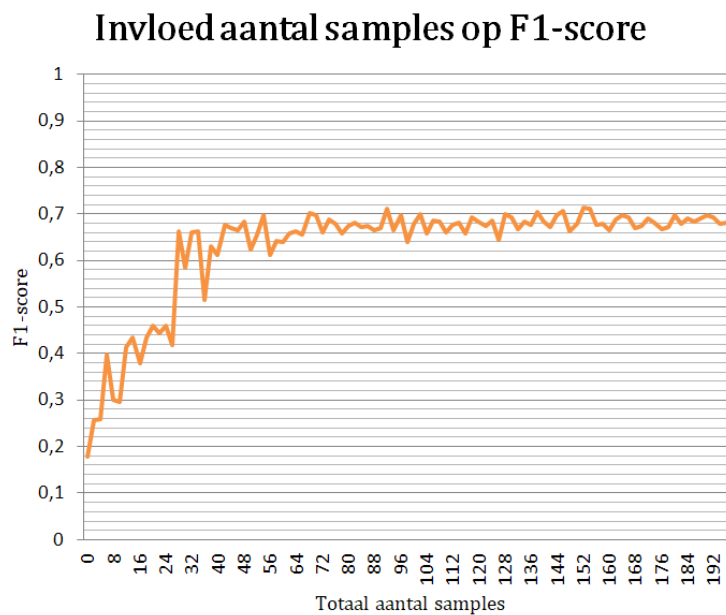
Uit de grafiek met de invloed op de F1-score blijkt dat de F1-waarde sneller tot een beter resultaat komt door een gestuurde keuze van gelabelde samples dan bij een random keuze. Een analoog effect doet zich voor bij de recall, terwijl de precision en accuracy een meer gelijkaardig patroon vertonen. Een goede selectie van samples leidt bijgevolg tot een beter resultaat met een beperkter aantal.

Hoeveelheid gelabelde samples

Een laatste onderzoek verband betreft het effect van het aantal gelabelde samples op de resultaten, zoals geïllustreerd wordt in figuur 5.10. Hiervoor werd het model beschreven in Sectie 5.1 gebruikt, waarbij het aantal gelabelde samples varieert van 1 per soort tot 100 per soort met gelijke hoeveelheid positieve en negatieve samples. De keuze van de samples gebeurde at random uit een lijst van 100 beschikbare samples. Dit experiment werd uitgemiddeld over 4 runs, om mogelijke uitschieters door de random selectie te minimaliseren.



Figuur 5.9: Grafische weergave van de invloed van een al dan niet gestuurde keuze van gelabelde samples op precision (linksboven), recall (rechtsboven), accuracy (linksonder) en F1-score (rechtsonder).



Figuur 5.10: Grafische weergave van de invloed van het aantal gelabelde samples op de F1-score.

In het begin neemt de performantie toe wanneer een groter aantal samples gebruikt wordt. Echter convergeert dit systeem naar ongeveer 68% wanneer meer dan 35 gelabelde samples per soort aangewend worden. Dit plateau suggereert dat een extra moeite qua labeling hoogstwaarschijnlijk niet zinvol is bij deze proof of concept.

5.2.4 Bijgesneden afbeelding

In de proof of concept wordt gebruik gemaakt van een volledige overzichtsfoto inclusief diverse soorten ruis. Er kan de vraag gesteld worden wat de bijdrage zou zijn van deze afbeelding bij te snijden net rondom het veld, waarbij echter nog ruis aanwezig is in de vorm van grassen rondom de planten. In een applicatie zou bijvoorbeeld in de eerste stap aan de landbouwer gevraagd kunnen worden om het gewenste gebied te selecteren.

In figuur 5.11 wordt deze bijgesneden afbeelding weergegeven. De resultaten van de methode toegepast op deze afbeelding, met slechts 24 positief en 12 negatief gelabelde samples resulteerde in een F1-score van 82,89%. Er is bijgevolg duidelijk marge om hiermee betere resultaten te behalen.

5.2.5 Test proof of concept op nieuwe afbeelding

Als aanvulling op bovenstaande resultaten, werd de methode toegepast op een andere afbeelding, namelijk een veld met spinazie. Exacte resultaten bepalen hiervoor is niet mogelijk omwille van het ontbreken van een ground truth en de keuze om geen verdere tijd aan te spenderen aan het labelingsproces.

In figuur 5.12 wordt het visuele resultaat getoond van een eerste doorloop van de methode, waarbij random 20 positieve en 20 negatieve samples gelabeld werden. De blauw gemarkeerde gebieden tonen de vakjes op de afbeelding die door de methode als positief en dus als plant gecategoriseerd werden. Wanneer in de toekomst deze methode in een applicatie geïntegreerd zou worden, kan met behulp van zo'n beeld aan de landbouwer gevraagd worden of hij de detectie al dan niet verder wenst bij te sturen.

Visueel kan hieruit besloten worden dat het systeem reeds een goede eerste benadering geeft voor de detectie van de spinazieplanten. De sporen van de tractoren zijn duidelijk zichtbaar, de middenstukken met enkel aarde, alsook het gedeelte uiterst links waar geen spinazie staat, worden niet gemarkeerd. Daarentegen worden op punten met veel planten dicht bij elkaar nog bepaalde planten niet gedetecteerd. Het resultaat kan vervolgens van hieruit bijgestuurd worden.

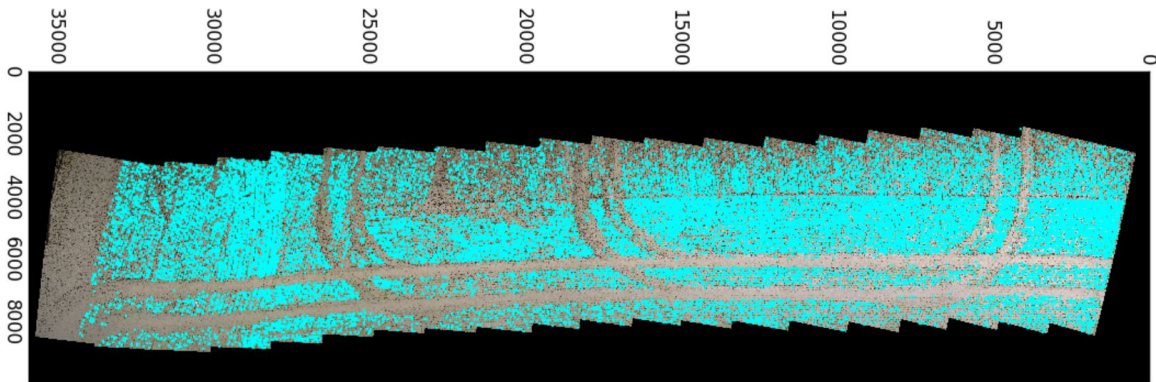


Figuur 5.11: Bijgesneden afbeelding.

5.3 Besluit

In dit hoofdstuk wordt een proof of concept voorgesteld van een nieuwe aanpak voor semi-supervised detectie van planten. Hierbij wordt onderzocht of het mogelijk is om zonder een volledige labeling alsnog een systeem te kunnen ontwikkelen voor de detectie van planten in een gevarieerde omgeving. In de ontwikkeling worden dus de overdraagbaarheid van de nieuwe methode en het minimaliseren van menselijke interventie centraal geplaatst. Hiervoor wordt gebruik gemaakt van deep learning.

De resultaten van de uitwerking van deze methode, weergegeven in Sectie 5.2, zijn veelbelovend en geven een aanzet om verder te gaan met deze conceptueel uitgebouwde methodiek. De behaalde F1-score van 78,15% ligt nog deels achter op de state of the art. Echter, zoals Kamilaris et al. (2018) [9] (zie hoofdstuk 3) aanhaalden in hun onderzoek naar deep learning in de landbouw, is



Figuur 5.12: Resultaat na toepassen methode op veld met spinazieplanten.

het vergelijken van resultaten met andere methoden niet eenvoudig is omwille van de diversiteit qua metrieken, modellen, toepassingen, datasets, randvoorwaarden ...

Uit de grafieken in voorgaande sectie blijkt bovendien dat de resultaten convergeren, ongeacht de hoeveelheid extra samples die worden toegevoegd. Het is bij deze aanpak bijgevolg niet nodig om te investeren in extra gelabelde samples. Het volledig labelen van afbeeldingen is immers een kost- en tijdsroevende methode. Deze proof of concept heeft dan ook als doel om op termijn een beter alternatief hiervoor te bieden.

Om de resultaten verder te optimaliseren en de methode robuuster te maken, zal het totaalconcept van deze methode op diverse vlakken bijgestuurd moeten worden. Verder onderzoek naar onder meer pre-processing (scheiding plant/achtergrond) en het automatisch genereren van een selectieve set samples om te labelen, dient te gebeuren om deze proof of concept verder te ontwikkelen tot een bruikbare methode. Enkele suggesties hiervoor worden gegeven in Sectie 6.2.

6

Conclusie en perspectieven

6.1 Conclusie

In deze masterproef worden enkele mogelijkheden van deep learning in computervisie-technieken binnen de preciselandbouw onderzocht. Zoals beschreven in hoofdstuk 3, is het gebruik van deep learning in de preciselandbouw een recentelijk en groeiend onderzoeksgebied waar nog veel mogelijkheid is tot verbetering. Dit werk omvat twee grote delen, opgebouwd rond diverse onderzoeksvragen zoals geformuleerd in sectie 1.1.

Het eerste deel van deze masterproef komt aan bod in hoofdstuk 4, waar twee verkennende case studies met betrekking tot classificaties in de preciselandbouw uitgewerkt worden. Dit sluit bijgevolg aan op de eerste deelvraag, namelijk nagaan of de automatisatie van bepaalde classificatie-toepassingen in de landbouw mogelijk is.

De eerste case studie betreft de herkenning van ritnaalden, een schadelijke insectensoort voor landbouwgewassen. De indeling in soorten gebeurt op dit moment op basis van DNA-onderzoek, wat het huidige kunnen van computervisie onmiddellijk op de proef stelde. Er kan hieruit besloten worden dat de computer tot een zeker niveau in staat is om bepaalde eigenschappen te herkennen, maar een geautomatiseerde classificatie in dit geval nog niet mogelijk is. Vermits de mens zelf nog niet in staat is om deze classificatie puur visueel uit te voeren, ligt dit resultaat in

de lijn der verwachtingen. Zoals Zhang [84] in zijn werk aanhaalde, blijft het steeds een uitdaging voor computers om op een robuuste manier bepaalde soorten visuele informatie te verwerken. Eventueel verder onderzoek in samenwerking met experts in het vakgebied van ritnaalden kan mogelijk nog betere resultaten opleveren.

De tweede casestudie kent een directe praktische doelstelling: het opbouwen van een geautomatiseerd systeem voor de ontharting van bloemkolen. Hierbij dient telkens de oriëntatie van de bloemkool (hart naar boven of onder) bepaald te worden om deze vervolgens naar de juiste machine te sturen. Het gebruik van een CNN voor supervised classificatie op basis van een beperkt aantal genomen foto's (449) van de bloemkolen op de proefopstelling van de transportband, zorgt voor een nauwkeurigheid van 98,89%. Hieruit kan besloten worden dat de praktische realisatie van een geautomatiseerd systeem in deze toepassing mogelijk is.

Zoals beschreven in hoofdstuk 3 gaat er in tal van huidige projecten veel tijd verloren met de noodzakelijke menselijke interventies in geautomatiseerde systemen. In het geval van computer-visie situeert het probleem zich meestal rond het tijdrovende labelen. Het doel van dit tweede deel van de masterproef is dan ook het aanbieden van een eenvoudige en efficiënte methode voor de detectie van planten zonder de noodzaak aan een volledig gelabeld systeem. In hoofdstuk 5 wordt hiervoor gefocust op de ontwikkeling van een proof of concept voor een semi-supervised detectie van planten in drone-afbeeldingen. Hierbij komen dus de twee andere deelvragen uit sectie 1.1 aan bod.

De beschreven methode maakt gebruik van een autoencoder die unsupervised getraind wordt op basis van een overzichtsfoto van een veld. Deze methode is interessant in termen van flexibiliteit, aangezien ze eenvoudig gebruikt kan worden voor een andere toepassing met andere afbeeldingen. Op die manier tracht de methode tegemoet te komen aan de enorme variatie van soorten gewassen en omgevingen in de landbouw.

Eens het netwerk getraind is, wordt aan de gebruiker gevraagd om enkele samples te categoriseren als positief (interessante zaken zoals bv. planten) en negatief (bv. achtergrond). Op basis van deze gelabelde samples wordt een voorstel gedaan voor een complete labeling, die vervolgens als actieve lus opnieuw bijgestuurd kan worden door de gebruiker. Met deze aanpak dient de expert/gebruiker dus enkel te grijpen bij de captatie van de data en de beperkte labeling. De rest van het algoritme kan zonder supervisie werken, waardoor er vlot ingespeeld kan worden op de specifieke noden van een bepaald veld.

Deze nieuwe aanpak van detectie in de precisielandbouw levert veelbelovende resultaten op en bevestigt hiermee wat Kamilaris et al. (2018) [9] in hun onderzoek naar deep learning in de landbouw aanhaalden. Tijdens de ontwikkeling van deze proof of concept kwamen diverse moeilijkheden naar voren, die met behulp van verder onderzoek de methode tot een hoger niveau van robuustheid kunnen brengen. Omwille van de beperkte tijd in deze masterproef worden enkele

suggesties voor in de toekomst gegeven in volgende sectie. Verder onderzoek moet bijgevolg uitmaken of het mogelijk is om deze methode uit te bouwen tot voldoende flexibele maar robuuste applicatie.

6.2 Perspectieven

In deze sectie worden enkele mogelijke onderzoekspistes geschetst voor een verdere uitbouw van de proof of concept uit hoofdstuk 5. Binnen dit nieuwe onderzoeksdomein zijn immers nog veel mogelijkheden tot verbetering. Bovendien kan de methode meer toegespitst worden op een bepaalde toepassing om vanuit die ingesteldheid verder te bouwen. Diverse mogelijkheden tot uitbreidingen worden aangehaald in verschillende werken uit de literatuurstudie in hoofdstuk 3. Hieronder volgt een korte beschrijving van deze mogelijke uitbreidingen.

Selectiekaders voor objectdetectie

In het domein van objectdetectie is het eerst en vooral de gewoonte om rond de gedetecteerde planten een selectiekader te tekenen. Op die manier kan de positie van de plant exact bepaald worden, om zo bijvoorbeeld een robot aan te sturen. Wanneer deze methode verder uitgebouwd wordt voor dit soort toepassingen, kan als bijkomende onderzoeksvraag 'hoe kan het raster omgezet worden naar selectiekaders' toegevoegd worden.

Pre-processing technieken

Er kan onderzocht worden hoe de methode uitgebreid kan worden met diverse pre-processing technieken. Uit de gemaakte literatuurstudie blijkt dat het merendeel van de werken hiervan gebruik maakt. Een plant/achtergrond scheiding kan bijvoorbeeld reeds een eerste stap van de indeling in interessant en oninteressante data zijn. Bovendien kan na het maken van dit onderscheid een groepering uitgevoerd worden, waarbij bepaalde gelijkaardige pixels geconnecteerd worden. Dit kan onder meer een hulp zijn bij de bepaling van de hierboven vermelde selectiekaders voor objectdetectie. Hierbij blijft er altijd de trade-off in welke mate eventuele parameters zoals de drempelwaarde ingesteld moeten worden, ten opzichte van de autonomie van het systeem. Het doel blijft immers een zoveel mogelijk geautomatiseerde pipeline van begin tot het einde met minimale menselijke interventie.

Spectra van sensoren

In deze proof of concept wordt gebruik gemaakt van RGB-afbeeldingen. Zoals in de literatuurstudie aan bod kwam, bestaat de mogelijkheid om een ander type camera te gebruiken waarbij extra spectrale banden worden toegevoegd. Er kan in verder onderzoek nagegaan worden of het nuttig is een NIR-band toe te voegen en de extra kosten hiervoor in te calculeren. Uit diverse werken [42, 44, 49] blijkt immers dat dankzij de reflectie van het nabij-infrarood licht door de vegetatie op eenvoudige wijze een eerste scheiding plant/achtergrond kan gebeuren.

Netwerkarchitectuur

Op vlak van de netwerkarchitectuur kan gedacht worden aan een systeem om de feature vectoren nog representatiever te maken voor de diverse afbeeldingen. Lottes et al. [43] maken hiervoor gebruik van een parallel pad met een extra decoder structuur die vervolgens meer ruimtelijke informatie meeneemt, bovenop de bepaalde visuele features.

Semi-supervised learning

Tenslotte kan ook op vlak van de labeling verder onderzoek verricht worden. Zoals in hoofdstuk 5 wordt aangehaald, kent de keuze van de gelabelde samples een belangrijke invloed op de benodigde hoeveelheid samples om tot een goed resultaat te komen. Een belangrijke onderzoeksvraag die hierbij rijst, is hoe er een referentiële subset samples geselecteerd kan worden die vervolgens aan de gebruiker aangeboden worden om te labelen. Dit proces kan vervolgens iteratief plaatsvinden om de resultaten te optimaliseren. Op die manier evolueert de aanpak richting active-learning, een speciale versie van semi-supervised learning waarbij het algoritme interactief de gebruiker gaat bevragen, om uit de ongelabelde data een zo'n klein mogelijke set te bepalen die de totale dataset zo goed mogelijk representeert [85]. Hieronder wordt een mogelijke aanpak beschreven die in de toekomst onderzocht kan worden.

Onderstaand voorstel voor de selectie van een subset, wordt gebaseerd op de proof of concept van Hung et al. [47] voor classificatie van onkruid. Hung et al. [47] maken hierbij gebruik van een gelabelde dataset met onkruid en niet-onkruid afbeeldingen. In eerste fase gaan ze zogenaamde afbeeldingsfilters bepalen op basis van de CIFAR10-dataset. Vervolgens worden de features van alle gelabelde afbeeldingen bepaald door deze afbeeldingsfilters te convolueren met de afbeelding. Daarna wordt er K-means clustering toegepast op de data de diverse klassen, waarbij voor elke klasse de zwaartepunten (kernen) van de clusters bepaald worden. Wanneer een classificatie van een stukje afbeelding uitgevoerd moet worden, wordt van dit gedeelte de feature vector bepaald en vervolgens vergeleken met die van de zwaartepunten. In dit geval wordt deze methode gebruikt

als classificatiesysteem.

Het is echter mogelijk om deze achterliggende redenering te gebruiken voor de selectie van interessante samples die gelabeld kunnen worden. Hiervoor kan als volgt te werk gegaan worden:

1. Bepaal de feature vectoren van elk vakje uit het raster (of een andere definiëring van vakjes) nadat het netwerk unsupervised getraind werd.
2. Maak gebruik van een unsupervised clusteringsalgoritme zoals K-means clustering. Hierbij wordt een K-waarde gekozen die aangeeft hoeveel zwaartepunten (kernen) en dus clusters er bepaald moeten worden in de hoogdimensionale ruimte van de feature vector. Zo'n kern representeert het centrum van de cluster. Alle andere datapunten in de ruimte worden toegekend aan de dichtste cluster, waarbij bovendien getracht wordt om de zwaartepunten zo klein mogelijk te houden [86].
3. Vraag aan de gebruiker om deze K zwaartepunten te labelen. Deze vormen immers de zwaartepunten van de diverse clusters en dus is de kans reëel dat deze punten een goede representatie zijn van het geheel, op het niveau van K samples.

Deze methode kan eventueel iteratief uitgevoerd worden om een zo laag mogelijke K-waarde te bepalen die toch een goed resultaat oplevert in de detectie. De K-waarde wordt bijgevolg een parameter die met behulp van trial and error bijgestuurd kan worden.

Een combinatie van deze methode met bijvoorbeeld een plant/achtergrond scheiding, kan ervoor zorgen dat de zwaartepunten en hun bijhorende cluster, op basis van die scheiding reeds een pre-labeling krijgen. Op die manier hoeft de gebruiker enkel de fout geclassificeerde punten aan te duiden om het systeem bij te sturen.

Kortom er zijn nog heel wat mogelijkheden om deze proof of concept uit te breiden en optimaliseren. Deze mogelijkheden kunnen in toekomstig onderzoek verder geëxploreerd worden.

Bibliografie

- [1] S. University, “Cs231n: Convolutional neural networks for visual recognition,” <http://cs231n.stanford.edu/>.
- [2] E. Pikel. (2017) Feed forward and recurrent ann architecture. [Online]. Available: https://www.researchgate.net/figure/Feed-forward-and-recurrent-ANN-architecture_fig1_315111480
- [3] U. Udofia. (2018) Basic overview of convolutional neural network (cnn). [Online]. Available: <https://medium.com/@udemeudofia01/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17>
- [4] A. Deshpande. (2016) A beginner’s guide to understanding convolutional neural networks - part 2. [Online]. Available: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner’s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>
- [5] A. Deshpande. (2016) A beginner’s guide to understanding convolutional neural networks - part 1. [Online]. Available: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner’s-Guide-To-Understanding-Convolutional-Neural-Networks/>
- [6] Y. Saeys, “Machine learning introductie,” 2018, slides available for students at University Ghent concerning the course ‘Artificiële intelligentie’.
- [7] S. Cool, “Precisielandbouw: een samenvatting,” Presentatie, 2019.
- [8] H. Yalcin, “Plant phenology recognition using deep learning: Deep-pheno,” pp. 1–5, Aug 2017.
- [9] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70 – 90, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169917308803>
- [10] S. Banu, “Precision agriculture: Tomorrow’s technology for today’s farmer,” *Journal of Food Processing & Technology*, vol. 6: 468, 2015.

- [11] J. McCarthy, “What is artificial intelligence,” 2007. [Online]. Available: <http://jmc.stanford.edu/articles/whatisai/whatisai.pdf>
- [12] Y. Saeys, “Artificiële intelligentie,” 2018, slides available for students at University Ghent concerning the course ‘Artificiële intelligentie’.
- [13] M. Molina, “Introduction to deep learning,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [14] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014.
- [15] D. I. Patrício and R. Rieder, “Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review,” *Computers and Electronics in Agriculture*, vol. 153, pp. 69 – 81, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169918305829>
- [16] D. Manrique Gamo, “Neural networks and deep learning: Foundations,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [17] V. Vanhoucke, A. Senior, and M. Z. Mao, “Improving the speed of neural networks on cpus,” 2011.
- [18] M. Ponti, L. Ribeiro, T. Santana Nazare, T. Bui, and J. Collomosse, “Everything you wanted to know about deep learning for computer vision but were afraid to ask,” pp. 17–41, 10 2017.
- [19] L. Baumela, “Introduction to computer vision,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [20] P. Veelaert, “Basic problems computer vision,” 2019, slides available for students at University Ghent concerning the course ‘Computervisie’.
- [21] A. Golda. (2005) Introduction to neural networks. [Online]. Available: <http://home.agh.edu.pl/~vlsi/AI/intro/>
- [22] M. A. Nielsen, *Neural networks and deep learning*. Determination Press, 2015.
- [23] A. M. Tekalp and O. Kirmemis. (2018) Deep learning for image and video processing. University Lecture.
- [24] D. Manrique Gamo, “Neural networks and deep learning: Learning. logistic regression,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.

- [25] D. Manrique Gamo., “Neural networks and deep learning: Scaling up. deep learning,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [26] L. Baumela, “Convolutional neural networks (i): Fundamentals,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [27] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” pp. 9–50, 1998. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645754.668382>
- [28] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [29] A. Pizurica, “Neural networks,” 2018, slides available for students at University Ghent concerning the course ‘Artificial intelligence’.
- [30] M. Molina, “Performance evaluation,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [31] S. university. Autoencoders. [Online]. Available: <http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/>
- [32] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, “The difficulty of training deep architectures and the effect of unsupervised pre-training,” vol. 5, pp. 153–160, 16–18 Apr 2009. [Online]. Available: <http://proceedings.mlr.press/v5/erhan09a.html>
- [33] A. Pizurica, “Supervised learning,” 2018, slides available for students at University Ghent concerning the course ‘Artificial intelligence’.
- [34] D. Soni. (2018) Supervised vs. unsupervised learning. [Online]. Available: <https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d>
- [35] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information,” *CoRR*, vol. abs/1703.00810, 2017. [Online]. Available: <http://arxiv.org/abs/1703.00810>
- [36] W. H. Maes and K. Steppe, “Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture,” *Trends in Plant Science*, vol. 24, no. 2, pp. 152 – 164, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1360138518302693>
- [37] W. Nugteren, “Precisielandbouw en gewasbescherming: hoe precies?” 2009. [Online]. Available: <https://www.knpv.org/db/upload/documents/Precisielandbouw/Nugteren.pdf>
- [38] A. Chlingaryan, S. Sukkarieh, and B. Whelan, “Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review,” *Computers and Electronics in Agriculture*, vol. 151, pp. 61–69, 08 2018.

- [39] J. V. Stafford, "Implementing precision agriculture in the 21st century," *Journal of Agricultural Engineering Research*, vol. 76, no. 3, pp. 267 – 275, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021863400905778>
- [40] N. Zhang, M. Wang, and N. Wang, "Precision agriculture-a worldwide overview," *Computers and Electronics in Agriculture*, vol. 36, no. 2-3, pp. 113 – 132, 2002.
- [41] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63 – 78, 2008, emerging Technologies For Real-time and Integrated Agriculture Decisions. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169907001688>
- [42] P. Lottes, M. Hoferlin, S. Sander, M. Müter, P. Schulze, and L. C. Stachniss, "An effective classification system for separating sugar beets and weeds for precision farming applications," pp. 5157–5163, May 2016.
- [43] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, "Fully convolutional networks with sequential information for robust crop and weed detection in precision farming," *CoRR*, vol. abs/1806.03412, 2018. [Online]. Available: <http://arxiv.org/abs/1806.03412>
- [44] C. Potena, D. Nardi, and A. Pretto, "Fast and accurate crop and weed identification with summarized train sets for precision agriculture," *Advances in Intelligent Systems and Computing*, vol. 531, pp. 105–121, 02 2017.
- [45] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," *CoRR*, vol. abs/1709.06764, 2017. [Online]. Available: <http://arxiv.org/abs/1709.06764>
- [46] M. Fawakherji, A. Youssef, D. Bloisi, A. Pretto, and D. Nardi, "Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation," pp. 146–152, Feb 2019.
- [47] C. Hung, Z. Xu, and S. Sukkariéh, "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav," *Remote Sensing*, vol. 6, pp. 12 037–12 054, 12 2014.
- [48] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "Uav-based crop and weed classification for smart farming," pp. 3024–3031, May 2017.
- [49] A. Milioto, P. Lottes, and C. Stachniss, "Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W3, pp. 41–48, 2017. [Online]. Available: <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2-W3/41/2017/>

- [50] M. D. Bah, A. Hafiane, and R. Canals, “Deep learning with unsupervised data labeling for weeds detection on UAV images,” *CoRR*, vol. abs/1805.12395, 2018. [Online]. Available: <http://arxiv.org/abs/1805.12395>
- [51] M. Pérez-Ortiz, P. A. Gutiérrez, J. M. Peña-Barragán, J. Torres-Sánchez, C. Martínez, and F. López-Granados, “An experimental comparison for the identification of weeds in sunflower crops via unmanned aerial vehicles and object-based analysis,” vol. 9094, 06 2015.
- [52] M. Pérez-Ortiz, J. M. Peña, P. A. Gutiérrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados, “Selecting patterns and features for between- and within- crop-row weed mapping using uav-imagery,” *Expert Systems with Applications*, vol. 47, pp. 85 – 94, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417415007472>
- [53] E. Hamuda, M. Glavin, and E. Jones, “A survey of image processing techniques for plant extraction and segmentation in the field,” *Computers and Electronics in Agriculture*, vol. 125, pp. 184 – 199, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169916301557>
- [54] J. Gao, W. Liao, D. Nuyttens, P. Lootens, J. Vangeyte, A. Pižurica, Y. He, and J. G. Pieters, “Fusion of pixel and object-based features for weed mapping using unmanned aerial vehicle imagery,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 67, pp. 43 – 53, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0303243417303252>
- [55] “Color indices for weed identification under various soil, residue, and lighting conditions,” *Transactions of the ASABE*, vol. 38, no. 1, pp. 259–269, 1 1995.
- [56] Y. Kyosuke, W. Guo, Y. Yoshioka, and S. Ninomiya, “On plant detection of intact tomato fruits using image analysis and machine learning methods,” *Sensors (Basel, Switzerland)*, vol. 14, pp. 12 191–12 206, 07 2014.
- [57] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, “Counting apples and oranges with deep learning: A data-driven approach,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, April 2017.
- [58] O. Csillik, “Superpixels: the end of pixels in obia. a comparison of state-of-the-art superpixel methods for remote sensing data.”
- [59] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311 – 318, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169917311742>
- [60] X. Pantazi, D. Moshou, T. Alexandridis, R. Whetton, and A. Mouazen, “Wheat yield prediction using machine learning and advanced sensing techniques,” *Computers*

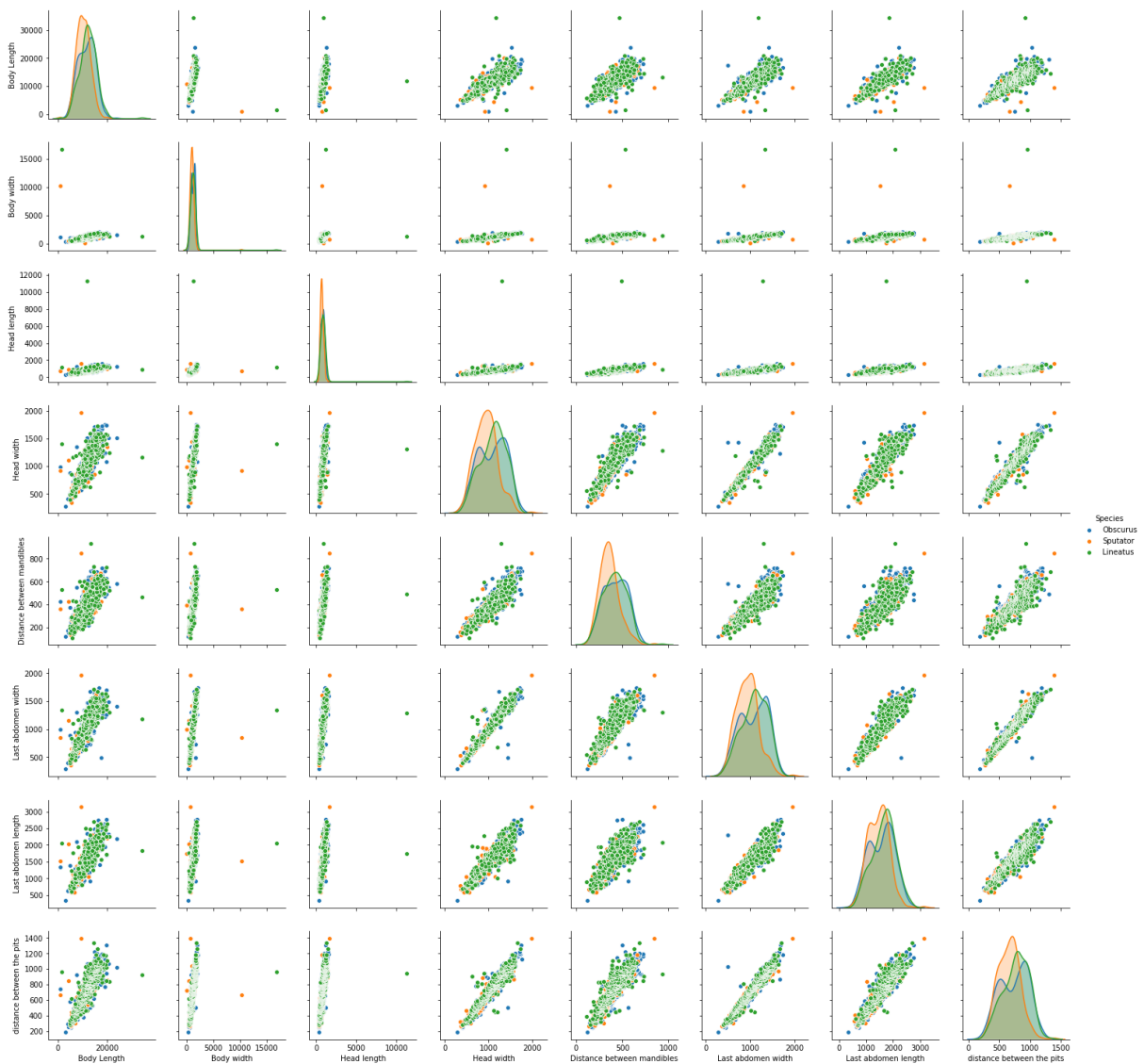
- and Electronics in Agriculture*, vol. 121, pp. 57 – 65, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169915003671>
- [61] J. Cnops and R. Stoop, “Algoritmen ii,” 2018, available for students at University Ghent-concerning the course ‘Gevorderde algoritmen’.
- [62] M. D. Cicco, C. Potena, G. Grisetti, and A. Pretto, “Automatic model based dataset generation for fast and accurate crop and weeds detection,” *CoRR*, vol. abs/1612.03019, 2016. [Online]. Available: <http://arxiv.org/abs/1612.03019>
- [63] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [64] TensorFlow, “Tensorflow.” [Online]. Available: <https://www.tensorflow.org/>
- [65] Keras, “Keras.” [Online]. Available: <https://keras.io/>
- [66] C. Van Ceulebroeck, L. Vanhaeren, F. Tememrman, V. De Blauwer, N. Rutten, J. Witters, L. Taning, N. Berckvens, and H. Casteels, “Onderzoek naar ritnaalden uit de startblokken,” pp. 10–13, 2 2016.
- [67] M. P. Paing and S. Choomchuay, “Improved random forest (rf) classifier for imbalanced classification of lung nodules,” pp. 1–4, July 2018.
- [68] R. G. Devi and P. Sumanjani, “Improved classification techniques by combining knn and random forest with naive bayesian classifier,” pp. 1–4, March 2015.
- [69] D. Manrique Gamo, “Neural networks and deep learning: Learning. linear regression,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [70] L. Baumela, “Convolutional neural networks (ii): Regularization,” 2018, slides available for students EIT master in DataScience at UPM concerning the course ‘Deep Learning’.
- [71] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: <http://arxiv.org/abs/1311.2901>
- [72] K. Sabanci, A. Kayabasi, and A. Toktas, “Computer vision-based method for classification of the wheat grains using artificial neural network,” *Journal of the Science of Food and Agriculture*, vol. 97, p. 2588–2593, 06 2017.
- [73] Agisoft LLC, “Agisoft metashape.” [Online]. Available: <https://www.agisoft.com/about/>
- [74] PyTorch, “Torchvision.transforms package.” [Online]. Available: <https://pytorch.org/docs/stable/torchvision/transforms.html>

- [75] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” *CoRR*, vol. abs/1901.03407, 2019. [Online]. Available: <http://arxiv.org/abs/1901.03407>
- [76] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, “Anomaly detection using autoencoders in high performance computing systems,” *CoRR*, vol. abs/1811.05269, 2018. [Online]. Available: <http://arxiv.org/abs/1811.05269>
- [77] A. Bhattad, J. Rock, and D. A. Forsyth, “Detecting anomalous faces with ‘no peeking’ autoencoders,” *CoRR*, vol. abs/1802.05798, 2018. [Online]. Available: <http://arxiv.org/abs/1802.05798>
- [78] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, “Deep autoencoding models for unsupervised anomaly segmentation in brain MR images,” *CoRR*, vol. abs/1804.04488, 2018. [Online]. Available: <http://arxiv.org/abs/1804.04488>
- [79] J. Jelínek. (2007) Metrics in similarity search. [Online]. Available: https://is.muni.cz/th/cxyzo/SimSearch_Metrics.pdf
- [80] E. S. M. and Y. N., “Robustness of canberra metric in computer intrusion detection,” *Proceedings of the IEEE Workshop on Information Assurance and Security*, 01 2001.
- [81] A. Kumar, A. Abdelhadi, and C. Clancy, “Novel anomaly detection and classification schemes for machine-to-machine uplink,” pp. 1284–1289, Dec 2018.
- [82] L. Derksen. (2016) Visualising high-dimensional datasets using pca and t-sne in python. [Online]. Available: <https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b>
- [83] PyTorch, “Pytorch.” [Online]. Available: <https://pytorch.org/>
- [84] B. Zhang, “Computer vision vs. human vision,” pp. 3–3, July 2010.
- [85] S. Hosein. (2018) Active learning: Curious ai algorithms. [Online]. Available: <https://www.datacamp.com/community/tutorials/active-learning>
- [86] M. J. Garbade. (2018) Understanding k-means clustering in machine learning. [Online]. Available: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

Bijlagen

Bijlage A - Diverse grafieken omtrent classificatie van ritnaalden

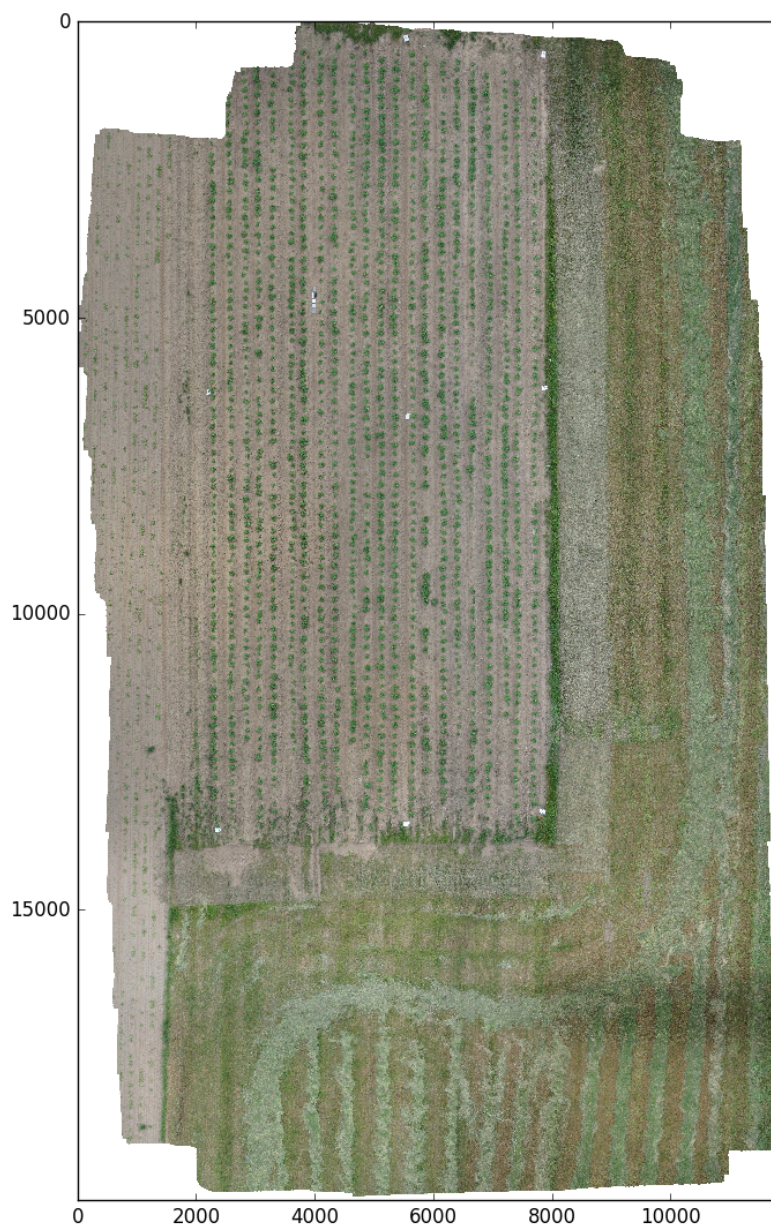
Hier wordt een overzicht gegeven van allerlei grafieken, met elk van de gemeten gegevens uit Sectie 4.1 (zie ook tabel 4.1) eens op de x-as, alsook op de y-as, waardoor alle mogelijke combinaties weergegeven worden. Zoals vermeld zijn er geen echt duidelijk waarneembare onderscheiden mogelijk.



Figuur 1: Weergave van diverse gegevens in functie van elkaar omtrent classificatie van ritnaalden.

Bijlage B - Overzichtsfoto aardappelplanten

Deze foto werd gebruikt ter ontwikkeling van de methode in hoofdstuk 5. De originele afmetingen bedragen 19904x11826 pixels.



Figuur 2: Overzichtsfoto aardappelveld.