

Een computervisie gebaseerde aanpak voor het bepalen van de snijlijn van witloof

Wout DESAEVER

Promotor: prof. dr. M. Verbeke

Co-promotor: dr. ing. M. De Ryck

Co-promotor: ir. E. Vandenbogaerde

Masterproef ingediend tot het behalen van de
graad van master of Science in de industriële
wetenschappen: elektronica - ICT,
afstudeerrichting ICT

Academiejaar 2022 - 2023

©Copyright KU Leuven

Without written permission of the supervisor(s) and the author(s) it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilise parts of this publication should be addressed to KU Leuven Campus Brugge, Spoorwegstraat 12, B-8200 Brugge, +32 50 66 48 00 or via e-mail iiw.brugge@kuleuven.be.

A written permission of the supervisor(s) is also required to use the methods, products, schematics and programs described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Samenvatting

Het grootste aandeel van de West-Vlaamse witlooftelers snijdt de wortel van witloof nog manueel af. Hierbij wordt de wortel op een roterende klem geplaatst en brengt de klem de wortel naar een statisch mes. Aangezien wortels en witloof nooit dezelfde vorm hebben en een persoon de wortels in de klem steekt, wordt de krop niet perfect van de wortel gesneden. Door deze imperfectie is er nadien vaak nog een stuk van de wortel dat extra met de hand bijgesneden moet worden of worden er te veel bladeren door de snijmachine van de krop gesneden. De imperfecties zorgen ervoor dat de teler veel economische verliezen ondervindt zoals extra personeelskosten, gewichtsverliezen en tijdsverliezen.

Dit project tracht dit probleem op te lossen door een software algoritme te ontwikkelen en te valideren dat in staat is om de optimale snijlijn te bepalen van witloof dat achteraf door een geautomatiseerde machine kan worden gebruikt om het witloof op die bepaalde lijn te snijden. Tevens wordt een algoritme ontwikkeld en gevalideerd dat gesneden witloof classificeert in goed of slecht gesneden.

Uiteindelijk wordt een neurale netwerk bekomen op basis van ResNet dat in staat is om het witloof te classificeren als witloof dat nog extra bijgesneden moet worden en witloof dat niet bijgesneden moet worden met een gemiddelde juistheid van 99,72%. Voor de detectie van het witloof worden de insluitende rechthoeken van de labels correct herschaald naar een kleiner formaat. Het model begint te leren, maar na enkele epochs kan het model niet meer verbeteren om momenteel nog onbekende redenen. Het detecteren van de ideale snijlijn is theoretisch onderzocht en een plan om dit in de praktijk te verwezenlijken is uitgedacht.

Kernwoorden: witloof snijden; objectdetectie; objectclassificatie; deep learning

Summary

The largest share of the West-Flemish chichory growers still cut the root of chichory manually. To accomplish this the root is put in a rotating clamp and the clamp brings the root to a stationary knife. Since the roots and the chichory never have the same shape and because a person puts the roots in a clamp, the root is not perfectly cut from the crop. Because of these imperfections there is afterwards often a part of the root that must be cut further by hand or there are a lot of leaves cut from the crop by the machine. The imperfections causes a lot of economic losses to the growers such as extra personnel costs, weight losses and time losses.

This project attempts to solve this problem by developing and validating a software algorithm that is capable of determining the optimal cutting line of chichory that can later be used by a automated machine to cut chichory along that line. At the same time an algorithm is developed and validated that classifies the cut chichory in good and bad cut chichory.

In the end a neural network was developed based on ResNet that is capable of classifying chichory as chichory that needs to be cut extra and chichory that does not need to be cut extra anymore with an average accuracy of 99,72%. For the detection of chichory the bounding boxes from the labels are correctly rescaled to a smaller format. The model starts to learn but after a couple of epochs for some still unknown reasons the model can't improve anymore. The detection of the optimal cutting line has been theoretically researched and a plan has been made to develop it in practice.

Keywords: dutting chichory; object detection; object classification; deep learning

Dankwoord

Doorheen het voorbije academiejaar heb ik een thesis geschreven over het detecteren van de optimale snijlijn van witloof en het classificeren van witloof in goed en slecht gesneden witloof. Ik heb dit uiteraard niet alleen verwezenlijkt. Daarom wil ik enkele personen bedanken die mij tijdens mijn onderzoek geholpen hebben.

De thesis kwam tot stand in nauwe samenwerking met Inagro. Een speciaal dankwoord voor de medewerkers Inagro die ervoor gezorgd hebben dat er veel data van witloof verzameld en gelabeld konden worden.

Verder wil ik mijn promotor prof. dr. M. Verbeke bedanken voor de begeleiding tijdens het jaar en voor zijn waardevol advies.

Ik werd ook bijgestaan door mijn co-promotor aan de KULeuven dr. ing. M. De Ryck. Hij heeft mij op regelmatige basis advies en waardevolle feedback bezorgd, waarvoor ik graag een dankbetuiging wil geven.

Daarnaast werd ik ook begeleid door mijn co-promotor van Inagro ir. E. Vandenbogaerde. Dank u wel om mij op regelmatige basis advies te geven, om mij te informeren omtrent de verwerking van witloof en voor uw waardevolle tips bij het uitwerken van deze masterproef.

Verder wil ik de KULeuven campus Brugge bedanken om mij de voorbije jaren bij te staan tijdens mijn volledige opleiding.

Tot slot een bijzonder dankwoord voor mijn ouders en familie, hun onafgebroken steun en geloof in mijn kunnen sterkten mij om mijn studies en mijn masterproef tot een goed einde te brengen.

Wout Desaevers

Brugge, 1 juni 2023

Inhoudsopgave

Samenvatting	iii
Summary	iv
Dankwoord	v
Inhoudsopgave	vii
Lijst van figuren	viii
Lijst van tabellen	x
1 Introductie	1
1.1 Motivatie	1
1.2 Objectieven	2
1.3 Verloop van de thesis	3
2 Literatuurstudie	4
2.1 Classificatie	4
2.2 Objectdetectie	8
2.3 Lijndetectie	10
3 Methodologie	17
3.1 Camera set-up	17
3.2 Camera specificaties	17
3.3 Data verzameling	18
3.4 Data labeling	20
3.5 Data inladen	21
3.6 Preprocessing	22
3.7 Model	23
4 Resultaten	26
4.1 Duiding van de deelresultaten	26
4.2 Classificatie van gesneden witloof	26
4.3 Detectie van witloof	28

4.4 Detectie van de optimale snijlijn	30
5 Conclusies	34
5.1 Conclusies	34
5.2 Toekomstig werk	35
6 Bibliografie	36

Lijst van figuren

1.1	De snijmachine waarmee het witloof van de wortels gesneden wordt.	2
1.2	Logo van Inagro.	2
2.1	Vergelijken van het histogram met de standaard histogrammen van elke klasse. . . .	5
2.2	De juistheid van het neurale netwerk dat de MATLAB toolbox gebruikt heeft.	6
2.3	De juistheid van het neurale netwerk dat de MATLAB toolbox niet gebruikt heeft. . . .	7
2.4	Opbouw van het model.	7
2.5	De opbouw van het model voor de detectie van wortelrot bij de ginseng plant.	8
2.6	Verskil in IoU bij een insluitende ronde (C-box) en insluitende rechthoek (R-box). . .	9
2.7	De verschillende stappen tijdens het bewerken van de suikerriet afbeelding.	11
2.8	Een histogram van de pixel intensiteit.	11
2.9	Bepaling van de middellijn volgens de middelpunten van 7 parallelle lijnen.	12
2.10	De stappen om de 4 vinnen te segmenteren na het vinden van de middellijn.	13
2.11	In positie (d) ligt de vis in de juiste positie. Er zijn meer witte pixels onderaan de afbeelding dan wanneer de vis omgekeerd ligt.	13
2.12	De sensitiviteit, specificiteit en juistheid voor de detectie van elke vin.	14
2.13	De insluitende rechthoeken van de knol, de wortel en de wortelstam. De groene rechthoek duidt de knol aan, de paarse rechthoek de wortel en de rode rechthoek de wortelstam.	14
2.14	Het gemiddelde (bolvorm) en de standaarddeviatie (ruitvorm) van de verschillende netwerken.	15
2.15	De leidende lijnen van foto's.	16
2.16	Voorstelling van een lijn met poolcoördinaten.	16
2.17	De lijnsegmentdetectie voor door mens gemaakte omgevingen. De blauwe lijnen zijn de voorspelde lijnen, de rode zijn de werkelijke lijnen.	16
3.1	De gebruikte camera constructie voor het nemen van de beelden.	18
3.2	Opstelling voor het nemen van foto's van witloofkroppen met de wortel	19
3.3	Opstelling voor het nemen van foto's van witloofkroppen met deels gesneden wortels. .	19
3.4	De gelabelde data in labelbox.com waarop het witloof en de snijlijn aangeduid zijn. .	21
3.5	Opslagstructuur voor classificatie.	21
3.6	Opslagstructuur voor detectie.	22
3.7	De omzetting van een bounding box naar een mask.	24

3.8	Originele afbeelding (apart_schuin.83.png).	25
3.9	Afbeelding na preprocessing met aangepaste insluitende rechthoek	25
4.1	Juistheid per epoch voor de eerste vijf getrainde modellen.	28
4.2	Juistheid per epoch voor de laatste vijf getrainde modellen.	28
4.3	De training en validatie loss van het classificatie-algoritme. De lichtgekleurde curves tonen de effectieve waarden van de loss, de donkere curves zijn afgevlakt met 60%.	29
4.4	Training loss van het detectiemodel met ResNet18 en een leersnelheid van 0,001.	29
4.5	Resultaat na 20 epochs met ResNet18. De blauwe rechthoek is het voorspelde resultaat, de groene is de werkelijke insluitende rechthoek.	30
4.6	Training loss van het detectiemodel met ResNet50 en een leersnelheid van 0,004.	30
4.7	Resultaat na 6 epochs met ResNet50.	31
4.8	De voorstelling van een snijlijn aan de hand van twee punten. De groene en de rode punten geven de juiste snijlijn voor. De blauwe punten liggen dicht bij de rode punten, maar stellen een heel andere snijlijn voor.	31
4.9	Voorstelling van een lijn met 2 waarden x_1 en x_2 door de snijlijn over de volledige afbeelding te laten lopen. De groene lijn is de gelabelde lijn. De rode lijn is de doorgetrokken lijn tot aan de grenzen van de afbeelding.	32
4.10	Voorstelling van een lijn met 2 waarden, θ en r . De groene lijn is de gelabelde snijlijn. De rode lijn is de doorgetrokken snijlijn. De blauwe lijn is de loodlijn op de snijlijn door de oorsprong.	33

Lijst van tabellen

3.1	Het aantal afbeeldingen per foto klasse.	20
4.1	Verwarringsmatrix van model 4 bij de hoogste juistheid.	27

1

Introductie

1.1 Motivatie

Het grootste aandeel van de West-Vlaamse witlooftelers snijdt de wortel van witloof nog manueel af. Om de wortel van het witloof te snijden wordt de wortel op een roterende klem gestoken (figuur 1.1). De roterende klem duwt het witloof op een statisch mes, waardoor de wortel afgesneden wordt en het witloof op de groene loopband valt. Aangezien wortels en witloof nooit dezelfde vorm hebben en een persoon de wortels in de klem steekt, wordt de krop niet perfect van de wortel gesneden. Er hangt vaak nog een stuk van de wortel aan de krop of er wordt te veel van de krop afgesneden waardoor er veel bladeren verloren gaan. Langs de groene loopband van de snijmachine staan er drie of vier personen die het witloof sorteren en indien nodig extra bijsnijden. Deze drie tot vier personen bepalen hoe snel de machine draait. Hoe meer er extra moet bijgesneden worden, hoe trager de machine zal moeten draaien. De teler heeft door deze manier van werken veel economische verliezen. Het extra personeel dat het witloof moet sorteren zorgt voor grotere personeelskosten. Indien er veel manueel moet bijgesneden worden, moet de machine trager werken en kan er minder witloof per tijdspanne gesneden en dus ook verkocht worden. Telkens dat er te veel van het witloof gesneden wordt, gaan er veel bladeren verloren. Hoe meer bladeren er verloren gaan, hoe minder gewicht aan witloof de teler kan verkopen.

In samenwerking met Inagro (figuur 1.2) is tijdens deze thesis geprobeerd om een algoritme te ontwikkelen en te valideren dat het witloof na het snijden automatisch controleert. De controle kijkt en geeft aan welke kappen er nog extra bijgesneden moeten worden. Er is ook geprobeerd om het witloof automatisch te detecteren en om de ideale snijlijn te bepalen zodat deze snijlijn in de toekomst gebruikt kan worden door een geautomatiseerde machine om het witloof over deze lijn te snijden.



Figuur 1.1: De snijmachine waarmee het witloof van de wortels gesneden wordt.



Figuur 1.2: Logo van Inagro.

1.2 Objectieven

Het algemene doel van deze thesis is om de economische verliezen van de teler te verminderen. Zoals eerder aangegeven gaat het over extra personeelskosten door controle en het bijsnijden van het witloof. Het tijdsverlies dat dit met zich meebrengt zorgt voor een rendementsverlies. Vermijden dat de wortel soms te ver wordt afgesneden, moet gewichtsverlies van verkoopbaar product reduceren. Om dit te bereiken wordt het algemene doel opgesplitst in de volgende subdoelen:

1. Het classificeren van witloof na het snijden in twee klassen, namelijk witloof dat nog bijgesneden moet worden en witloof dat niet meer bijgesneden moet worden. Hiervoor wordt een model gemaakt dat een afbeelding van een gesneden witloofkrop analyseert en aangeeft tot welke klasse die behoort.
2. Het detecteren van de optimale snijlijn van witloof. Hierbij analyseert een model een afbeelding van witloof met een wortel en zoekt naar de optimale snijlijn. De snijlijn is een lijn die over de volledige afbeelding loopt en wordt voorgesteld met twee waarden. De eerste waarde is de hoek tussen de x-as en de loodrechte op de snijlijn door de oorsprong. De tweede waarde is de lengte tussen de oorsprong en de optimale snijlijn.

3. Het detecteren van het witloof. Hierbij wordt een insluitende rechthoek getekend rond het witloof. De rechthoek wordt voorgesteld door de coördinaten van de linkerbovenhoek en de rechteronderhoek. Het detecteren van het witloof dient om bij het classificeren van het witloof en het detecteren van de optimale snijlijn gericht te kunnen werken door niet naar de volledige afbeelding te kijken, maar enkel naar het deel van de afbeelding waar het witloof zich bevindt. Door de focus te verleggen naar het deel van de afbeelding waar het witloof zich bevindt, is het mogelijk om meerdere witloofkroppen te analyseren op eenzelfde afbeelding.

1.3 Verloop van de thesis

In hoofdstuk 2 wordt de literatuurstudie besproken omtrent het gebruik van machinevisie om objecten te classificeren en om objecten te detecteren. Verder wordt in de literatuur onderzocht hoe een snijlijn bepaald kan worden. Er is studie gedaan naar zowel data-gedreven als niet data-gedreven machinevisie.

In hoofdstuk 3 wordt besproken hoe alle nodige data verzameld worden en wat de concrete stappen zijn om tot de ontwikkeling van de nodige algoritmes te komen.

In hoofdstuk 4 worden de bekomen resultaten besproken.

In hoofdstuk 5 wordt een conclusie over de thesis geformuleerd en wordt mogelijk toekomstig werk gesuggereerd.

2

Literatuurstudie

Machinevisie toepassingen kunnen gebruik maken van verschillende technieken. Een belangrijk onderscheid hierbij is het verschil tussen niet data-gedreven en data-gedreven machinevisie. Bij niet data-gedreven machinevisie kijkt een algoritme of een bepaald kenmerk van een gewenst object zich in een afbeelding bevindt en bepaalt hiervan de gewenste eigenschappen. Hierbij baseert het algoritme zich puur op basis van de door de mens gedefinieerde kenmerken. Bij data-gedreven machinevisie wordt er niet meegegeven hoe een object herkend kan worden. Er worden in de plaats data geleverd aan het algoritme. De trainingsdata bevatten beelden waarbij er beschreven staat welke aan de gewenste voorwaarden voldoen en welke niet. Het algoritme maakt op basis van de geleverde data een model. Na deze stap wordt het model getest aan de hand van testdata. Het model kan zichzelf trainen door herhaaldelijk de voorspelde waarden volgens het model te vergelijken met de werkelijke waarden en vervolgens zijn parameters aan te passen.

2.1 Classificatie

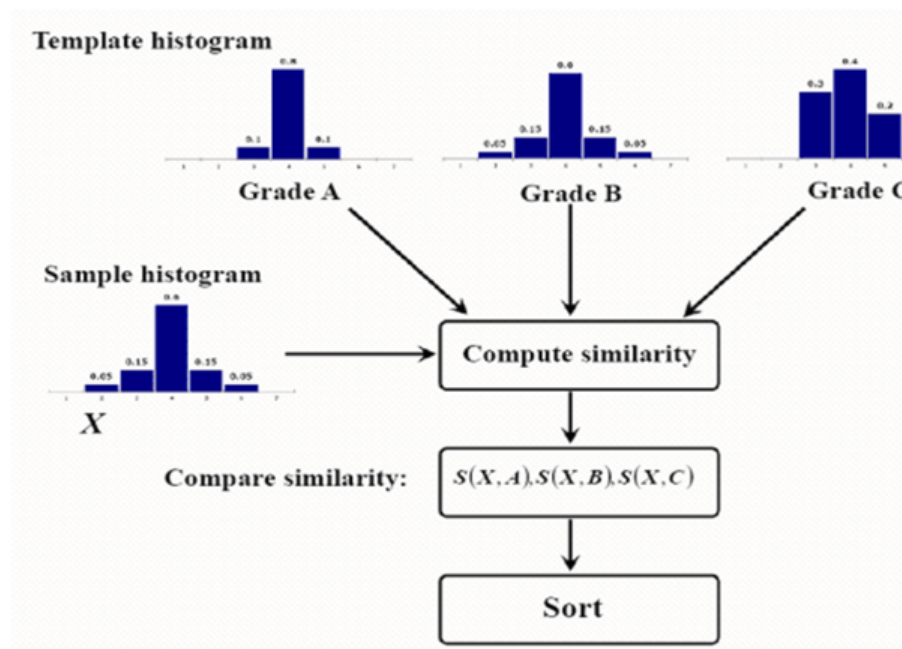
Li et al. [1] maken gebruik van niet data-gedreven technieken voor de inspectie van fruit. Het manueel inspecteren van de kwaliteit van fruit is een lastige, trage en inconsistente taak. Deze studie probeert de kwaliteit van fruit te beoordelen op basis van de kleur. De kleur van het fruit kan namelijk veel vertellen over de smaak, zuurtegraad, suikergehalte, enzovoort. Nadien moet het fruit verdeeld worden in vijf kwaliteitsklassen.

Hierbij worden er beelden gemaakt van het fruit met een charge-coupled device (CCD) camera. Omdat we als mens denken over kleuren in de "Hue, Saturation en Value" (HSV) kleurruimte, moeten de "Red Green Blue" (RGB) beelden omgezet worden in HSV. De driedimensionale informatie maakt de bewerkingen een stuk moeilijker. Daarom wordt de informatie omgezet in een eendimensionale reeks dat de verschillende kleurniveaus voorstelt. Door deze methode vermindert de bewerkingsbelasting sterk en gaat het proces sneller.

Er wordt naar de volledige afbeelding gekeken in plaats van op voorhand naar het fruit te zoeken. Hierdoor is het belangrijk dat de achtergrond van het fruit verwijderd wordt. De afbeelding wordt hiervoor omgezet naar de OHTA kleurruimte. Via een grenswaarde-algoritme wordt bepaald welke pixels tot de achtergrond behoren en welke tot het fruit.

Om ruis te verwijderen wordt vectorieel mediaan filtering gebruikt. De vectoriële methode filtert de pixels als een driedimensionale pixel terwijl scalair filteren voor elke kleur filtert en dan de resultaten samenvoegt. Scalair filteren zou nieuwe kleuren kunnen doen ontstaan die bij deze studie erg storend kunnen zijn. Vectorieel mediaan filteren zorgt ervoor dat de ruis kan weggefilterd worden zonder de kleur te veranderen. In de studie wordt de vectoriële methode gebruikt.

Voor elke kleur wordt er een histogram gemaakt die de frequentieverdeling beschrijft. Om het fruit te classificeren worden deze histogrammen zoals in figuur 2.1 vergeleken met de standaard histogrammen die gekoppeld zijn aan elke kwaliteitsklasse.



Figuur 2.1: Vergelijken van het histogram met de standaard histogrammen van elke klasse.

[1]

Deze methode wordt getest op tomaten. De tomaten worden verdeeld over vijf kwaliteitsklassen op basis van hun rijpheid. Elke klasse bevat vijftig tomaten. De klassen A en E geven voor alle tomaten het correcte resultaat, klasse B en D voor 96% en klasse C voor 94%.

Rossetto et al. [2] hebben een algoritme ontwikkeld met data-gedreven machinevisie die op CT-scans longkanker probeert te ontdekken. Alle 150 000 scans worden uit een al bestaande database gehaald, namelijk de Kaggle Data Science Bowl. Bij elke patiënt is gekend of die longkanker heeft of niet. Niet elke scan is op dezelfde manier genomen.

Alle afbeeldingen worden bewerkt door een Gaussiaanse filter. Een Gaussiaanse filter geeft de Gauss curve als impulsantwoord. Zowel de bewerkte als de onbewerkte foto's worden gebruikt om het algoritme te trainen. De data worden opgesplitst in training- en testdata met een verhouding van zeven ten opzichte van drie.

Om de afbeeldingen te classificeren worden twee verschillende convolutionele neurale netwerken gebruikt. Beide zijn op dezelfde manier gemaakt in MATLAB, maar de ene gebruikt de MATLAB Neural Network Toolbox en de andere niet. Om vals positieve resultaten te vermijden moeten beide neurale netwerken het resultaat als positief aanschouwen vooraleer het ultieme resultaat ook positief is.

De convolutionele laag gebruikt een 10 x 10 convolutiefilter om kenmerken te vinden. Om de dimensies van de kenmerken te verkleinen wordt er gepoold. De kenmerken worden vervolgens naar de softmax en fully connected layers gestuurd.

Het neurale netwerk dat de MATLAB toolbox gebruikt, heeft gemiddeld 55 minuten nodig om alle berekeningen te doen met een Nvidia GTX 1080 GPU. Hierbij heeft de grootste testgrootte een gemiddelde accuraatheid van 97,5%. Als de testgrootte groot genoeg is, blijft de ratio van vals positieve waarden onder de 10%. Ongefilterde afbeeldingen doen hier beter dan de gefilterde afbeeldingen (figuur 2.2).

Bij het neurale netwerk waar de MATLAB toolbox niet gebruikt wordt, variëren de vals positieve waarden sterk. Door de vereenvoudigde implementatie is de accuraatheid lager dan wanneer de MATLAB toolbox gebruikt wordt (figuur 2.3).

TABLE I: Accuracy for the toolbox Implementation.

Test Size	Unsmoothed	Smoothed	Accuracy	Precision	Recall	False Positives
10	80.00%	80.00%	80.00%	100.00%	100.00%	30.00%
11	91.00%	69.00%	91.67%	75.00%	100.00%	9.00%
23	78.00%	93.00%	78.26%	85.71%	60.00%	4.35.00%
92	97.00%	91.00%	97.83%	95.65%	100.00%	1.00%
92	98.00%	98.00%	98.91%	97.56%	100.00%	1.00%
92	95.00%	94.00%	95.62%	91.67%	100.00%	4.35%

Figuur 2.2: De juistheid van het neurale netwerk dat de MATLAB toolbox gebruikt heeft.

[2]

Classificatie via deep learning wordt bij Shingh et al. [3] in de medische sector nog gebruikt om microcalcificatie te categoriseren. Er worden drie klassen gedefinieerd: goedaardig zonder terugkommoment, goedaardig met terugkommoment en kwaadaardig. De data worden opgesplitst in 80% trainingdata en 20% testdata. De afbeeldingen worden bijgesneden tot het meest relevante gebied en vervolgens herschaald via interpolatie naar een 299 x 299 pixels afbeelding.

Om minder tijd te moeten besteden aan het trainen van het model worden eerder getrainde modellen gebruikt om het nieuwe model te trainen. Het transfermodel dat in deze studie gebruikt wordt, is

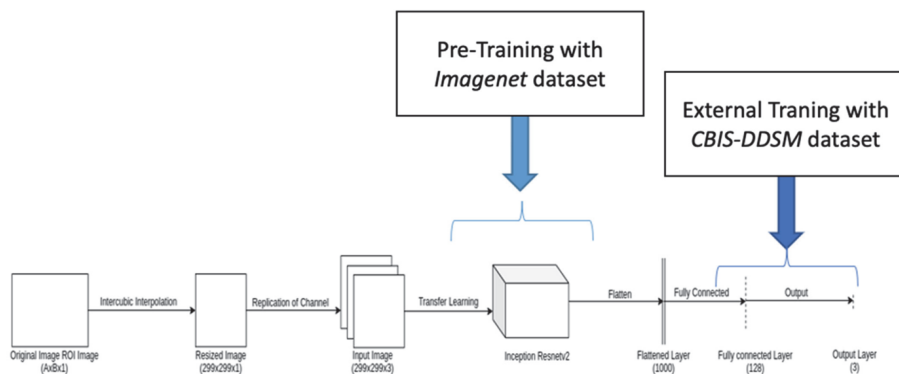
TABLE II: Accuracy for the non-toolbox implementation.

Accuracy	Precision	Recall	False Positives
62.00%	66.00%	45.00%	33.00%
77.00%	64.00%	45.00%	44.00%
78.00%	97.00%	56.00%	2.00%

Figuur 2.3: De juistheid van het neurale netwerk dat de MATLAB toolbox niet gebruikt heeft.

[2]

InceptionResNetV2 dat al getraind was met de ImageNet dataset. De eendimensionale uitgang van dit model wordt aan de ingang van een fully connected layer met 128 neuronen gegeven. In de fully connected layer wordt een rectified linear unit (ReLU) gebruikt en in de uitgangslaag wordt SoftMax gebruikt (figuur 2.4). Relu en softmax zijn activatiefuncties. Activatiefuncties beschrijven en beslissen welke neuronen er geactiveerd moeten worden. Het neurale netwerk wordt getest met vier verschillende verbeteringsalgoritmes of optimizers namelijk ADAM, ADAGrad, ADADelta en RMSProp. De hoogste trainingsaccuraatheid van 98% en validatie-accuraatheid van 94% worden bekomen met ADADelta.

**Figuur 2.4:** Opbouw van het model.

[3]

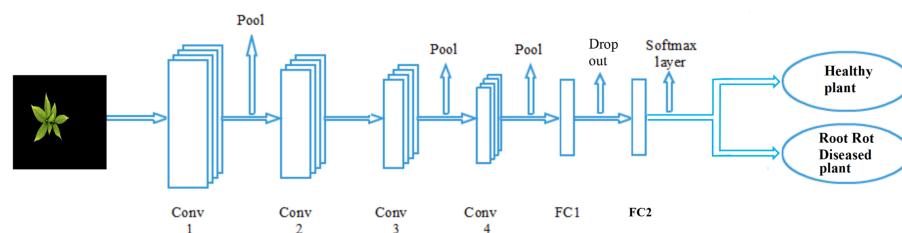
Kim et al. [4] gebruiken data-gedreven machinevisie ook buiten de medische sector. Om radioactief afval te sorteren volgens het materiaal waaruit het gemaakt is, worden 86 084 afbeeldingen manueel gecategoriseerd uit een video. Er worden ook extra klassen gedefinieerd voor wanneer er geen object aanwezig is, enkel maar handen aanwezig zijn of wanneer er helemaal niets te zien is.

De afbeeldingen in de dataset worden verschoven, geroteerd, herschaald en de ruis wordt weggefilterd om betere resultaten op te leveren. De afbeeldingen worden geconverteerd naar een resolutie van 512 x 512 pixels zodat er gemakkelijk aan data augmentation gedaan kan worden zoals herschaling, kleurverandering, rotatie, spiegeling en inzooming. Om overfitting te vermijden worden de afbeeldingen per batch naar het trainingsmodel gestuurd in plaats van één per één. ResNet wordt gebruikt om de accuraatheid te verhogen waardoor de gemiddelde accuraatheid van alle klassen 99,67% wordt.

Deep learning classificaties worden ook al gebruikt in de landbouwsector. Een studie van Jayapal et al. detecteert wortelrot bij de ginseng plant [5]. Wortelrot kan ontdekt worden door de bladeren van de plant te analyseren.

90% van de afbeeldingen worden gebruikt als trainingsdata, de andere 10% als validatie. De RGB afbeeldingen worden herschaald van een 3456 x 5184 pixel naar een 227 x 227 pixel afbeelding om vervolgens omgezet te worden in een HSV afbeelding. Ruis op de data wordt weggefilterd. Om extra data te creëren wordt data augmentation toegepast.

Tijdens de studie wordt er met verschillende transfermodellen geëxperimenteerd zoals AlexNet, VGG19, SqueezeNet, DarkNet19, ResNet18 en ResNet101. Er worden vier convolutilagen gebruikt waarvan drie aangevuld met een maxpoolinglaag (figuur 2.5). Deze lagen worden opgevolgd door twee fully connected lagen en een SoftMax laag. Dit blijkt het meest optimale aantal lagen te zijn. Bij minder lagen kan het model de juiste kenmerken niet goed genoeg herkennen en bij te veel lagen is er te veel overfitting. Bij vier convolutie- en drie maxpoolinglagen was er een accuraatheid van 90,73%. Het trainen van het model gebeurt bij een leersnelheid van 0,001. Een hogere of lagere leersnelheid geeft een minder goede accuraatheid. De beste mini-batchgrootte is 64 afbeeldingen.



Figuur 2.5: De opbouw van het model voor de detectie van wortelrot bij de ginseng plant.

[5]

2.2 Objectdetectie

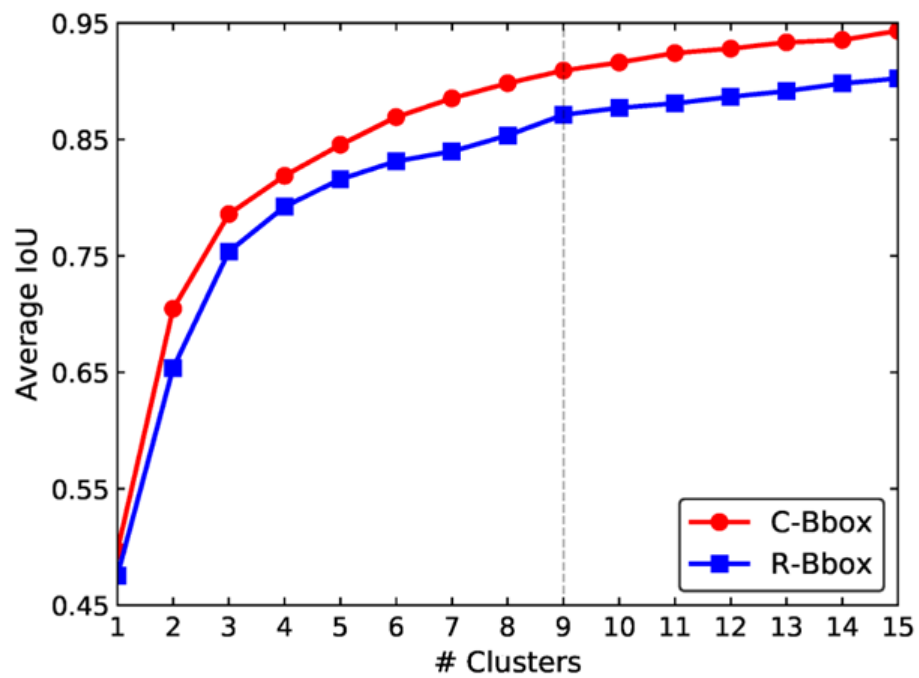
Li et al. [6] proberen druiventrossen te detecteren en te classificeren in zes klassen. Hierbij worden de ingangsdata opgesplitst in 70% trainingsdata en 30% testdata. Op de data wordt er willekeurig data augmentation toegepast om zo meer diversiteit in de data te krijgen.

YOLOv4-tiny met CSPdarknet53-tiny wordt gebruikt zodat er aan real-time detectie gedaan kan worden met een lage bewerkingsbelasting. Dit model wordt verfijnd naar een nieuw model genaamd Yolo-Grape. Dit levert een juistheid van 92,21% bij een snelheid van 12,34 ms.

Liu et al. [7] gebruiken deep learning om tomaten te detecteren in het veld. In deze studie wordt er data augmentation gebruikt op de trainingsdata. Het uiteindelijke model is afgeleid van YOLOv3. Op de basis van het YOLOv3 model wordt een Dense Convolutional Network geïmplementeerd voor een betere representatie van de kenmerken. Er wordt ook gekozen om een circulaire insluitende

regio te gebruiken in plaats van een rechthoekige om beter de ronde vorm van de tomaten te capteren.

Uit testresultaten blijkt dat een cirkelvormige insluitende regio een betere intersection over union (IoU) geeft dan een rechthoekige. Het uiteindelijke algoritme levert een juistheid van 94,75% op (figuur 2.6).



Figuur 2.6: Verschil in IoU bij een insluitende ronde (C-box) en insluitende rechthoek (R-box).

[7]

2.3 Lijndetectie

Bij Zhou et al. [8] wordt om suikerriet in zijn afzonderlijke delen te snijden niet data-gedreven machinevisie gebruikt om de snijlijn automatisch te detecteren. Doordat de snijlijnen zich op de dikkere en donkere stukken van het riet bevinden, kan dit goed visueel achterhaald worden. Het riet wordt op een zwarte achtergrond geplaatst en de beelden worden zo gemaakt dat het riet altijd van de linkerkant naar de rechterkant loopt (figuur 2.7 A).

De afbeeldingen worden omgezet naar grijswaarden door enkel de rode waarden van de RGB afbeelding te behouden (figuur 2.7 B). Om betere resultaten te bekomen worden storende factoren zoals korrelruis weggefilterd (figuur 2.7 C). Dit gebeurt aan de hand van een median filter. Een median filter zet een pixelwaarde om in de gemiddelde waarde van de pixel zelf en de omliggende pixels. Na het filteren moeten er zo veel mogelijk details van de randen behouden worden. Er bevinden zich veel holtes in de binaire afbeelding (figuur 2.7 D). Holtes zijn zones in de binaire afbeelding die een zwarte kleur hebben en omringd zijn door witte zones. Een groot deel van de holtes wordt dichtgemaakt aan de hand van een morfologische transformatie. De morfologische transformatie breidt eerst de witte delen van de afbeelding uit zodat de kleine zwarte vlekken verdwijnen (figuur 2.7 E). Dit zorgt ervoor dat het riet breder wordt dan het in de realiteit is. Vervolgens worden de zwarte delen van de afbeelding terug uitgebreid, zodat het riet terug zijn oorspronkelijke breedte heeft (figuur 2.7 F).

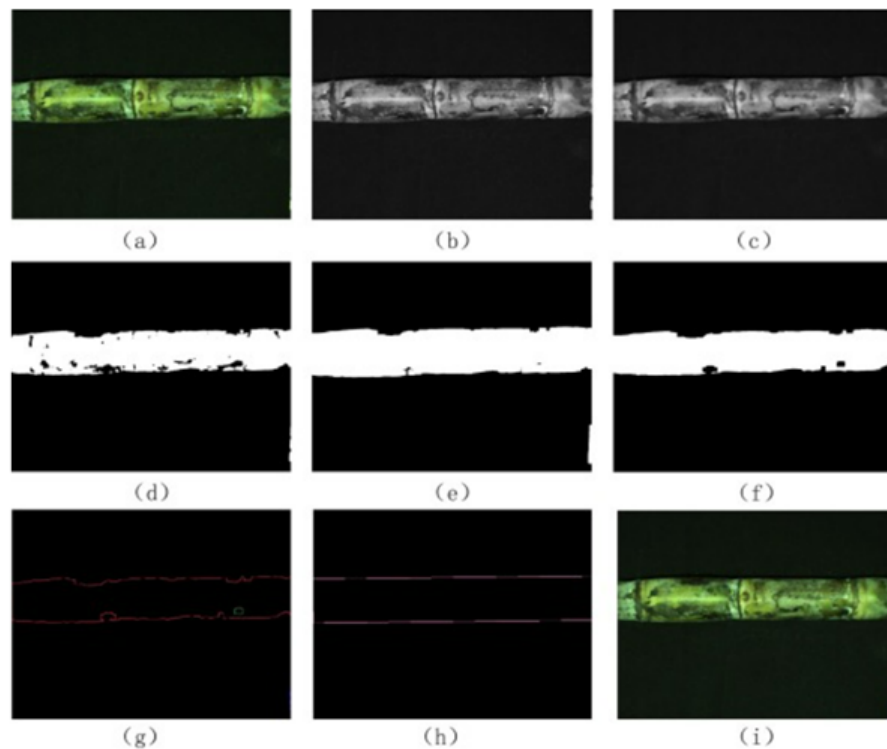
Op de achtergrond van de afbeelding is er geen nuttige informatie. De achtergrond kan zelfs storende factoren bevatten zoals afgebroken bladeren. Om deze storingen weg te werken moeten de afbeeldingen bijgesneden worden tot hun interessebereik. De OpenCV library wordt gebruikt om de contour van het riet te vinden en vervolgens om de kleinste mogelijke rechthoek waarin het riet zich bevindt te bepalen (figuur 2.7 G en H).

Doordat het riet zich altijd volgens dezelfde oriëntatie op de afbeelding bevindt en de snijlijn altijd loodrecht staat ten opzichte van het riet, zal de snijlijn altijd verticaal zijn. Een sobelfilter detecteert deze verticale lijnen.

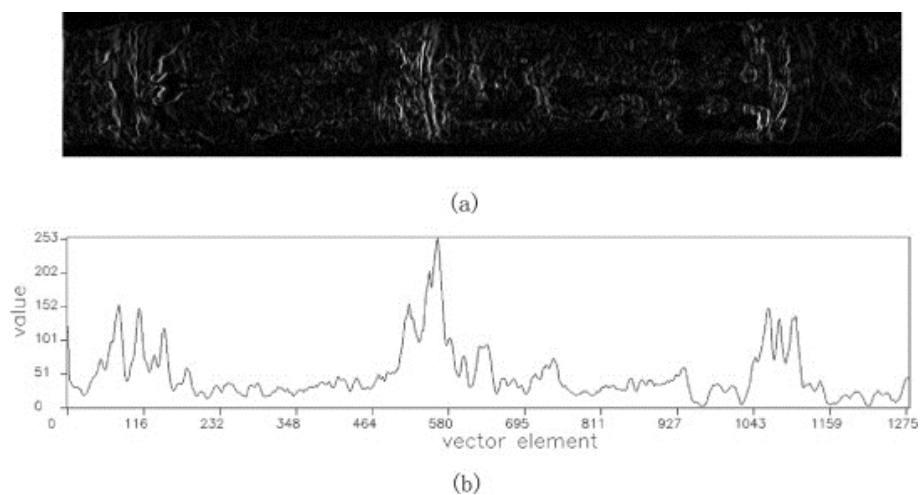
Er bevinden zich in onderstaand histogram (figuur 2.8) meerdere pieken. Om de snijlijnen te kunnen onderscheiden van andere pieken worden er enkele regels toegepast. De piek van een snijlijn moet ten eerste een minimale hoogte hebben. Ten tweede liggen de snijlijnen minstens veertig millimeter of 266 pixels uit elkaar. Tot slot wordt bij een groep van pieken telkens de grootste genomen. Het detecteren van de suikerrietsegmenten aan de hand van een sobelfilter had een gemiddelde succesvolle herkenningratio van 93% bij een gemiddelde snelheid van 0,539 seconden.

In de visindustrie is er ook al onderzoek gedaan door Azarmdel et al. [9] naar het bepalen van een snijlijn aan de hand van niet data-gedreven machinevisie. De snijlijn wordt bepaald aan de hand van de positie van de vinnen. In dit onderzoek worden forellen van verschillende groottes gebruikt. Bij het verzamelen van afbeeldingen passeren de forellen één voor één door een metalen doos om licht van buitenaf te vermijden. Alle afbeeldingen worden van bovenaf genomen.

Het algoritme werkt enkel als de vis zich in de juiste positie bevindt. Als de vis anders georiënteerd is, zal het algoritme niet correct werken. Om deze fouten te voorkomen zal eerst aan de hand van



Figuur 2.7: De verschillende stappen tijdens het bewerken van de suikerriet afbeelding. [8]



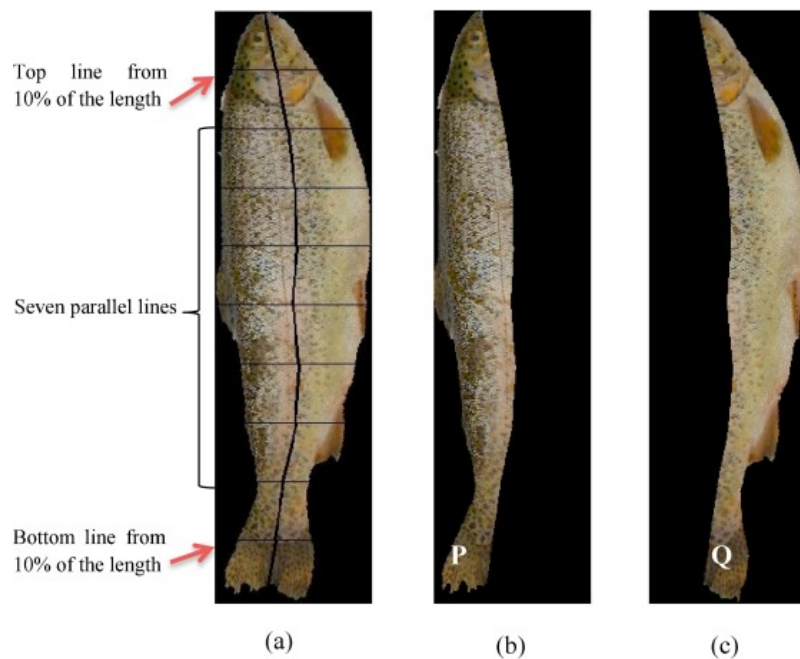
Figuur 2.8: Een histogram van de pixel intensiteit. [8]

beeldbewerking gecontroleerd worden of de vis zich in een geldige positie bevindt. Indien dit niet het geval is, zal heel het systeem stopgezet worden.

Om de vissen van de loopband te kunnen onderscheiden moet het contrast zo groot mogelijk blijven. Om de afbeelding naar grijswaarden om te zetten met oog op het grootste contrast, worden enkel de blauwe waarden uit de RGB-afbeelding behouden. De afbeelding wordt naar een

binaire afbeelding omgezet en de holtes worden via een morfologische transformatie verwijderd. De originele dimensies en de positionering van de vis worden bepaald om vervolgens de afbeelding te kunnen herpositioneren zodat de vis altijd perfect naar boven gericht is op de binaire afbeelding. Om de bewerkingsbelasting te verkleinen, wordt de afbeelding ook bijgesneden tot het insluitende kader van de vis.

De middellijn van de vis wordt bepaald aan de hand van de middelpunten van zeven parallelle lijnen (figuur 2.9). Aangezien de buik van de vis lichter van kleur is dan de rug kan aan de hand van de lichtintensiteit gecontroleerd worden of de vis zich in een ongeldige positie bevindt.

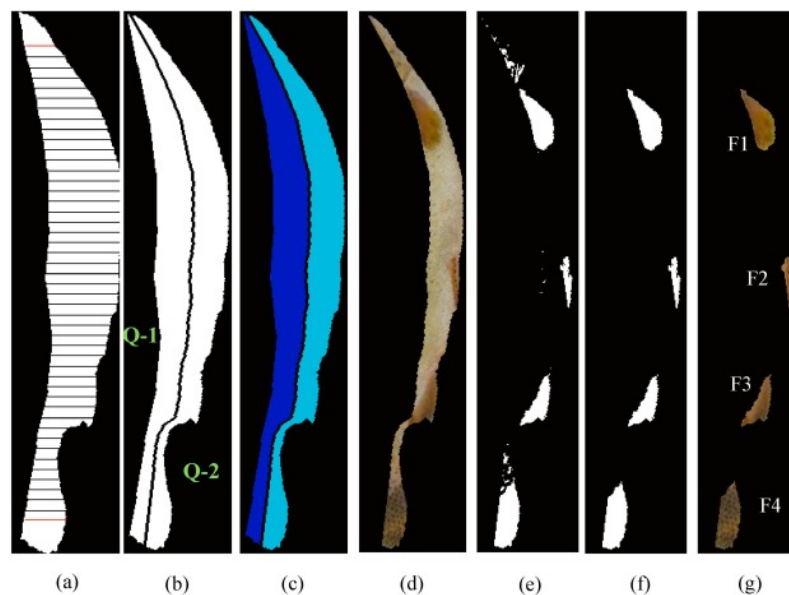


Figuur 2.9: Bepaling van de middellijn volgens de middelpunten van 7 parallelle lijnen.
[9]

Na het bepalen van de middellijn wordt de afbeelding opgesplitst in twee delen. Op het deel met de buikzijde (Q) wordt de middellijn op dezelfde manier bepaald (figuur 2.10). Aan de hand van deze tweede middellijn kan afbeelding Q opnieuw opgesplitst worden in Q1 en Q2. De vinnen die gedetecteerd moeten worden bevinden zich altijd in Q2.

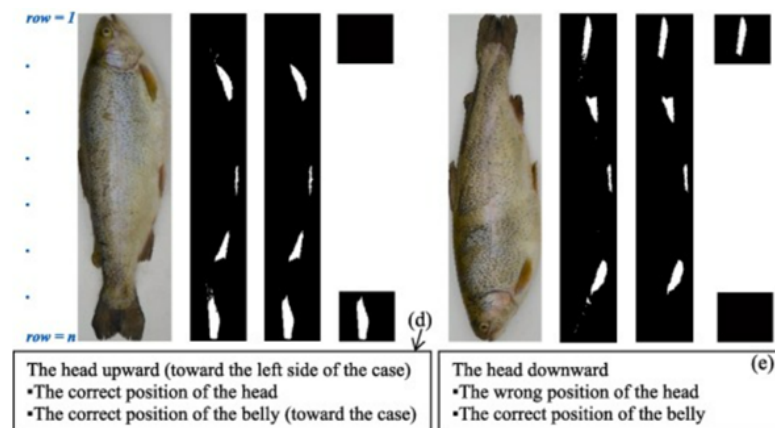
Om de vinnen te detecteren wordt opnieuw van de originele afbeelding gestart, maar er wordt nu enkel gekeken naar het deel Q2. De afbeelding wordt terug omgezet naar grijswaarden en vervolgens met een andere drempelwaarde omgezet naar een binaire afbeelding. Na het omzetten naar binaire waarden zullen enkel de vinnen nog wit gekleurd zijn.

Na het detecteren van de vinnen kan ook gecontroleerd worden of de vis met het hoofd naar boven of beneden gericht is. Als de vis met het hoofd naar boven gericht ligt, zal op de binaire afbeelding het bovenste deel van de afbeelding minder witte pixels hebben dan het onderste deel (figuur 2.11).



Figuur 2.10: De stappen om de 4 vinnen te segmenteren na het vinden van de middellijn.
[9]

De gevoeligheid van het detecteren van de buikvin (F2) is met 76% een stuk lager dan de andere vinnen, gemiddeld 86% (figuur 2.12). Dit komt doordat de buikvin een stuk kleiner is op de afbeelding aangezien de buik van de vis er voor een deel boven ligt. Ook de borstvin (F1) heeft een iets lagere sensitiviteit van 81% doordat deze vin licht doorzichtig is. De gemiddelde accuraatheid van de vier vinnen is 99,874%.



Figuur 2.11: In positie (d) ligt de vis in de juiste positie. Er zijn meer witte pixels onderaan de afbeelding dan wanneer de vis omgekeerd ligt.

[9]

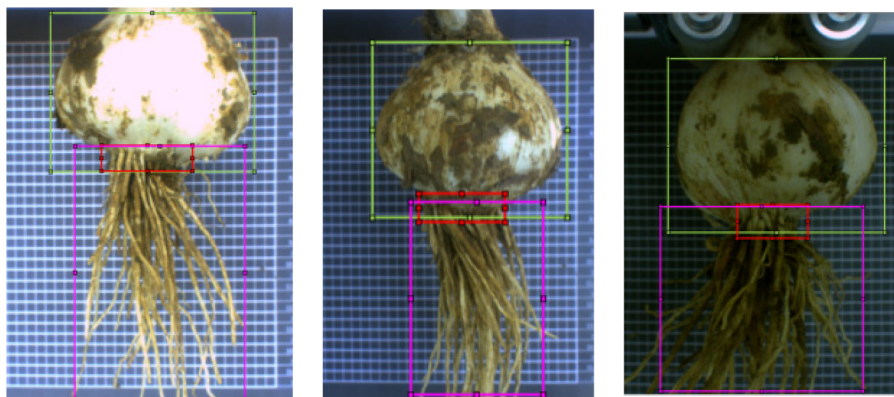
Er is al eerder een studie gedaan door Yang et al. die aan de hand van deep learning de snijlijn van de wortel van look probeert te vinden [10]. Er wordt deep learning gebruikt omdat er grote verschillen zijn tussen verschillende knollen.

	Pectoral Fin (F1)	Pelvic Fin (F2)	Anal Fin (F3)	Caudal Fin(F4)	Overall
Sensitivity%	81.251	75.760	88.546	98.662	86.054
Specificity%	99.964	99.968	99.975	99.951	99.965
Accuracy%	99.774	99.893	99.892	99.938	99.874

Figuur 2.12: De sensitiviteit, specificiteit en juistheid voor de detectie van elke vin.
[9]

Er worden 2500 afbeeldingen genomen waar er telkens maar één lookplant op afgebeeld staat. De RGB afbeeldingen worden omgezet in een "Yellow, Blue, Red" (YUV) kleurenruimte. Van de 2500 afbeeldingen worden er 500 gebruikt als testdata en de andere 2000 worden nogmaals opgesplitst in een trainingsset en een verificatieset met een verhouding van zeven tegenover drie.

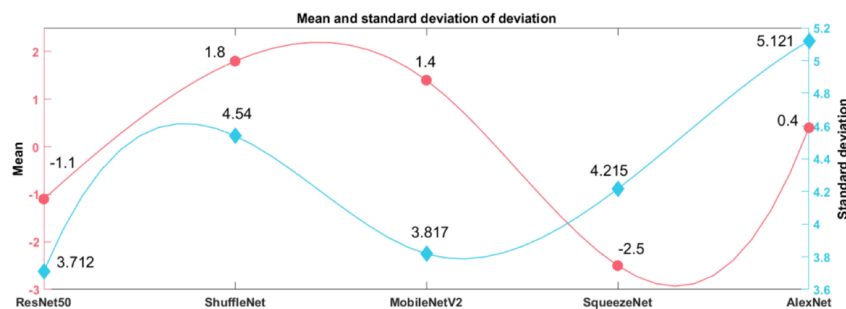
Om de performantie en de snelheid van het neurale netwerk te verhogen wordt de resolutie van de afbeeldingen voor het labelen verkleind naar 224×168 pixels. Een You Only Look Once (YOLO) model detecteert dan de insluitende rechthoeken van zowel de knol, de wortel als de wortelstam (figuur 2.13). Het detecteren van de wortelstam geeft vaak foute waarden. De insluitende rechthoek van de wortel gaat meestal door de knol en kan hierdoor niet gebruikt worden. De snijlijn van de wortel wordt bepaald door de onderste lijn van de insluitende rechthoek van de knol.



Figuur 2.13: De insluitende rechthoeken van de knol, de wortel en de wortelstam. De groene rechthoek duidt de knol aan, de paarse rechthoek de wortel en de rode rechthoek de wortelstam.
[10]

In de uiteindelijke trainingsdata worden enkel de knollen nog gelabeld. De afbeeldingen worden opgedeeld in twee regio's, knol en geen-knol. Om meer data te bekomen en om overfitting te vermijden zonder zelf nog manueel extra foto's te moeten nemen, wordt data augmentation toegepast. De trainingsset wordt uitgebreid van 1400 naar 5600 afbeeldingen. Het detecteren van de snijlijn moet zo snel mogelijk gebeuren. Daarom wordt er gekozen om met lichte convolutionele neurale netwerken te werken zoals SqueezeNet, ShuffleNet en MobileNetV2. Verschillende versies van het YOLO algoritme worden getest met verschillende backbone netwerken. Uit de verschillende modellen blijkt dat het YOLOv2-MobileNetV2 model het beste scoort op basis van accuraatheid en snelheid.

Het is belangrijk om in het achterhoofd te houden dat het beter is om te veel wortel over te laten dan dat er een stuk van de knol mee wordt afgesneden. Het is dus beter dat de voorspelde snijlijn een beetje reserveruimte laat. De gemiddelde afstanden van de snijlijnen ten opzichte van de knol die de verschillende modellen voorspellen worden met elkaar vergeleken (figuur 2.14). Hierbij is een voldoende grote gemiddelde waarde gewenst met en zo klein mogelijke standaarddeviatie. Van alle YOLOv2 modellen is het YOLOv2-MobileNetV2 model het meest betrouwbaar. Het model heeft een accuraatheid van 99,15% bij een snelheid van 0,0354 seconden.



Figuur 2.14: Het gemiddelde (bolvorm) en de standaarddeviatie (ruitvorm) van de verschillende netwerken. [10]

Zhao et al. [11] proberen met een neurale netwerk de leidende lijnen van een foto te vinden. Leidende lijnen in een afbeelding worden gebruikt om een mooie fotografische compositie te verkrijgen (figuur 2.15).

De leidende lijnen lopen oneindig lang door. Een oneindig lange lijn kan voorgesteld worden als de loodrechte op een lijn met een hoek θ ten opzichte van de x-as en een afstand r ten opzichte van de oorsprong. De oorsprong bevindt zich in het midden van de afbeelding (figuur 2.16). Om de leidende lijnen te detecteren worden de onderscheidende eigenschappen uit de afbeelding gehaald en wordt vervolgens een Hough transformatie op de eigenschappen toegepast. Deze transformatie zet de afbeelding om van het cartesisch coördinatenstelsel naar het polaire coördinatenstelsel. Het vertalen naar het polaire coördinatenstelsel zorgt ervoor dat de lijnstructuren compacter voorgesteld kunnen worden. Het neurale netwerk voorspelt de poolcoördinaten van de leidende lijnen.

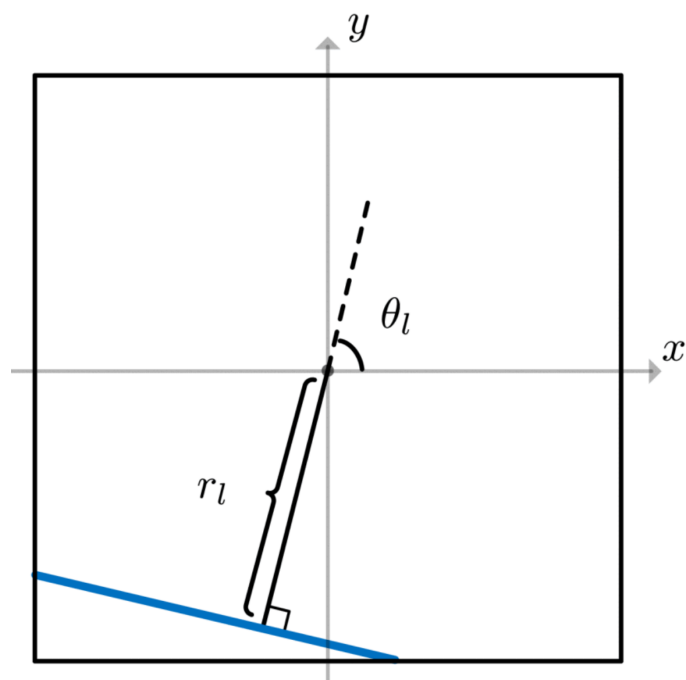
Om de lijnen te evalueren wordt er vergeleken met de werkelijke lijnen. Hoe groter de hoek tussen de twee lijnen en hoe groter de afstand tussen de middelpunten van de lijnen, hoe groter de fout. De modellen worden getraind met een leersnelheid van 0,0002 en een momentum van 0,9 over dertig epochs. De toegepaste methode behaalt een gemiddelde precisie van 0,819 en een gemiddelde terugroeping van 0,755.

Zhang et al. [12] willen aan de hand van lijnsegmentdetectie het 3D modelleren van door mens gemaakte omgevingen vergemakkelijken (figuur 2.17). Om dit te verwezelijken worden alle lijnsegmenten voorgesteld met een grafiek. Zo moet er niet langer naar de eindpunten van de lijn gezocht worden, maar in de plaats een grafiek geconstrueerd worden. Het gebruikte model is gebaseerd op ResNet50. Het model wordt getraind aan de hand van "stochastic gradient decent" (SGD) met een leersnelheid van 0,2 en een momentum van 0,9.



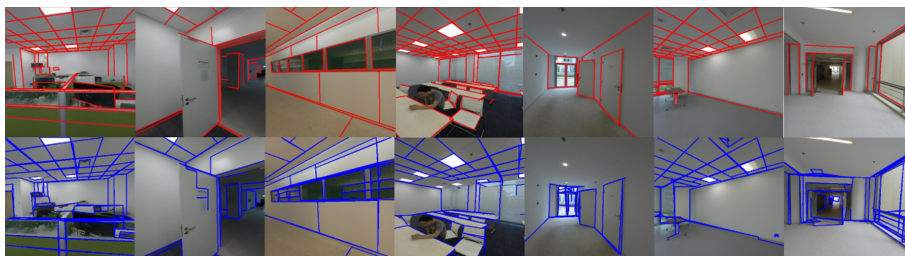
Figuur 2.15: De leidende lijnen van foto's.

[11]



Figuur 2.16: Voorstelling van een lijn met poolcoördinaten.

[11]



Figuur 2.17: De lijnsegmentdetectie voor door mens gemaakte omgevingen. De blauwe lijnen zijn de voorspelde lijnen, de rode zijn de werkelijke lijnen.

[12]

3

Methodologie

3.1 Camera set-up

De camera set-up is gemaakt uit item profielen (figuur 3.1). Het bestaat uit een verticale staaf die op een drie-puntsvoet geplaatst is. Aan de verticale staaf zijn twee zijtakken bevestigd. De zijtakken kunnen in hoogte versteld worden tot op een hoogte van 130 cm. De camera kan aan de zijtakken bevestigd worden en over de tak heen verschoven worden. De camera set-up is licht gevoelig aan oscillaties. Als de camera set-up oscilleert, dan is het moeilijker om mooie zuivere beelden te nemen. Om deze oscillaties tegen te gaan, wordt er gewerkt met een drie-puntsvoet. Aan de onderkant van elke poot is een verstelbare vijs bevestigd zodat de oppervlakte van de poot zo klein mogelijk is. Deze voet zorgt ervoor dat de constructie ook op oneffen terrein stabiel kan staan.

De dimensies van de constructie zijn bepaald aan de hand van de afmetingen van de snijmachine bij Inagro.

3.2 Camera specificaties

De gebruikte camera is de Intel RealSense D415 [13]. De foto's worden genomen met de RGB camera met een resolutie van 1280 x 720 pixels. De field of view of beeldhoek hierbij is 69° x 42°. Aangezien de grootste witloofkroppen ongeveer twintig centimeter lang zijn, moet de camera 26 cm van het witloof verwijderd zijn om deze in alle mogelijke posities volledig in beeld te krijgen.

De camera filmt met vijftien frames per seconde en heeft een kleurendiepte van 24 bits. Bij het nemen van snapshots zal de afbeelding in theorie $1280 * 720 * 24 = 2,76$ megabyte in beslag nemen. In werkelijkheid is het door compressie 1,33 megabyte.



Figuur 3.1: De gebruikte camera constructie voor het nemen van de beelden.

3.3 Data verzameling

Op 29 november 2022 werden tussen 8.30 uur en 17.15 uur in totaal 1463 afbeeldingen van witloof genomen. Aangezien het aantal beschikbare witloofkroppen beperkt is, wordt als volgt te werk gegaan. Eerst worden er 218 afbeeldingen genomen van witloof waar de wortel er nog volledig aan hangt. Het witloof wordt uit de bak gehaald zonder het eerst schoon te maken en het wordt op een willekeurige positie op een werktafel geplaatst. Aan elke afbeelding wordt manueel een andere naam gegeven. Na het nemen van een reeks foto's wordt de tafel afgekuist zodat er terug minder aarde en slijk op de tafel blijft liggen (figuur 3.2).

Na het nemen van deze foto's wordt een deel van de wortel met de hand afgesneden, maar niet de volledige wortel. Deze foto's dienen als referentie voor wanneer de wortel niet voldoende afgesneden zou zijn door de snijmachine. Alle foto's die genomen worden na het snijden, worden genomen boven de transportband zoals in figuur 3.3 om zo goed mogelijk de echte situatie, zoals bij de teler, na te bootsen. Dit is niet nodig voor de witloofkroppen waar de wortel nog aanhangt omdat de detectie van de snijlijn losgekoppeld is van de huidige situatie. Er zijn zowel foto's genomen van witloof waar nog een overdreven stuk van de wortel aanhangt als van witloof waar er nog maar een stukje van de wortel aanhangt. Soms staan er meerdere witloofkroppen op een afbeelding en soms staat er maar één op. Op het einde van de dag worden ook foto's genomen bij een minder

felle lichtinval. Al de foto's bij deze variaties worden in een aparte map bewaard.



Figuur 3.2: Opstelling voor het nemen van foto's van witloofkroppen met de wortel



Figuur 3.3: Opstelling voor het nemen van foto's van witloofkroppen met deels gesneden wortels.

Hierna worden op dezelfde manier foto's genomen van witloof dat perfect afgesneden wordt en daarna van witloof waarvan een deel van de krop wordt afgesneden. Er wordt telkens een deel van het witloof gebruikt dat ook gebruikt werd voor het nemen van foto's bij de vorige klasse. Daar bovenop wordt nieuw witloof gebruikt om de klasse te vervolledigen. Zo zijn er ondanks het beperkte aantal witloofkroppen toch genoeg om van elke klasse voldoende beelden te maken.

Er is tot slot nog een laatste klasse witloof waarvan de wortel schuin wordt afgesneden. Hierbij wordt er aan de ene kant een stuk wortel overgelaten en aan de andere kant een stuk van de krop afgesneden.

Er zijn 418 afbeeldingen gemaakt van witloof dat te weinig wordt afgesneden, 282 afbeeldingen van witloof dat te veel wordt afgesneden, 123 afbeeldingen van schuin afgesneden witloof en 421 afbeeldingen van witloof dat ideaal afgesneden wordt (Tabel 3.1). Doorheen de dag wordt het witloof dat op dat moment niet gebruikt wordt afgedekt van het licht zodat het niet kan verkleuren.

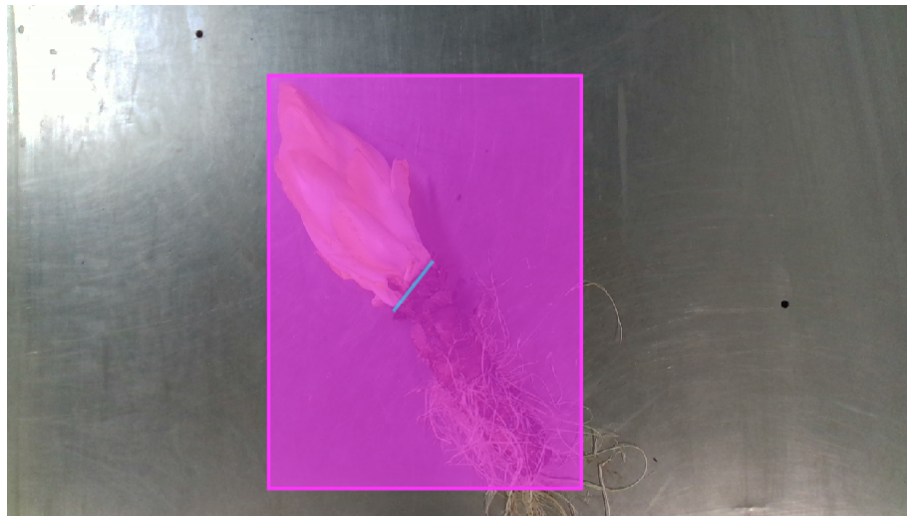
Foto klasse:	met de volledige wortel	te weinig gesneden	te veel gesneden	ideaal gesneden	schuin gesneden
Aantal afbeeldingen:	218	418	282	421	123

Tabel 3.1: Het aantal afbeeldingen per foto klasse.

3.4 Data labeling

Er zijn twee soorten data die gelabeld moeten worden. De eerste soort data labeling bestaat uit het toewijzen van de verschillende foto's aan de juiste klasse. De foto's kunnen tot twee verschillende klassen behoren. De klasse "Root" en "No Root". Alle witloofkroppen die schuin of te weinig werden afgesneden behoren tot de klasse "Root", omdat er daar altijd nog een stukje wortel aan de krop hangt. Alle kroppen die perfect of te veel werden afgesneden behoren tot de klasse "No Root". Een afbeelding wordt aan een klasse toegekend door het in de corresponderende map op de pc op te slaan.

De tweede soort data labeling wordt door Inagro volbracht. Alle foto's worden opgeladen op labelbox.com. Er is gekozen om met labelbox.com te werken omdat Inagro via deze website gemakkelijk alle data kan labelen zonder dat de data of een tool gedownload moet worden. Alles kan vanop afstand klaargezet en gecontroleerd worden. Via deze website wordt er rond alle witloofkroppen (de gesneden en de niet gesneden) een insluitende rechthoek getekend. Bij het witloof waar er nog niet gesneden werd, is ook de ideale snijlijn getekend (figuur 3.4). Na alle foto's gecontroleerd te hebben, worden de data geëxporteerd naar een JSON bestand. Voor elke mogelijke variatie van de data zoals de snijlengte en de belichting, wordt een apart JSON bestand aangemaakt. De insluitende rechthoeken worden volgens volgend formaat bewaard: {top, left, height, width}. Voor de snijlijn worden de twee uiterste punten bewaard. [{x1, y1}, {x2, y2}].

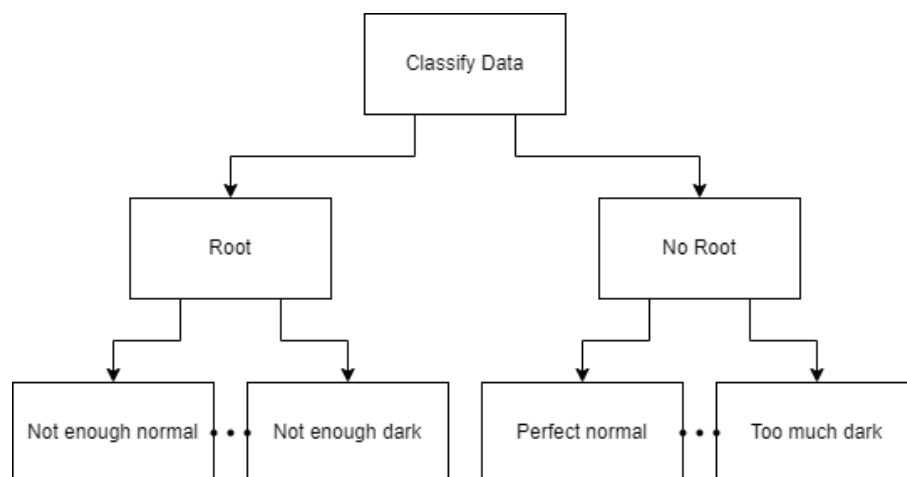


Figuur 3.4: De gelabelde data in labelbox.com waarop het witloof en de snijlijn aangeduid zijn.

3.5 Data inladen

Voor het trainen van het model om het witloof te kunnen classificeren worden de gebruikte data anders bewaard dan bij het model om het witloof te detecteren. Er is een andere structuur aangezien de data voor de verschillende toepassingen anders gelabeld zijn.

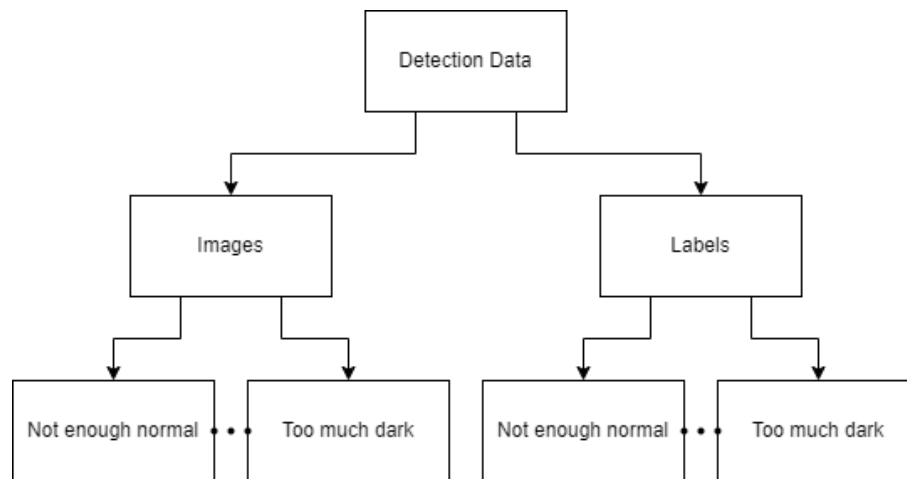
De classificatie dataset bevindt zich in een hoofdmap op de pc genaamd "Classify Data" (figuur 3.5). Binnen de hoofdmap bevinden zich twee submappen "Root" en "No Root". Alle mappen met foto's die behoren tot de de klasse "Root" worden in het mapje "Root" bewaard. Idem voor de klasse "No Root". Wanneer een afbeelding ingeladen wordt, kan het programma aan de hand van de locatie van de afbeelding zien tot welke klasse de afbeelding behoort.



Figuur 3.5: Opslagstructuur voor classificatie.

Voor de datasets nodig voor de detectie van het witloof en de detectie van de snijlijn worden alle mapjes met afbeeldingen onder een submap "Foto's" bewaard (figuur 3.6). Naast de submap

"Foto's" is er ook nog een tweede submap "Labels". Hier worden alle JSON bestanden met de bijhorende labels bewaard. Het is belangrijk dat alle JSON bestanden exact dezelfde naam hebben als de map waarin de bijhorende afbeelding zich bevindt. De afbeelding "perfect_donker_366.png" bevindt zich in een map "perfect_donker" onder de map "images". Als het programma de insluitende rechthoekcoördinaten wil verkrijgen, zal het programma kijken waar de afbeelding bewaard wordt. Als het programma ziet dat de afbeelding zich in het mapje "perfect_donker" bevindt, zal het daarom in "perfect_donker.json" zoeken naar de juiste coördinaten.



Figuur 3.6: Opslagstructuur voor detectie.

Bij het inladen van de data worden al bestandspaden van de afbeeldingen in een lijst geplaatst. Een volledig bestandspad kan er als volgt uitzien:

"..\..\..\Foto's classify\No Root\apart perfect\apart_perfect267.png".

Het pad is relatief ten opzichte van de locatie van het programma. De bestandspaden in die lijst worden vervolgens in een willekeurige volgorde geplaatst. De lijst wordt vervolgens opgesplitst in training, validatie en testdata volgens een 70, 15, 15 splitsing.

Doordat de afbeeldingen willekeurig verdeeld worden in trainings- en testdata, worden bij elke nieuwe training andere data gebruikt. De afbeelding "apart_perfect267.png" kan om model A te trainen gebruikt zijn als trainingsdata en om model B te trainen gebruikt zijn als testdata. Om achteraf nog extra testen met het model te kunnen doen, wordt in een tekstbestand bijgehouden welke data bij welk model gebruikt worden om te testen, te valideren of te trainen.

3.6 Preprocessing

Bij het inladen van de data wordt elke afbeelding ingelezen met openCV. De afbeeldingen worden ook omgezet van "blue, green, red" (BGR) formaat naar RGB. Nu wordt de corresponderende klasse aan elke afbeelding toegekend door middel van een index in plaats van een naam. Index 0 staat voor de klasse "No Root", index 1 staat voor de klasse "Root". De insluitende rechthoeken worden ook van formaat veranderd. Ze worden van {top, left, height, width} omgezet in het [xmin, ymin, xmax, ymax] formaat waarbij xmin en ymin de coördinaten zijn van de linkerbovenhoek

van de insluitende rechthoek en $x_{\max}$ en $y_{\max}$ de coördinaten zijn van de rechteronderhoek.

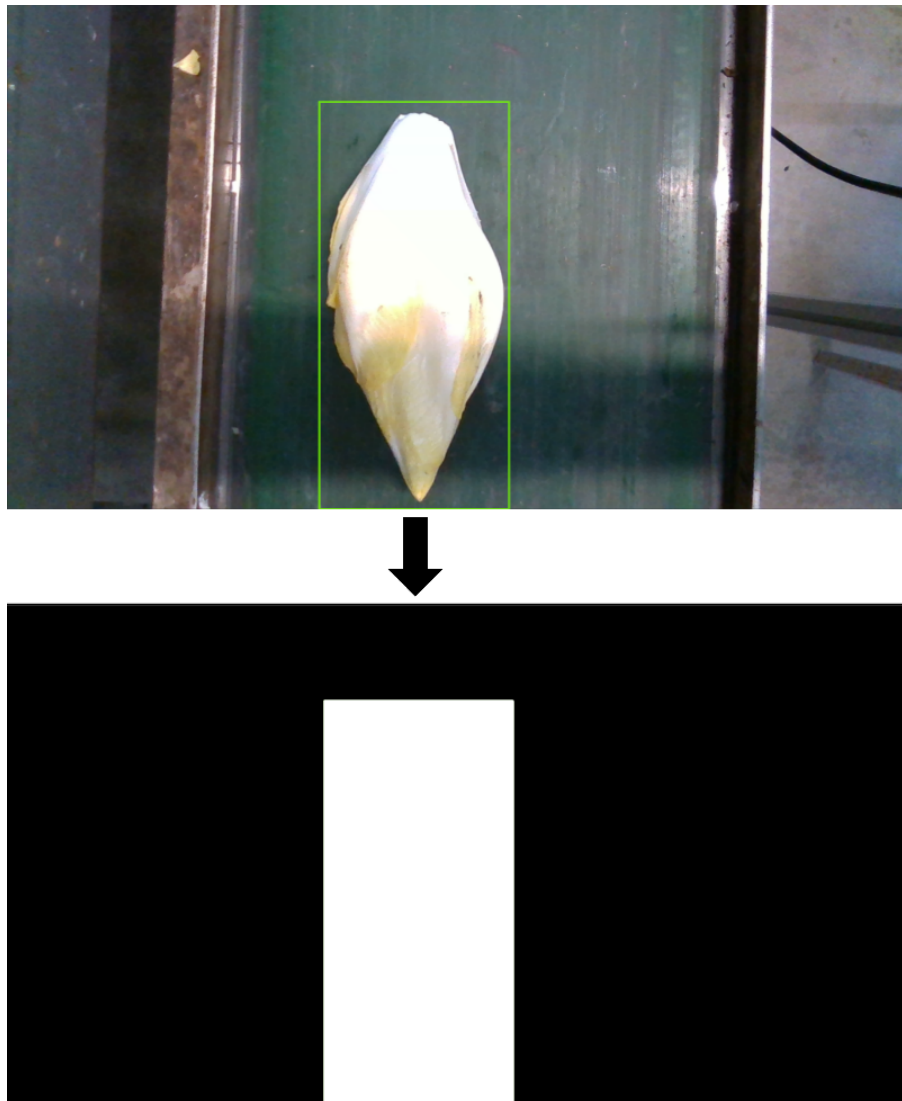
Om het model goed te kunnen trainen worden de afbeeldingen gepreprocessed. Alle afbeeldingen worden in een tensor gezet en geconverteerd naar floating point getallen. Vervolgens worden de afbeeldingen herschaald naar een 244 op 244 pixels afbeelding. Tot slot worden de afbeeldingen genormaliseerd volgens een gemiddelde [0.485, 0.456, 0.406] en een standaardafwijking van [0.229, 0.224, 0.225]. Deze waarden zijn afkomstig van de Pytorch documentatie voor ResNet18 [14].

Omdat de afbeeldingen herschaald worden is het belangrijk dat de insluitende rechthoeken ook op dezelfde manier herschaald worden. Om dit te bekomen wordt van de insluitende rechthoek een mask gemaakt. Een mask is een binaire afbeelding waarbij de pixels binnen de insluitende rechthoek wit zijn en alle andere pixels zwart (figuur 3.7). De mask wordt dan via openCV herschaald naar dezelfde afmetingen als de bewerkte afbeelding. Tot slot wordt de mask terug omgezet in een lijst met de aangepaste rechthoekcoördinaten (figuur 3.8 en 3.9).

3.7 Model

Voor het classificatiemodel wordt ResNet18 gebruikt. Het model heeft twee uitgangen. De eerste uitgang geeft aan hoe zeker het model is dat de afbeelding tot klasse "No Root" behoort en de tweede uitgang geeft een indicatie hoe zeker het model is dat de afbeelding tot de klasse "Root" behoort. De klasse met de hoogste zekerheid wordt als voorspeld resultaat beschouwd. De keuze om een model te maken op basis van ResNet18 is genomen omdat het model in de paper van Jayapal et al. [5] op basis van ResNet18 een hoge trainingssnelheid en een goede accuraatheid heeft. Er wordt een leersnelheid van 0,001 en een batchgrootte van 64 gebruikt omdat deze waarden in de paper de beste resultaten opleveren. Jayapal et al. gebruiken net als het classificatiemodel voor witloof kruisentropy om de loss te berekenen en een SGD optimizer met een momentum van 0,9.

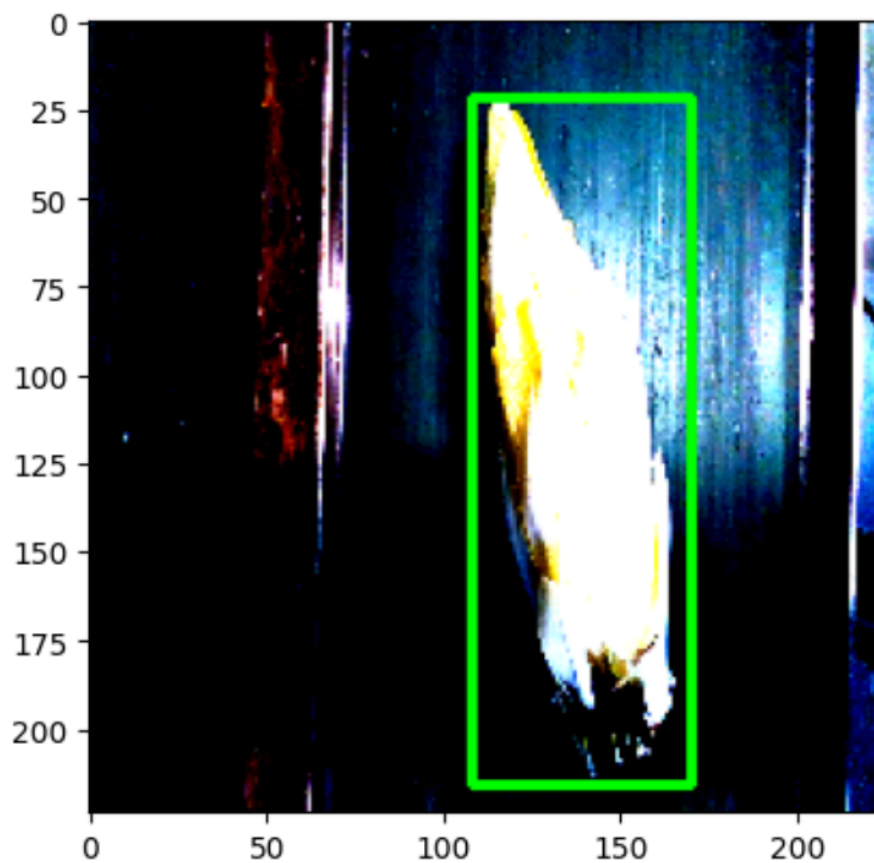
Het detectiemodel gebruikt net als het classificatiemodel ResNet. Er is zowel met ResNet18 als ResNet50 geëxperimenteerd. Het model heeft vier uitgangen namelijk $x_{\min}$, $y_{\min}$, $x_{\max}$ en $y_{\max}$. De uitgangen geven de x en y coördinaat van de linkerbovenhoek van de insluitende rechthoek en de x en y coördinaat van de rechteronderhoek.



Figuur 3.7: De omzetting van een bounding box naar een mask.



Figuur 3.8: Originele afbeelding (apart.schuin.83.png).



Figuur 3.9: Afbeelding na preprocessing met aangepaste insluitende rechthoek

4

Resultaten

4.1 Duiding van de deelresultaten

In overleg met de promotor en de co-promotoren wordt deze masterproef met als titel "Een computervisie gebaseerde aanpak voor het bepalen van de snijlijn van witloof" aangevuld met twee belangrijke elementen.

Ten eerste zou er oorspronkelijk enkel naar de ideale snijlijn van witloof gezocht worden, maar uiteindelijk wordt er ook gewerkt aan een model voor de classificatie van gesneden witloof. Dit moet na het snijden leiden tot een substantiële vermindering van de economische verliezen bij de telers. De economische verliezen zouden moeten verminderen omdat niet langer elk witloofstaal een manuele controle nodig heeft, wat leidt tot een besparing in personeelskosten en een tijdswinst.

Ten tweede kan de detectie van het witloof belangrijk zijn voor het bepalen van de optimale snijlijn en het classificeren van gesneden witloof. Door het witloof eerst te detecteren, kan er gericht gezocht worden naar de optimale snijlijn en kunnen er meerdere witloofkroppen tegelijk geclassificeerd worden.

Omdat de twee voorgaande processen meer tijd dan verwacht in beslag nemen, beperkt de ontwikkeling van de optimale snijlijn zich tot een uitgeschreven plan van aanpak. De nadruk van dit derde deelresultaat ligt op de wiskundige analyse om de snijlijn voor te stellen met zo weinig mogelijk vrijheidsgraden.

4.2 Classificatie van gesneden witloof

Voor het trainen van het classificatie-algoritme worden er 681 afbeeldingen van gesneden witloof gebruikt als trainingsset, 146 afbeeldingen als validatieset en 147 als testset. Tijdens het trainen

wordt de trainingsset verdeeld in batches van 64 afbeeldingen en wordt er voor tien epochs getraind. Er wordt niet langer dan tien epochs getraind omdat uit de resultaten blijkt dat het model na tien epochs niet veel meer verbetert. De optimizer maakt gebruik van SGD. Er wordt een leersnelheid van 0,001 en een momentum van 0,9 gebruikt. De loss wordt bepaald via kruisentropie.

Als er naar de loss curves (figuur 4.3) gekeken wordt, dan daalt de loss sterk voor de eerste twee epochs en daalt de loss nog maar licht vanaf de zevende epoch. Dit is een aanvaardbare vorm voor de curve, aangezien dit impliceert dat het trainen in het begin veel vooruitgang boekt en dat er pas gestopt wordt met trainen wanneer het model geen grote verbeteringen meer kent, die zich zouden vertalen in een verdere daling van de curve.

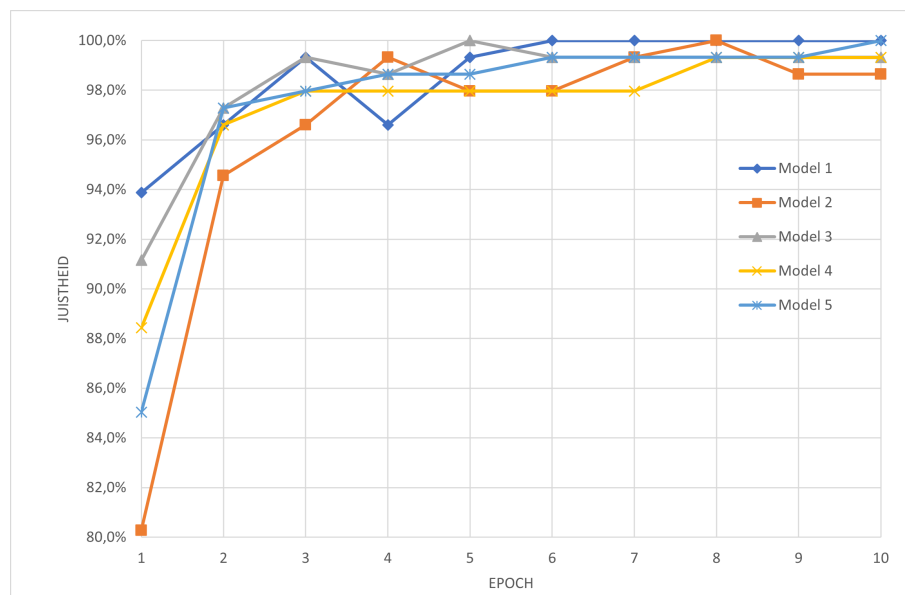
Het model wordt meermaals getraind. In figuur 4.1 en 4.2 wordt de juistheid per epoch geplot voor tien opeenvolgende trainingssessies. De verschillende modellen zijn met andere trainings-, validatie- en testdata getraind en alle modellen gebruiken dezelfde parameters. Model 4, 6, 9 en 10 halen na tien epochs nooit een juistheid van 100% op de testset. De maximale juistheid die ze bereiken over de tien epochs heen is 99,3%. Bij een juistheid van 99,3% wordt op een testset van 147 afbeeldingen één afbeelding fout ingedeeld (tabel 4.1). De tien modellen hebben een gemiddelde maximale juistheid van 99,72%.

Model 4	Klasse "Root"	Klasse "No Root"
Voorspeld als "Root"	50	1
Voorspeld als "No Root"	0	96

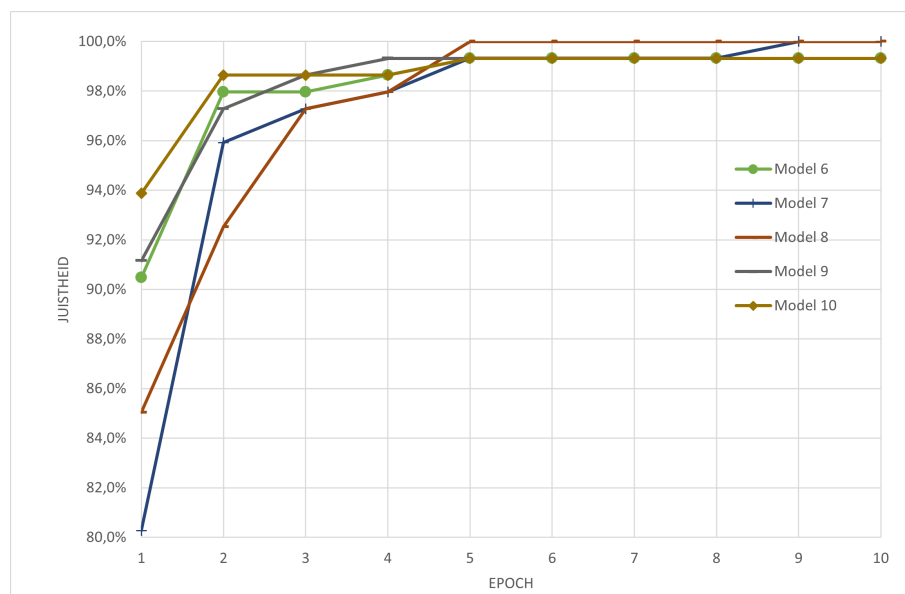
Tabel 4.1: Verwarringsmatrix van model 4 bij de hoogste juistheid.

Op figuur 4.1 is te zien dat model 2 een juistheid van 100% bereikt na acht epochs. Een model wordt bewaard wanneer de validatie loss lager is dan de vorige epoch. De validatie loss van model 2 voor epoch negen is lager dan de loss voor epoch acht, hierdoor wordt het model bewaard ook al is de juistheid voor de achtste epoch beter.

Op figuur 4.2 is te zien dat model 8 een juistheid van 100% bereikt vanaf epoch vijf en de juistheid niet meer zakt. Dit wil niet zeggen dat het model niet verder traint. Vanaf epoch vijf tot en met epoch tien blijft de validatie loss zakken. Het model is naarmate de epochs vorderen steeds zekerder over de voorspelde resultaten. Voor een afbeelding van klasse "No Root" kunnen de volgende fictieve uitgangen verkregen worden: uitkomst 1 = [0.971 0.243] en uitkomst 2 = [0.952 0.562]. Elke uitkomst bevat twee waarden. De eerste waarde toont de zekerheid aan voor klasse 0 ("No Root") en de tweede waarde toont de zekerheid voor klasse 1 ("Root"). Bij beide uitkomsten is de waarde van klasse 0 groter dan klasse 1, dus beide uitkomsten zullen de afbeelding juist classificeren als "No Root". Uitkomst 1 is wel een betere uitkomst dan uitkomst 2 omdat er een groter verschil is tussen beide klassen. Het model is bij uitkomst 1 dus zekerder over zijn antwoord dan bij uitkomst 2.



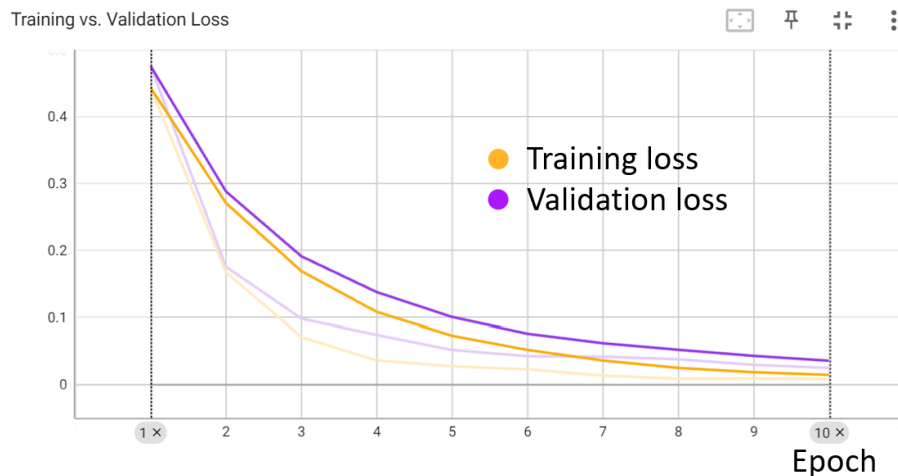
Figuur 4.1: Juistheid per epoch voor de eerste vijf getrainde modellen.



Figuur 4.2: Juistheid per epoch voor de laatste vijf getrainde modellen.

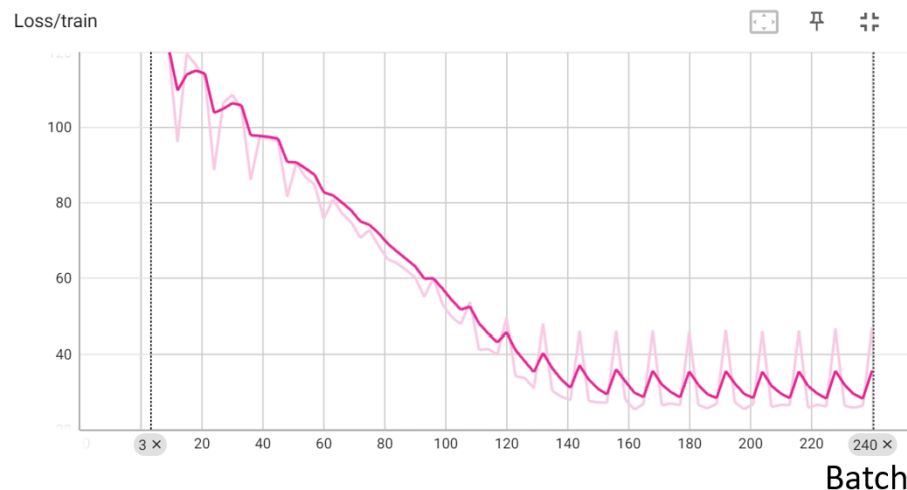
4.3 Detectie van witloof

Om het witloof te detecteren wordt net zoals bij het classificatie-algoritme eerst getest met een ResNet18 model. De afbeeldingen worden in batches van 64 afbeeldingen verdeeld en er wordt voor twintig epochs getraind. Er wordt een SGD optimizer gebruikt met een leersnelheid van 0,001 en een momentum van 0,9. Om de loss te bepalen wordt de L1 loss functie gebruikt. L1 loss is de som van de absolute waarden van het verschil tussen de echte coördinaten en de voorspelde. Li et al. [6] en Liu et al. [7] werken ook met een leersnelheid van 0,001.



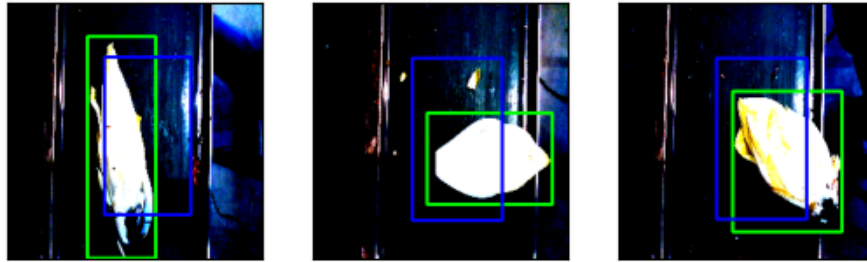
Figuur 4.3: De training en validatie loss van het classificatie-algoritme. De lichtgekleurde curves tonen de effectieve waarden van de loss, de donkere curves zijn afgevlakt met 60%.

Bij dit model vertoont de loss curve (figuur 4.4) bij de eerste epochs geen sterke daling. Dit kan wijzen op een te kleine leersnelheid. Het model kan vanaf ongeveer de elfde epoch niet meer verbeteren. De training loss bevindt zich dan telkens rond de twintig, maar bij de laatste batch van elke epoch schiet de loss terug naar ongeveer veertig. Op het internet zijn weinig vergelijkbare curves gevonden. Volgens een machine learning cursus van Google [15] kan dit fenomeen te verklaren zijn door herhalingen in de data zelf. Omdat het model ongeveer bij de elfde epoch geen verdere verbetering meer ondervindt, wordt er slechts voor twintig epochs getraind zodat de herhaling in de loss curve gezien kan worden zonder dat er overdreven lang getraind moet worden. Op figuur 4.5 is te zien dat dit zorgt voor een slechts heel beperkte nauwkeurigheid van de voorspelde insluitende rechthoeken. De groene rechthoek is de werkelijke insluitende rechthoek, de blauwe rechthoek is de voorspelde rechthoek volgens het neurale netwerk.



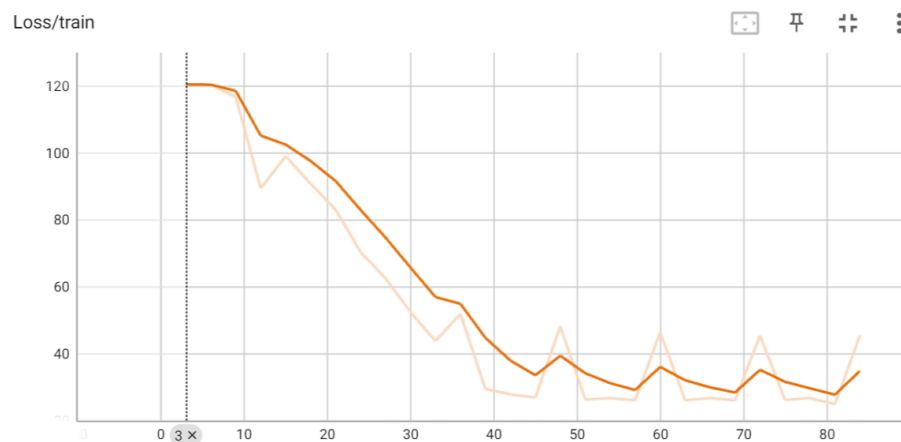
Figuur 4.4: Training loss van het detectiemodel met ResNet18 en een leersnelheid van 0,001.

Om dit probleem op te lossen wordt geprobeerd om over te schakelen naar ResNet50. ResNet50



Figuur 4.5: Resultaat na 20 epochs met ResNet18. De blauwe rechthoek is het voorspelde resultaat, de groene is de werkelijke insluitende rechthoek.

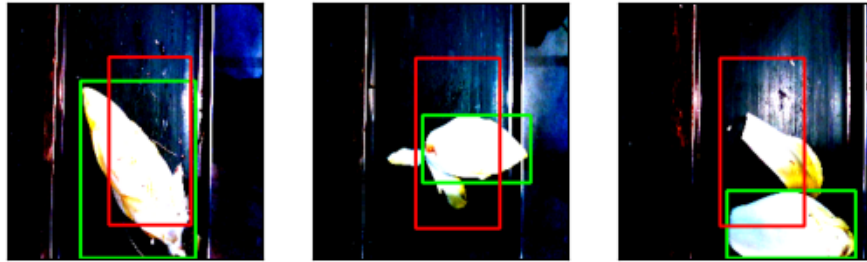
heeft meer lagen dan ResNet18. Er wordt ook geprobeerd om een hogere leersnelheid te gebruiken van 0,004. De loss curve (figuur 4.6) is nu bij de eerste epochs steiler en dus beter, maar loopt nog steeds vast na enkele epochs. Hierdoor kan ook dit model niet voorspellen waar het witloof zich bevindt (figuur 4.7). Het gebruik van een scheduler om bij de eerste epochs een hoge leersnelheid te gebruiken en bij de latere epochs een lagere leersnelheid te gebruiken lost dit probleem van een lage IoU ook niet op. Het experimenteel aanpassen van het momentum helpt ook niet. Het model werkt niet optimaal met de huidige versie gebaseerd op ResNet50. De manier waarop het model opgebouwd is kan deze problemen mogelijks veroorzaken. Het gebruik van extra of andere lagen kan overwogen worden om tot betere resultaten te komen.



Figuur 4.6: Training loss van het detectiemodel met ResNet50 en een leersnelheid van 0,004.

4.4 Detectie van de optimale snijlijn

De snijlijn van witloof kan nog niet gedetecteerd worden. De data zijn wel al gelabeld. Doordat de data volgens de structuur $[\{x1, y1\}, \{x2, y2\}]$ gelabeld zijn, zijn er veel vrijheidsgraden voor het tekenen van de snijlijn. In figuur 4.8 wordt de lijn waarlangs er gesneden moet worden voorgesteld met de paarse lijn. De paarse lijn wordt in de labels voorgesteld door de twee rode punten. Er zijn echter nog een oneindig aantal andere combinaties van punten die deze lijn kunnen voorstellen zoals de groene punten op 4.8. Als de rode punten gebruikt worden om een model te trainen,



Figuur 4.7: Resultaat na 6 epochs met ResNet50.

zal het model proberen om een combinatie van punten te voorspellen die zo dicht mogelijk bij de rode punten liggen. Volgens deze methode zou het model beter geëvalueerd worden voor het voorspellen van de blauwe punten, dan voor de groene, terwijl de snijlijn beter voorgesteld wordt door de groene punten.



Figuur 4.8: De voorstelling van een snijlijn aan de hand van twee punten. De groene en de rode punten geven de juiste snijlijn voor. De blauwe punten liggen dicht bij de rode punten, maar stellen een heel andere snijlijn voor.

Door de lijn voor te stellen als een oneindige lijn moet het model maar twee waarden genereren en is er maar één enkele combinatie van waarden die de juiste lijn kunnen voorstellen. Er zijn twee manieren om dit te doen. De eerste manier is door de gelabelde lijn door te trekken zodat de lijn tot aan de grenzen van de afbeelding loopt. Dit kan bereikt worden door de richtingscoëfficiënt van de lijn te berekenen. $Richtingscoëfficiënt = \Delta Y / \Delta X$. Doordat alle afbeeldingen hetzelfde formaat hebben van 1280 x 720 pixels kan y_1 gelijkgesteld worden aan 0 en y_2 aan 720. De nieuwe x_1 en x_2 kunnen berekend worden aan de hand van y_1 , y_2 en de richtingscoëfficiënt. Nu moet het algoritme enkel nog maar de nieuwe x_1 en x_2 voorspellen. Figuur 4.9 stelt deze methode

visueel voor. Het is mogelijk dat deze methode niet werkt indien de snijlijn perfect horizontaal is ten opzichte van de x-as, want dan kunnen de y waarden nooit 0 en 720 zijn.

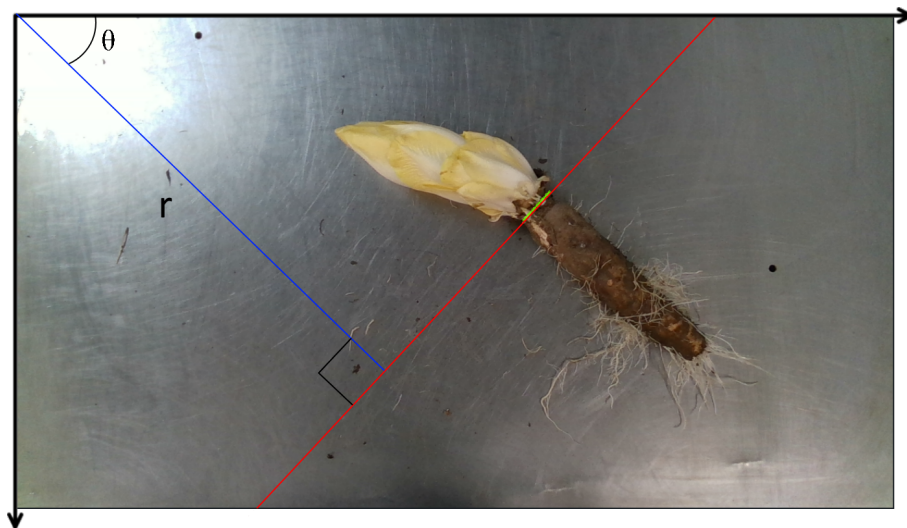
Met de tweede methode is het de bedoeling om de lijn voor te stellen met poolcoördinaten. De poolcoördinaten stellen een lijn op die loodrecht op de snijlijn staat en door de oorsprong gaat. Net als de vorige methode wordt hiervoor de gelabelde lijn doorgetrokken. De vergelijking van deze lijn wordt bepaald met de formule $y = \frac{y_2 - y_1}{x_2 - x_1} \cdot x - x_1 + y_1$. De afstand R van de lijn tot de oorsprong kan berekend worden met de formule $R = \frac{|w|}{\sqrt{u^2 + v^2}}$. De waarden van u, v en w worden gehaald uit de vergelijking van de rechte door deze om te zetten in de vorm $u \cdot x + v \cdot y + w = 0$.

Nu de afstand tot de oorsprong gekend is kan het snijpunt (S_x , S_y) van de snijlijn en de loodrechte op de snijlijn berekend worden. Het product van de richtingscoëfficiënt van de loodrechte en de snijlijn is gelijk aan -1. De richtingscoëfficiënt van de snijlijn werd al berekend bij de eerste methode. Het snijpunt wordt bepaald aan de hand van de stelling van Pythagoras: $R^2 = (S_x)^2 + (S_y)^2$ en de definitie van de richtingscoëfficiënt: $Richtingscoëfficiënt = \frac{\Delta y}{\Delta x} = \frac{S_y - 0}{S_x - 0}$. Met het snijpunt kan de hoek van de loodrechte θ ten opzichte van de x-as berekend worden. $\theta = \arctan(\frac{S_y}{S_x})$. Het algoritme moet nu proberen om R en θ te voorspellen. Figuur 4.10 toont visueel aan hoe de snijlijn als poolcoördinaten kan voorgesteld worden.

Na het omzetten in poolcoördinaten kan geprobeerd worden om de poolcoördinaten te voorspellen. Een voorstel is om eerst te proberen om een objectdetectie-algoritme te gebruiken. De uitgang van het objectdetectiemodel wordt aangepast waardoor het niet langer vier uitgangen heeft voor de insluitende rechthoek coördinaten, maar slechts twee voor de lengte R en de hoek θ .



Figuur 4.9: Voorstelling van een lijn met 2 waarden x_1 en x_2 door de snijlijn over de volledige afbeelding te laten lopen. De groene lijn is de gelabelde lijn. De rode lijn is de doorgetrokken lijn tot aan de grenzen van de afbeelding.



Figuur 4.10: Voorstelling van een lijn met 2 waarden, theta en r . De groene lijn is de gelabelde snijlijn. De rode lijn is de doorgetrokken snijlijn. De blauwe lijn is de loodlijn op de snijlijn door de oorsprong.

5

Conclusies

5.1 Conclusies

Er zijn een heleboel data verzameld. Er zit variatie in de data op vlak van positie van het witloof, belichting, storingen in de achtergrond en kleine verplaatsingen van de camera. Er zijn in totaal 1463 afbeeldingen van witloof genomen. Er kunnen later nog extra data gecreëerd worden met data augmentation. De data zijn met de hulp van Inagro genomen en gelabeld.

Het witloof wordt geclassificeerd als witloof dat nog meer bijgesneden moet worden en witloof dat niet meer bijgesneden moet worden met een gemiddelde juistheid van 99,72%. De modellen zijn ontworpen aan de hand van ResNet18. De modellen worden voor tien epochs getraind en een SGD optimizer is gebruikt met een leersnelheid van 0,001 en een momentum van 0,9. Dit model kan telers ten goede komen in hun streven naar kostenbesparingen.

Er werd een model om witloof te detecteren ontwikkeld, maar het model werkt nog niet optimaal. De insluitende rechthoek coördinaten worden al succesvol getransformeerd. Na vier epochs stopt het model met leren en verbetert het resultaat niet meer. Er wordt geëxperimenteerd met een scheduler om de leersnelheid aan te passen om zo een betere loss curve te bekomen. De modellen gebruiken ResNet18 en ResNet50.

Het detecteren van de snijlijn werd theoretisch, maar nog niet in de praktijk uitgewerkt. Via de literatuur is een plan opgesteld om de snijlijndetectie uit te werken. De snijlijn zou voorgesteld worden aan de hand van poolcoördinaten en het snijlijndetectie probleem wordt gezien als een objectdetectie probleem.

5.2 Toekomstig werk

Het witloof wordt voor de classificatie momenteel enkel onderverdeeld in twee klassen: "Root" en "No Root". In de realiteit is er nog een derde mogelijkheid die zich kan voordoen. Als het algoritme gebruikt wordt met live beelden, dan kan het zijn dat er beelden binnen komen van een lege loopband of beelden met andere objecten op. Het classificatiemodel kan nog verder uitgebreid worden zodat er een derde klasse "No Chichory" is. Dit kan momenteel nog niet ontwikkeld worden aangezien er hier nog geen beelden voor gemaakt zijn. Een laatste mogelijke uitbreiding is om te proberen een onderscheid te maken tussen de perfect gesneden witloofkroppen en de witloofkroppen waarvan er te veel van de krop zelf afgesneden is. De data om het model voor deze uitbreiding te trainen zijn voor deze toepassing al verzameld. De opmerking hierbij is dat het verschil tussen te veel en perfect gesneden witloof visueel moeilijker te onderscheiden is dan perfect gesneden en te weinig gesneden witloof.

Hoe de voorstelling van de snijlijn omgezet moet worden naar poolcoördinaten is reeds bepaald. Nu rest nog een model te ontwikkelen dat de poolcoördinaten kan voorspellen. Om het model op te bouwen kan geprobeerd worden om een model dat gebruikt wordt om een object te detecteren aan te passen. Het model zal niet langer vier uitgangen hebben die een insluitende rechthoek voorstellen, maar twee uitgangen die de hoek θ en de lengte R van een lijn voorstellen.

6

Bibliografie

- [1] C. Li, Q. Cao, and F. Guo, "A method for color classification of fruits based on machine vision," *WSEAS TRANSACTIONS on SYSTEMS*, vol. 8, pp. 312–321, 2 2009. [Online]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=704b3a4a608e930a8cf13b5c033dd30186dd2e2b>
- [2] A. M. Rossetto and W. Zhou, "Deep learning for categorization of lung cancer ct images," *Proceedings - 2017 IEEE 2nd International Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE 2017*, pp. 272–273, 8 2017. [Online]. Available: <https://ieeexplore-ieee-org.kuleuven.e-bronnen.be/abstract/document/8010653>
- [3] K. K. Singh, S. Kumar, M. Antonakakis, K. Moirgiorgou, A. Deep, K. L. Kashyap, M. K. Bajpai, and M. Zervakis, "Deep learning capabilities for the categorization of microcalcification," *International Journal of Environmental Research and Public Health 2022, Vol. 19, Page 2159*, vol. 19, p. 2159, 2 2022. [Online]. Available: <https://www.mdpi.com/1660-4601/19/4/2159/htmlhttps://www.mdpi.com/1660-4601/19/4/2159>
- [4] J. G. Kim, S. C. Jang, I. S. Kang, D. J. Lee, J. W. Lee, and H. S. Park, "A study on object recognition using deep learning for optimizing categorization of radioactive waste," *Progress in Nuclear Energy*, vol. 130, p. 103528, 12 2020. [Online]. Available: <https://www-sciencedirect-com.kuleuven.e-bronnen.be/science/article/pii/S0149197020302766>
- [5] P. K. Jayapal, E. Park, M. A. Fageerzada, Y. S. Kim, H. Kim, I. Baek, M. S. Kim, D. Sandanam, and B. K. Cho, "Analysis of rgb plant images to identify root rot disease in korean ginseng plants using deep learning," *Applied Sciences 2022, Vol. 12, Page 2489*, vol. 12, p. 2489, 2 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/5/2489/htmlhttps://www.mdpi.com/2076-3417/12/5/2489>
- [6] H. Li, C. Li, G. Li, and L. Chen, "A real-time table grape detection method based on improved yolov4-tiny network in complex background," *Biosystems Engineering*, vol. 212, pp. 347–359, 12 2021. [Online]. Available: <https://www-sciencedirect-com.kuleuven.e-bronnen.be/science/article/pii/S1537511021002786>

- [7] G. Liu, J. C. Nouaze, P. L. T. Mbouembe, and J. H. Kim, "Yolo-tomato: A robust algorithm for tomato detection based on yolov3," *Sensors* 2020, Vol. 20, Page 2145, vol. 20, p. 2145, 4 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/7/2145/htmlhttps://www.mdpi.com/1424-8220/20/7/2145>
- [8] D. Zhou, Y. Fan, G. Deng, F. He, and M. Wang, "A new design of sugarcane seed cutting systems based on machine vision," *Computers and Electronics in Agriculture*, vol. 175, p. 105611, 8 2020. [Online]. Available: <https://www-sciencedirect-com.kuleuven.e-bronnen.be/science/article/pii/S0168169920313715>
- [9] H. Azarmdel, S. S. Mohtasebi, A. Jafari, and A. R. Muñoz, "Developing an orientation and cutting point determination algorithm for a trout fish processing system using machine vision," *Computers and Electronics in Agriculture*, vol. 162, pp. 613–629, 7 2019. [Online]. Available: <https://www-sciencedirect-com.kuleuven.e-bronnen.be/science/article/pii/S0168169919300766>
- [10] K. . Yang, Z. . Yu, F. . Gu, Y. . Zhang, S. . Wang, B. . Peng, Z. Hu, K. Yang, Z. Yu, F. Gu, Y. Zhang, S. Wang, B. Peng, and Z. Hu, "Experimental study of garlic root cutting based on deep learning application in food primary processing," *Foods* 2022, Vol. 11, Page 3268, vol. 11, p. 3268, 10 2022. [Online]. Available: <https://www.mdpi.com/2304-8158/11/20/3268/htmlhttps://www.mdpi.com/2304-8158/11/20/3268>
- [11] K. Zhao, Q. Han, C. B. Zhang, J. Xu, and M. M. Cheng, "Deep hough transform for semantic line detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 4793–4806, 9 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9422200/?casa_token=basUn5kEXu0AAAAA:mQbLcamV4DJkK024cad686B2lqNVZ3YBdxYO6dhvFDa40nQyVd_45CD25MFvnDvmHsZLMiqyQOG5
- [12] Z. Zhang, Z. Li, N. Bi, J. Zheng, J. Wang, K. Huang, W. Luo, Y. Xu, and S. Gao, "Ppgnet: Learning point-pair graph for line segment detection," 2019.
- [13] I. Corporation, "Intel realsense tm product family d400 series datasheet intel realsense tm vision processor d4, intel realsense tm vision," 4 2022. [Online]. Available: <https://www.intelrealsense.com/wp-content/uploads/2022/05/Intel-RealSense-D400-Series-Datasheet-April-2022.pdf>
- [14] PyTorch, "resnet18 — torchvision main documentation," 2017. [Online]. Available: <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html>
- [15] Google, "Interpreting loss curves — machine learning — google for developers," 7 2022. [Online]. Available: <https://developers.google.com/machine-learning/testing-debugging/metrics/interpretic>

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
CAMPUS BRUGGE
Spoorwegstraat 12
8200 BRUGGE, België
tel. + 32 50 66 48 00
iiw.brugge@kuleuven.be
www.iw.kuleuven.be

