BB8 herkansing onderzoeksraport

Optimalisatie van Dataset en Evaluatie van Modellen voor Objectdetectie

Matthew Jim

1812418

Hogeschool Utrecht

24-8-2023

Inhoudsopgave

1. Inleiding

In dit onderzoeksrapport word de optimalisatie van een vegetatie objectdetectiedataset en de evaluatie van verschillende modellen voor objectdetectie onderzoekt. Door gebruik te maken van het Roboflow AutoML-model "Roboflow 3.0 Object Detection (Fast)" en verschillende YOLOv8-configuraties (YOLOv8-N, YOLOv8-S, YOLOv8-M en YOLOv8-L), beogen we de verbetering van de dataset en de identificatie van het meest geschikte model. Dit rapport presenteert de achtergrond, methodologie, resultaten en advies van het onderzoek.

2. Theoretisch kader

2.1. Objectdetectie

Objectdetectie is een cruciale taak binnen computer vision die tot doel heeft om objecten in afbeeldingen of video's te lokaliseren en te classificeren. Het proces van objectdetectie omvat het identificeren van de locatie van objecten door middel van begrenzende rechthoeken, ook wel bounding boxes genoemd, en het toewijzen van een bijbehorende classificatie aan elk object.

Een veelgebruikte methode voor objectdetectie is het gebruik van convolutionele neurale netwerken (CNN's). Deze netwerken zijn in staat om automatisch kenmerken uit afbeeldingen te extraheren en deze te gebruiken om objecten te detecteren en classificeren. YOLO (You Only Look Once) is een populaire CNN-gebaseerde methode voor objectdetectie die bekend staat om zijn snelheid en efficiëntie (*What is object detection?*, z.d.).

2.2. You Only Look Once (YOLO)

You Only Look Once (YOLO) is een reeks van krachtige deep learning-modellen voor objectdetectie die real-time prestaties mogelijk maken. Binnen het YOLO-framework zijn verschillende versies ontwikkeld, waaronder YOLOv8-N, YOLOv8-S, YOLOv8-M en YOLOv8-L, elk met verschillende configuraties en complexiteitsniveaus, waardoor ze verschillende toepassingsmogelijkheden bieden.

2.2.1. YOLOv8-N, YOLOv8-S, YOLOv8-M en YOLOv8-L

Elk van deze YOLO-versies, aangeduid met 'N', 'S' 'M' en 'L', vertegenwoordigt een schaalvariant binnen het YOLO-framework. Deze varianten variëren in termen van modelgrootte, diepte en capaciteit.

- **YOLOv8-N (Nano):** Dit is de kleinste en lichtste configuratie binnen het YOLOv8-framework. Het is geoptimaliseerd voor zeer snelle real-time toepassingen met minimale resources, waardoor het ideaal is voor scenario's waarbij snelheid van cruciaal belang is.

- **YOLOv8-S (Small):** Deze variant is geoptimaliseerd voor snelheid en efficiëntie, met minder parameters en een lichtere rekenkundige belasting. Het is ideaal voor real-time toepassingen waarbij hoge snelheid essentieel is, zoals autonome voertuigen en realtime surveillance.

- **YOLOv8-M (Medium):** YOLOv8-M bevindt zich tussen YOLOv8-S en YOLOv8-L in termen van modelcomplexiteit en nauwkeurigheid. Met iets meer parameters dan YOLOv8-S biedt het een hogere nauwkeurigheid, terwijl het nog steeds redelijk efficiënt is voor gebruik in real-time scenario's.
- **YOLOv8-L (Large):** Deze variant heeft een grotere en diepere architectuur dan de andere YOLOv8-varianten, waardoor het in staat is om complexere objecten en scenario's aan te pakken. Dit gaat echter gepaard met een hogere rekenkundige complexiteit en kan meer resources vereisen.

De keuze voor een specifieke YOLO-configuratie hangt af van de vereisten van het project, zoals de balans tussen grootte van de dataset, nauwkeurigheid, snelheid en de beschikbare rekenkracht (Solawetz, 2023).

2.3. Data-augmentatie

Data-augmentatie is een techniek om de hoeveelheid en variatie van trainingsdata te vergroten. Dit wordt bereikt door de bestaande dataset te verrijken met gewijzigde versies van dezelfde afbeeldingen. Augmentatietechnieken omvatten horizontale en verticale flips, rotatie, schaalveranderingen en meer. Door data-augmentatie wordt het model blootgesteld aan verschillende weergaven van objecten, wat de robuustheid en generalisatie van het model kan verbeteren (Shorten & Khoshgoftaar, 2019).

2.3.1 Online & offline augmentatie

In dit project zijn zowel offline als online data-augmentatietechnieken gebruikt om de trainingsdata te verrijken. Bij offline data-augmentatie worden de transformaties op de trainingsdata toegepast vóór het feitelijke leerproces. In dit onderzoek is de trainingsdataset geaugmenteerd met behulp van Roboflow en vervolgens geëxporteerd. Daarentegen wordt online data-augmentatie toegepast tijdens het trainingsproces zelf. Bij deze benadering worden willekeurige transformaties toegepast op elke batch trainingsgegevens voordat deze naar het model wordt gestuurd. In dit onderzoek is YOLOv8 ingezet, dat automatisch online data-augmentatie implementeert. Het gebruik van YOLOv8 als online data-augmentatiemechanisme stelt het model in staat om tijdens elke iteratie te leren van verschillende perspectieven van dezelfde objecten, wat bijdraagt aan de robuustheid en algemene prestaties van het model (*The full Guide to Data augmentation in Computer Vision*, z.d.).

3. Data

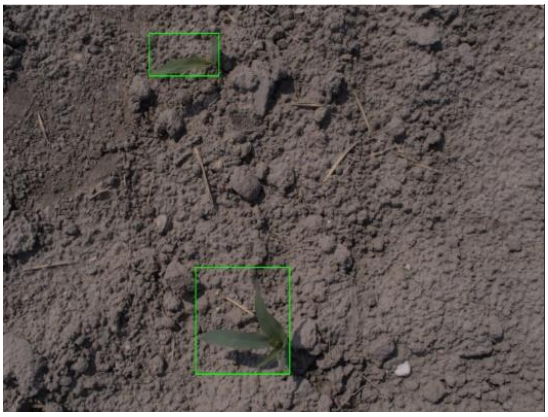
3.1. Achtergrond

In het kader van deze opdracht moest ik een keuze maken tussen het gebruik van online data of de data van Asset Insight. Ik besloot om een online dataset te gebruiken, omdat ik niet over de volledige dataset van Asset Insight kon beschikken. Mijn eigen dataset dat ik nog over had van Asset Insight bestond uit ongeveer 600 afbeeldingen, wat onvoldoende was voor een representatieve dataset. Daarom begon ik op zoek te gaan naar de "weed25" dataset, die veelbelovend leek. Helaas bleek het onmogelijk om deze dataset te downloaden, aangezien dit een Chinees telefoonnummer vereiste. Uiteindelijk stuitte ik op de "WE3DS" dataset (Kitzler et al., 2023). Hoewel oorspronkelijk ontworpen voor semantische segmentatie in landbouw, heb ik deze dataset grondig aangepast om deze te optimaliseren voor objectdetectie. Het doel van deze datasetoptimalisatie was om de efficiëntie en effectiviteit van de modellen te verbeteren.

3.2. Dataset omzetten

In eerste instantie heb ik een proces ontwikkeld om de dataset, die oorspronkelijk was ontworpen voor semantische segmentatie, om te zetten naar een formaat dat geschikt is voor objectdetectie. Dit betekende dat ik de bestaande "segmentation masks" (Figuur 1) transformeerde naar afbeeldingen met bijbehorende "bounding boxes" (Figuur 2). Hoewel dit proces veel veelbelovende bounding boxes opleverde, kwam ik ook enkele uitdagingen tegen. Een deel van de gegenereerde bounding boxes bleek van lage kwaliteit te zijn. Sommige van deze bounding boxes waren bijvoorbeeld erg klein en leken niet relevant voor objectdetectie.

Figuur 1: Originele 'segmentation masks



Figuur 2: Afbeelding met 'bounding box'

Om de kwaliteit van de dataset te verbeteren, heb ik een grondige evaluatie uitgevoerd. Hierbij heb ik alle afbeeldingen met slechte bounding boxes geïdentificeerd en verwijderd uit de dataset. Van de oorspronkelijke 2568 afbeeldingen bleven er na dit proces 2071 afbeeldingen over. Deze aanpassingen waren essentieel om ervoor te zorgen dat de dataset alleen hoogwaardige voorbeelden bevat die geschikt zijn voor objectdetectie.

Na deze aanpassingen heb ik een tweede controle uitgevoerd op de gehele dataset om ervoor te zorgen dat er geen verdere problemen waren met de bounding boxes en dat de dataset klaar was voor het trainingsproces.

3.3. Dataset optimalisaties

Voor het optimaliseren van de dataset, met als doel het versnellen en efficiënter maken van het trainingsproces, zijn verschillende strategieën en technieken toegepast. Hieronder worden de gebruikte methoden en begrippen toegelicht.

3.3.1. Data Augmentatie

Om de variabiliteit en generalisatie van de dataset te vergroten, is data-augmentatie uitgevoerd. Verschillende transformaties zijn toegepast op de afbeeldingen, waaronder horizontale en verticale flips, rotaties, bijsnijden, shearing, wijzigingen in kleurwaarden zoals hue, saturation en brightness, aanpassingen van de belichting, vervaging, toevoeging van ruis, cutout en het toepassen van de mosaic-techniek. Deze technieken zijn ingezet om diverse weergaven van de objecten te creëren en overfitting te verminderen (Shorten & Khoshgoftaar, 2019).

De technieken die zijn toegepast via Roboflow zijn: flip, rotate, crop, shear, hue, saturation, brightness, exposure, blur, noise, cutout en mosaic.

Flip: Horizontal, Vertical
90° Rotate: Clockwise, Counter-Clockwise, Upside Down
Crop: 0% Minimum Zoom, 20% Maximum Zoom
Rotation: Between -15° and +15°
Shear: ±15° Horizontal, ±15° Vertical
Hue: Between -25° and +25°
Saturation: Between -25% and +25%
Brightness: Between -25% and +25%
Exposure: Between -25% and +25%
Blur: Up to 2.5px
Noise: Up to 5% of pixels
Cutout: 3 boxes with 10% size each
Mosaic: Applied

Figuur 3: Augmentaties via Roboflow

3.3.1.1. Flip, 90° Rotate, Crop, Rotation en Shear

Deze technieken zijn ingezet om de variabiliteit van de dataset te vergroten. De keuzes voor deze technieken zijn gemotiveerd door het feit dat de vegetatie in een echte omgeving in verschillende hoeken kunnen worden waargenomen. Deze techniek heeft geholpen om het model te leren objecten te herkennen, ongeacht hun oriëntatie, waardoor het vermogen van het model om accuraat te voorspellen werd verbeterd.

Ondersteund door het werk van Wang et al. (2020) werd aangetoond dat deze augmentatie technieken leiden tot een betere generalisatie van modellen voor objectdetectie. Door objecten te presenteren in verschillende hoeken tijdens het trainingsproces, kan het model robuustere en nauwkeurigere voorspellingen doen, zelfs wanneer objecten in uiteenlopende posities verschijnen in de praktijk.

Figuur 4: Originele afbeelding voor augmentatie

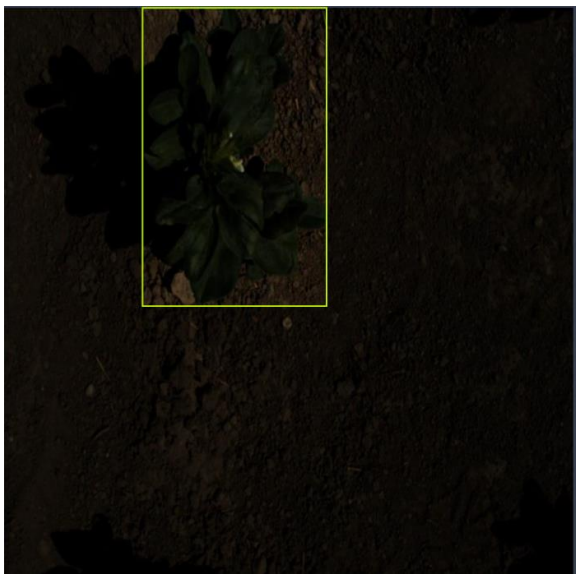
Figuur 5: Afbeelding na deze augmentaties

3.3.1.2. Hue, Saturation, Brightness en Exposure

De beslissing om hue, saturation,brightness en exposure-augmentatie toe te passen was gebaseerd op de behoefte om het model te trainen met variaties in belichting en kleurintensiteit. Door de hue (kleurtoon), saturation (verzadiging), brightness (helderheid) en exposure (gamma) van de afbeeldingen willekeurig aan te passen, werd het model blootgesteld aan een breder scala aan lichtomstandigheden en kleurvariaties. Deze technieken hebben geholpen om het model robuuster te maken tegen veranderende lichtomstandigheden en omgevingen met verschillende kleurkenmerken. Tegelijkertijd is er een bewuste keuze gemaakt om geen Grayscale-augmentatie te implementeren. Het vermijden van volledige desaturatie en het behouden van de kleurinformatie is erg belangrijk, aangezien kleur een essentieel aspect is bij vegetatiedetectie.

Tijdens de kleur augmentatie werden verzadigings aanpassingen gedaan, waarbij de verzadiging op -25 en +25 werd ingesteld. Hierdoor wordt voorkomen dat de afbeeldingen volledig desatureerd worden en in een grayscale-modus terechtkomen. Deze subtielere aanpassingen zijn nuttig, omdat vegetatie vaak in verschillende kleurtinten verschijnt, en het model moet worden getraind om deze variatie op te nemen (Nelson, 2020).

Figuur 6: Originele afbeelding voor augmentatie



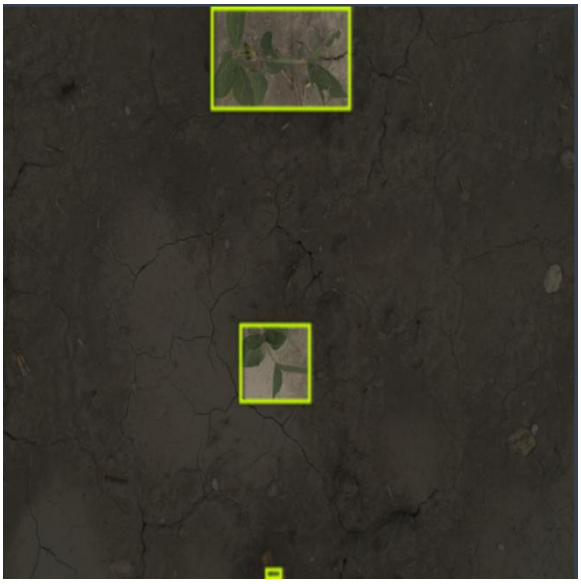
Figuur 7: Afbeedling na deze augmentaties

3.3.1.3. Blur, Noise en Cutout

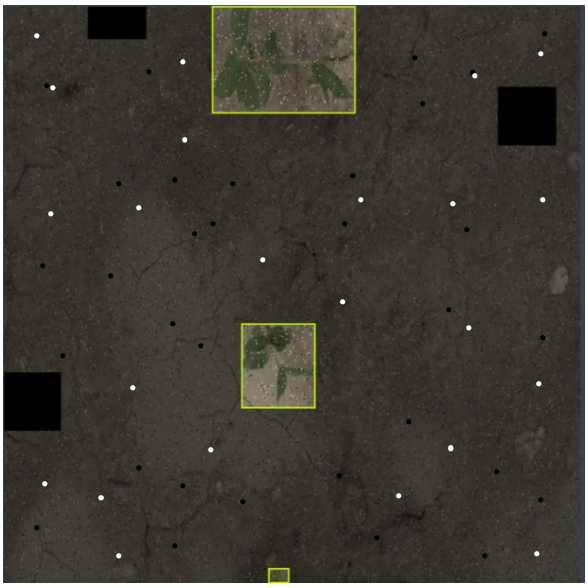
Blur, noise en cutout-augmentatie zijn toegepast om het model robuuster te maken tegen wazige of ruisige omgevingen, zoals vaak voorkomt in praktische scenario's. Het toevoegen van lichte vervaging en ruis in de trainingsbeelden helpt het model om te gaan met realistische omstandigheden en voorkomt overmatige afhankelijkheid van perfecte, heldere beelden tijdens het trainingsproces. De Cutout-augmentatie heeft betrekking op het willekeurig verwijderen van rechthoekige secties uit de afbeeldingen. Hiermee wordt het model gedwongen om te leren van verschillende delen van de afbeeldingen en niet te afhankelijk te zijn van specifieke kenmerken op een bepaalde locatie. Dit stimuleert het model om objecten op verschillende plaatsen en schalen te detecteren, wat bijdraagt aan een betere algemene generalisatie.

Onderzoekers van de Arizona State University hebben de impact van vervaging en ruis onderzocht. In hun onderzoek getiteld “Understanding How Image Quality Affects Deep Neural Networks," hebben Samuel Dodge en Lina Karam de effecten van verschillende beeldvervormingen op convolutionele neurale netwerkarchitecturen onderzocht. Hun onderzoek toonde aan dat vervaging en ruis de meest significante negatieve invloed hadden op eenvoudige classificatietaken. Dit onderzoek benadrukt het belang van het overwegen van beeldkwaliteit in machine vision systemen en belicht de gevoeligheid van diepe neurale netwerken voor verschillende soorten kwaliteitsvervormingen, met name vervaging en ruis (Dodge & Karam, 2016).

Hoewel vervaging en ruis zijn toegepast als onderdeel van de augmentatie, is ervoor gekozen om ze subtiel aan te passen om de prestaties van het model niet nadelig te beïnvloeden. Vervaging is bijvoorbeeld ingesteld op een lichte vervaging van 2.5 pixels, en ruis is begrensd tot een maximum van 5%. Dit zorgt ervoor dat de kwaliteitsvervormingen binnen aanvaardbare grenzen blijven, terwijl het model wordt blootgesteld aan realistische omstandigheden en variaties in beeldkwaliteit. Door deze benadering zijn we erin geslaagd om het model beter bestand te maken tegen kwaliteitsvervormingen zonder dat dit ten koste gaat van de prestaties in praktische toepassingen.



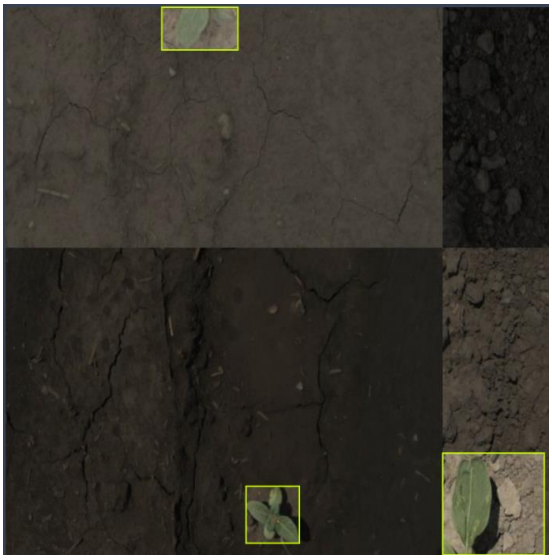
Figuur 8: Originele afbeeldinig voor augmentatie



Figuur 9: Afbeelding na deze augmentaties

3.3.1.4. Mosiac

De keuze voor mosaic-augmentatie was gebaseerd op het verlangen om het model te trainen met complexere scènes waarin meerdere objecten dicht bij elkaar kunnen verschijnen. Mosaic-augmentatie combineert vier afbeeldingen tot één enkele afbeelding, wat resulteert in een samengestelde scène met verschillende objecten in verschillende contexten.



Figuur 10: Afbeelding van mosaic augmentatie

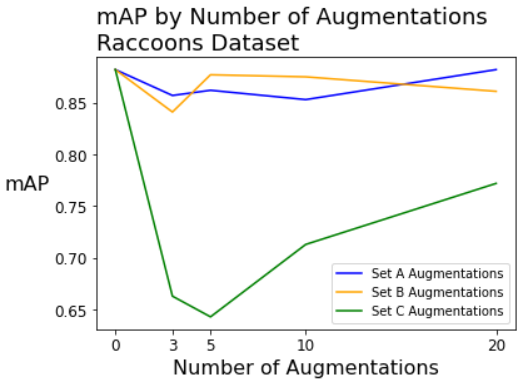
3.3.1.5. Bounding box augmentaties

Na uitvoerig onderzoek naar verschillende augmentatietechnieken, werd ik geïnspireerd door een veelzijdige benadering van data augmentatie die door Brems (2022) in zijn onderzoek werd gepresenteerd. Brems categoriseerde de augmentatietechnieken in drie verschillende sets, elk gericht op het verbeteren van verschillende aspecten van het model:

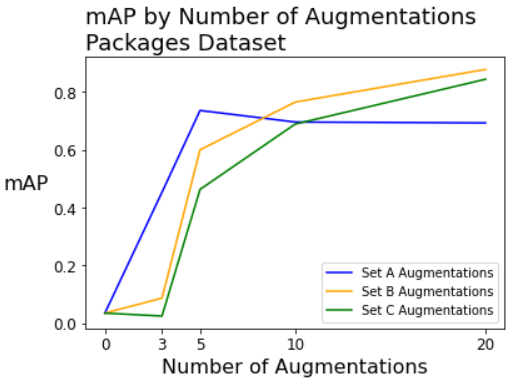
- **Set A:** Random rotate, random crop, and random noise: Deze set omvatte technieken zoals het willekeurig roteren van afbeeldingen tot 10 graden linksom of rechtsom, het willekeurig bijsnijden van afbeeldingen tot 20% om in te zoomen op details, en het toevoegen van willekeurige ruis aan 10% van de pixels in de afbeelding. Deze technieken dienden om variatie te introduceren in de oriëntatie, schaal en textuur van de objecten in de afbeeldingen.

- **Set B:** Random brightness, random blur, and random cutout, plus all “Set A” augmentations: Hier werden technieken toegevoegd zoals het willekeurig aanpassen van de helderheid met maximaal 30%, het toevoegen van willekeurige Gaussische vervaging tot 1.75 pixels om bewegingsonscherpte te simuleren, en het willekeurig verwijderen van 5 rechthoekige secties (cutout) tot 15% van de afbeelding om occlusie na te bootsen. Deze augmentaties waren bedoeld om het model beter te maken in het omgaan met verschillende belichtings- en onscherpe omstandigheden.

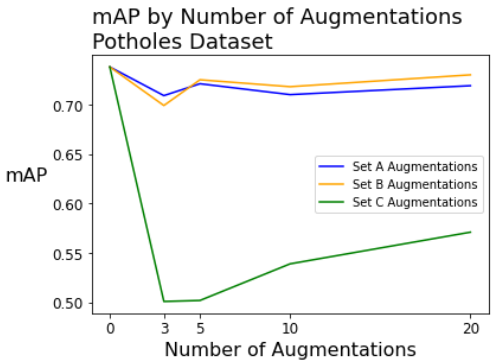
- **Set C:** Bounding box-level rotation, bounding box-level crop, and bounding box-level brightness, plus all “Set A” and “Set B” augmentations: In deze geavanceerdere set werden bounding box-level augmentaties geïntroduceerd, waarbij de focus lag op het aanpassen van specifieke gebieden binnen de begrenzingskaders. Dit omvatte het roteren en bijsnijden van de inhoud binnen de begrenzingskaders en het aanpassen van de helderheid van het begrenzingskadergebied. Deze augmentaties waren met name relevant voor situaties waarin nauwkeurigheid binnen de begrenzingskaders van groot belang was. (Brems, 2022)



Figuur 11: De verschillende augmentaties op de Raccoons Dataset (Brems, 2022)



Figuur 12: De verschillende augmentaties op de Packages Dataset (Brems, 2022)



Figuur 13: De verschillende augmentaties op de Potholes Dataset (Brems, 2022)

Onder de verschillende datasets die vertoont werden heeft de potholes dataset de meeste overeenkomst met de mijne. In het onderzoek werd duidelijk dat ondanks de veelbelovende aard van bounding box-level augmentaties, "Set C" niet leidde tot de gewenste verbeteringen in de prestaties van het model (Brems, 2022). Deze bevindingen benadrukken het feit dat niet alle augmentatietechnieken even effectief zijn in elke context en dat hun impact op de prestaties van het model afhankelijk is van de van de dataset, het model zelf en de specifieke taken die worden uitgevoerd.

Op basis van deze bevindingen en het feit dat "Set C" niet de gewenste resultaten opleverde, heb ik er bewust voor gekozen om geen gebruik te maken van bounding box-level augmentaties in mijn eigen onderzoek. In plaats daarvan heb ik me gericht op andere augmentatietechnieken zoals “Set B”, die meer geschikt leken voor mijn specifieke context van objectdetectie in vegetatiebeelden.

3.3.1. Labelcorrectie en Verbetering

Labelcorrectie en verbetering zijn processen die zijn uitgevoerd om ervoor te zorgen dat de annotaties van bounding boxes nauwkeurig de ware locaties en omvang van objecten in de afbeeldingen weergeven. Tijdens dit proces zijn slecht gepositioneerde of onnauwkeurige bounding boxes geïdentificeerd en gecorrigeerd of verwijderd om de dataset van hoogwaardige labels te voorzien. Dit zorgt ervoor dat het model kan vertrouwen op consistente en betrouwbare labels tijdens het trainingsproces.

3.3.2. Hyperparameter Tuning

in het kader van dit onderzoek zijn hyperparameter tuning aanpassingen gedaan, maar alleen op het gebied van augmentatie hyperparameters. Dit omvatte het experimenteren met verschillende instellingen voor data-augmentatietechnieken, zoals rotatie, schaalvergroting, helderheid, en andere parameters die de eigenschappen van de trainingsdata manipuleren. Andere hyperparameters, zoals learning rate, filters en batchgrootte, zijn niet gewijzigd en zijn gehouden op hun standaardwaarden. Hiermee werd gefocust op het verkennen van het potentieel van de gebruikte data-augmentatietechnieken om de prestaties van de modellen te verbeteren zonder in te grijpen in andere aspecten van het model.

3.3.3. GPU Training

Het trainen van de modellen is aanzienlijk versneld dankzij het gebruik van grafische verwerkingseenheden (GPUs). Deze krachtige hardware maakt parallelle verwerking van neurale netwerken mogelijk, waardoor de trainingscycli aanzienlijk sneller verlopen. Dit resulteerde in een aanzienlijke verkorting van de trainingsduur, waarbij bijvoorbeeld de tijd die nodig was per epoch voor een YOLOv8-M model daalde van 30 minuten naar slechts 30 seconden. De inzet van GPU's heeft dus aantoonbaar bijgedragen aan een efficiëntere en snellere ontwikkeling van de modellen.

3.3.4. Early Stop

Om onnodige rekentijd te vermijden en overfitting tegen te gaan, is het "early stop" mechanisme gebruikt. Dit betekent dat het trainingsproces automatisch wordt beëindigd wanneer de prestaties niet meer verbeteren gedurende een vooraf bepaald aantal epochs. In het geval van YOLOv8 is dit mechanisme geconfigureerd met het argument "patience=50", wat betekent dat het trainingsproces wordt gestopt als er gedurende 50 opeenvolgende epochs geen verbetering wordt waargenomen (Song et al., 2019).

Door deze strategieën te combineren, is getracht de dataset te optimaliseren voor effectievere en efficiëntere modeltraining, wat uiteindelijk moet leiden tot verbeterde objectdetectiemodellen.

3.4. Roboflow augmentatie

Nadat ik de verschillende data-augmentatietechnieken had geselecteerd, heb ik de oorspronkelijke dataset geüpload naar Roboflow. Hier heb ik gebruik gemaakt van de augmentatiefuncties van Roboflow om de dataset verder te verrijken. De initiële dataset bestond uit 2071 afbeeldingen voordat de augmentatie werd toegepast. Door de augmentatieprocessen uit te voeren binnen het Roboflow-platform, werd de dataset uitgebreid tot 4971 afbeeldingen.

Met deze geaugmenteerde dataset heb ik vervolgens een trainingsmodel ontwikkeld en getraind met behulp van het "Roboflow Train Model Type: Roboflow 3.0 Object Detection (Fast)" model. Daarnaast heb ik de geaugmenteerde dataset ook geëxporteerd naar het YOLOv8-formaat. Deze uitgebreide dataset stelde me in staat om later verschillende YOLOv8-modellen te trainen en te vergelijken, met behulp van de diversiteit aan gegevens die zorgvuldig waren voorbereid en geaugmenteerd in Roboflow.

3.5. YOLOv8 augmentatie

Naast de geaugmenteerde dataset die ik heb verkregen via Roboflow, heb ik ook de originele dataset laten ondergaan aan een augmentatieproces met behulp van YOLOv8. Deze augmentatie vond plaats tijdens het trainingsproces, waarbij online augmentatie werd toegepast om de variatie en diversiteit van de trainingsgegevens te vergroten. Hierdoor kon het model beter worden getraind om om te gaan met verschillende scenario's en omgevingscondities.

Voor de augmentatie binnen YOLOv8 heb ik verschillende parameters ingesteld om variatie en complexiteit aan de originele dataset toe te voegen. Sommige van deze parameters staan op 0, wat betekent dat die specifieke augmentaties niet worden toegepast.

In tabel 1 zie je de default augmentatieparameters in YOLOv8 met de betekenis:

Tabel 1: Default YOLOv8 augmentatie waardes

Soort augmentatie	Waarde	Betekenis
hsv_h	0.015	Kleurtoonverschuiving (fraction)
hsv_s	0.7	Verzadigingsaanpassing (fraction)
hsv_v	0.4	Helderheidsaanpassing (fraction)
degrees	0.0	Rotatie in graden (+/- deg)
translate	0.1	Vertaling in fractie van de afbeeldingsgrootte +/- (fraction)
scale	0.5	Schaalverandering van de afbeelding.: 0.0: Schuiftransformatie (+/- gain)
shear	0.0	Schuiftransformatie (+/- deg)
perspective	0.0	Perspectivische vervorming (+/- fraction)
flipud	0.0	Verticaal spiegelen (probability)
fliplr	0.5	Horizontaal spiegelen (probability)
mosaic	1.0	Gebruik van mosaic-augmentatie (probability)
mixup	0.0	Mixup-augmentatie (probability)
copy_paste	0.0	Segment copy paste (probability)

Later heb ik tests uitgevoerd om te onderzoeken hoe de augmentatie-instellingen van Roboflow zich gedragen in combinatie met de augmentatieparameters van YOLOv8. Binnen YOLOv8 zijn er specifieke augmentatie-opties beschikbaar die kunnen worden geconfigureerd om ze na te bootsen zoals die van Roboflow. Hieronder worden enkele van de gebruikte augmentatiewaarden van Roboflow vermeld, samen met de overeenkomstige augmentatieparameters in YOLOv8 en hun betekenis:

Tabel 2: Roboflow augmentatie waardes in YOLOv8

Soort augmentatie	Waarde	Betekenis
hsv_h	0.25	Kleurtoonverschuiving (fraction)
hsv_s	0.25	Verzadigingsaanpassing (fraction)
hsv_v	0.25	Helderheidsaanpassing (fraction)
degrees	0.90	Rotatie in graden (+/- deg)
translate	0.0	Vertaling in fractie van de afbeeldingsgrootte +/- (fraction)
scale	0.0	Schaalverandering van de afbeelding.: 0.0: Schuiftransformatie (+/- gain)
shear	0.15	Schuiftransformatie (+/- deg)
perspective	0.0	Perspectivische vervorming (+/- fraction)
flipud	0.5	Verticaal spiegelen (probability)
fliplr	0.5	Horizontaal spiegelen (probability)
mosaic	1.0	Gebruik van mosaic-augmentatie (probability)
mixup	0.0	Mixup-augmentatie (probability)
copy_paste	0.0	Segment copy paste (probability)

4. Modellen

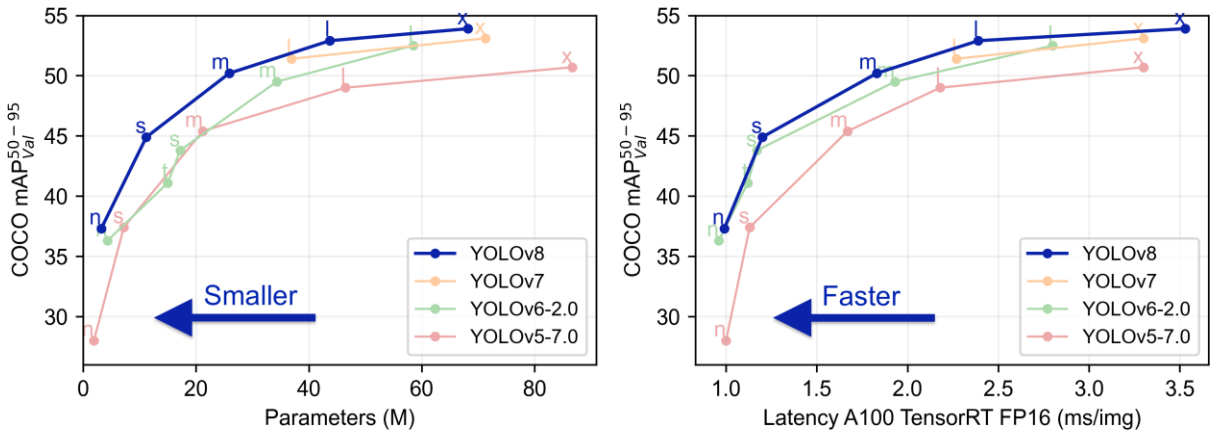
4.3. Model keuze

De keuze voor YOLOv8 is ingegeven door de bekendheid en bewezen effectiviteit van dit model in objectdetectietaken. YOLOv8 staat bekend om zijn snelheid en efficiëntie, waardoor het bij uitstek geschikt is voor real-time toepassingen zoals objectdetectie in beelden. Met zijn meerdere schaalvarianten en configuraties biedt YOLOv8 de flexibiliteit om te voldoen aan verschillende vereisten qua nauwkeurigheid en rekentijd, waardoor het een aantrekkelijke keuze is voor dit onderzoek. In Figuur 14 (Ultralytics, z.d.) zie je een plot van de verschillende YOLO versies. Hier zie je dat YOLOv8 kleiner en veel sneller is dan de andere versies van YOLO.

Daarnaast zie je in figuur 15 verschilende object detection modellen die getrained zijn op verschillende vegetatie datasets. Hier zie je dat YOLOv8l consistent het beste presteert op verschillende vegetatiescenario's. Deze bevindingen tonen aan dat YOLOv8 superieure prestaties levert in vergelijking met andere modellen zoals YOLOv5m, YOLOv6l, YOLOv7 en YOLOv8l, evenals EfficientDet, wanneer toegepast op objectdetectie in vegetatie-omgevingen. (Sportelli et al., 2023)

Wat betreft het Roboflow-model is de keuze gebaseerd op de unieke augmentatiefunctionaliteiten die Roboflow biedt. Roboflow stelt gebruikers in staat om uitgebreide augmentaties toe te passen op de dataset, wat resulteert in een verrijkte dataset met variaties die zich in real-world scenario's kunnen voordoen. De gebruiksvriendelijke interface en de mogelijkheid om online data-augmentatie toe te

passen, maakten Roboflow een waardevolle aanvulling op dit onderzoek, waardoor ik een breder scala aan geaugmenteerde datasets kon verkennen en vergelijken.



Figuur 14: Vergelijking van YOLO modellen (Ultralytics, z.d.)

Model	Dataset	P	R	mAP_0.5	mAP_0.5:0.95	Inference (ms) [a]
EfficientDet	'Weeds' public	0.9133	0.9273	0.9426	0.7023	44.3
	Home Lawn	0.5982	0.5149	0.5195	0.4155	52.0
	Baseball Field	0.6138	0.7069	0.6614	0.4136	54.2
	Manila grass	0.6047	0.6954	0.5691	0.4369	50.1
YOLOv5m	'Weeds' public	0.9433	0.9663	0.9772	0.7828	16.2
	Home Lawn	0.6331	0.5272	0.5399	0.4263	19.2
	Baseball Field	0.6856	0.8126	0.7135	0.4716	24.1
	Manila grass	0.6441	0.5433	0.6412	0.5007	18.7
YOLOv6l	'Weeds' public	0.9442	0.9494	0.9747	0.7612	22.8
	Home Lawn	0.7836	0.6446	0.7057	0.5022	32.5
	Baseball Field	0.6098	0.6491	0.5379	0.4108	47.9
	Manila grass	0.5865	0.7571	0.7014	0.5248	26.6
YOLOv7	'Weeds' public	0.9265	0.9627	0.9745	0.7685	16.1
	Home Lawn	0.7118	0.6454	0.7108	0.5209	29.1
	Baseball Field	0.6223	0.7579	0.6379	0.4009	35.6
	Manila grass	0.6549	0.7571	0.6461	0.4614	27.5
YOLOv8l	'Weeds' public	0.9476	0.9610	0.9795	0.8123	34.0
	Home Lawn	0.6567	0.6422	0.6564	0.4721	37.4
	Baseball Field	0.6672	0.6474	0.6459	0.4312	19.7
	Manila grass	0.7635	0.6519	0.7589	0.5296	36.6

[a] Inference time refers to the average time needed for the model to detect weeds on a single digital image.

Figuur 15: Prestatie en inferentie tijd van de beste YOLO-modelschalen voor de vier bestudeerde YOLO versies (YOLOv5m, YOLOv6l, YOLOv7 en YOLOv8l) en EfficientDet (Sportelli et al., 2023)

4.4. Modeltraining:

Het modeltrainingsproces omvatte het trainen van zowel YOLOv8- als Roboflow-modellen voor vegetatiedetectie. Voor YOLOv8 werden verschillende configuraties van het model gebruikt, waaronder de nano, small, medium en large varianten. Elk van deze configuraties vertegenwoordigt verschillende schalen en capaciteiten, waardoor ik een reeks modellen kon verkennen en vergelijken op basis van hun prestaties en snelheid. Voor Roboflow heb ik een Roboflow 3.0 Object Detection (Fast) model getrained.

Aan de YOLOv8-kant heb ik modellen getraind met behulp van de originele dataset zonder augmentatie, de geaugmenteerde dataset van Roboflow en de geaugmenteerde dataset met de standaard

augmentatieparameters van YOLOv8. Bovendien heb ik getest hoe de augmentatieparameters van Roboflow zouden presteren in combinatie met de YOLOv8-augmentatie-instellingen.

Bij Roboflow heb ik zowel een model getraind met de geaugmenteerde dataset als een model met de originele dataset zonder enige vorm van augmentatie. Dit bood de mogelijkheid om de impact van Roboflow-augmentatie op de prestaties van het model te beoordelen.

Alle YOLO-modellen zijn getraind tot 120 epochs met een batchgrootte van 16, wat heeft bijgedragen aan een gedegen trainingsproces en betere prestaties. Het gebruik van verschillende combinaties van trainingsdata en augmentaties stelde me in staat om te begrijpen hoe elk model reageerde op verschillende contexten en omstandigheden.

Hoewel er pre-trained modellen beschikbaar waren die specifiek gericht waren op vegetatiedetectie, heb ik vastgesteld dat deze modellen niet geschikt waren voor mijn specifieke doeleinden. Bij nader onderzoek bleek dat de beschikbare vegetatiedatasets aanzienlijk klein waren en dat de bijbehorende labels van lage kwaliteit waren. Om deze reden heb ik besloten om geen gebruik te maken van pre-trained modellen en in plaats daarvan mijn eigen modellen te trainen.

Voor de data split heb ik ervoor gekozen om een 70/20/10 verdeling te gebruiken voor de train-, validatie- en testsets. Dit is gedaan in Roboflow. Deze verdeling is een veelgebruikte aanpak in machine learning om voldoende data te reserveren voor het trainen en valideren van modellen, terwijl er ook genoeg data overblijft om de uiteindelijke prestaties op een onafhankelijke testset te evalueren (Solawetz, 2022).

4.5. Validatie

Om de prestaties van de modellen te evalueren, zijn verschillende evaluatiemetrieken gebruikt, waaronder precision, recall, mAP en F1-score. Deze metrieken bieden inzicht in de nauwkeurigheid, volledigheid en het evenwicht tussen precision en recall van de modellen. Bovendien is de mean Average Precision (mAP) berekend, een maatstaf voor de precision van objectdetectie in meerdere klassen.

De resultaten van de validatie worden weergegeven in Tabel 3, waarbij de verschillende metrieken voor elk model zijn opgenomen. Uit de resultaten blijkt dat het YOLOv8-M-model met de Roboflow waardes augmentaties de hoogste precision heeft gehaald met een precision van 94.3%. Dit betekent dat bij de voorspellingen van dit model een hoog percentage van de gedetecteerde vegetatie daadwerkelijk correct is. Het model met de hoogste recall is YOLOv8-S-model met de default augmentatie waardes met een recall van 88.8%, wat betekent dat dit model een hoog percentage van de werkelijke vegetatie correct heeft gedetecteerd.

Daarnaast zijn ook de F1-score en mAP berekend om het evenwicht tussen precision en recall te beoordelen en de consistentie van de detectie over verschillende IoU-thresholds te meten. Uit de resultaten blijkt dat YOLOv8-S-model met de default augmentatie waardes de hoogste F1-score heeft behaald, wat wijst op een goede balans tussen precision en recall. Ook toont YOLOv8-S-model met de Roboflow waardes augmentaties de hoogste mAP-waarde, wat aangeeft dat dit model consistent hoge precision in de detectie behoudt over verschillende IoU-thresholds.

Op basis van deze evaluatiemetrics lijkt het YOLOv8-S-model met de default augmentatie waardes over het algemeen het best presterende model te zijn. Dit wordt ondersteund door het feit dat, hoewel het

YOLOv8-M-model met Roboflow augmentatie waardes de hoogste precision had, het verschil slechts 0.1% was. Bovendien was de precision van het YOLOv8-S-model met standaard augmentatieparameters slechts 0.2% lager dan die van het YOLOv8-M-model met standaard augmentatie waardes. Hierbij komt dat het trainen van een YOLOv8-S-model ook aanzienlijk sneller verliep.

Tabel 3: Gevalideerde modellen met de metrics

Dataset	Model	Precision	Recall	F1-score	mAP50
Zonder Augmentatie	Roboflow	0.880	0.821	0.849	0.881
	YOLOv8-M	0.888	0.883	0.859	0.892
YOLOv8 default Augmentatie	YOLOv8-N	0.936	0.881	0.907	0.927
	YOLOv8-S	0.942	**0.888**	**0.941**	0.932
	YOLOv8-M	0.941	0.876	0.907	0.933
	YOLOv8-L	0.935	0.880	0.906	0.928
YOLOv8 Roboflow waardes augmentatie	YOLOv8-N	0.937	0.887	0.910	0.932
	YOLOv8-S	0.937	0.905	0.921	**0.934**
	YOLOv8-M	**0.943**	0.882	0.912	0.931
Roboflow Augmentatie	YOLOv8-M	0.837	0.772	0.775	0.782
	Roboflow	0.939	0.836	0.884	0.881

Een interessante observatie uit de resultaten is dat het YOLOv8-S-model als het beste presterende model naar voren komt, zij het met een minimale marge. Dit verschil in prestatie tussen de modellen is echter subtiel en geeft aan dat de verschillende modellen zeer competitieve resultaten laten zien.

Een andere opvallende bevinding is dat het gebruik van grotere YOLOv8-modellen, zoals YOLOv8-M en YOLOv8-L, niet noodzakelijkerwijs heeft geleid tot significant hogere precision-waarden. Dit kan deels worden verklaard door de omvang van de dataset, met name de beperkte omvang van de validatieset, die 20% van de totale gegevens vertegenwoordigt. De relatief kleine validatieset kan enige onzekerheid introduceren in de evaluatie en resultaten, waardoor kleine variaties tussen modellen minder duidelijk worden.Deze observatie is in lijn met een vergelijkbare bevinding in een andere bron. Volgens Glue (2023) wordt vermeld dat YOLOv8n (nano) sneller, kleiner en nauwkeuriger is dan YOLOv8 Small of YOLOv8 Medium bij gebruik van kleinere datasets. Dit benadrukt verder de mogelijke invloed van de datasetgrootte op de prestaties van verschillende modelconfiguraties.

Een andere intrigerende bevinding is dat de overstap van de default augmentatie van YOLOv8 naar de YOLOv8-roboflow augmentatie waardes een merkbare verbetering in recall opleverde. Deze observatie suggereert dat de specifieke augmentatie-instellingen een positieve invloed kunnen hebben op het vermogen van het model om objecten te detecteren en correct te classificeren.

Samenvattend is het duidelijk dat alle geteste modellen indrukwekkende prestaties hebben geleverd in het detecteren van vegetatie. Het YOLOv8-S-model onderscheidde zich als het beste presterende model binnen de grenzen van de validatieset. Het is echter belangrijk op te merken dat de verschillen tussen de modellen over het algemeen subtiel zijn en dat de datasetomvang, met name de validatieset, invloed kan hebben op de interpretatie van de resultaten. Deze bevindingen benadrukken de complexiteit van

modelselectie en prestatiebeoordeling in de context van objectdetectie, waarin meerdere factoren een rol kunnen spelen bij het bepalen van het meest geschikte model voor een specifieke taak.

4.6. Test:

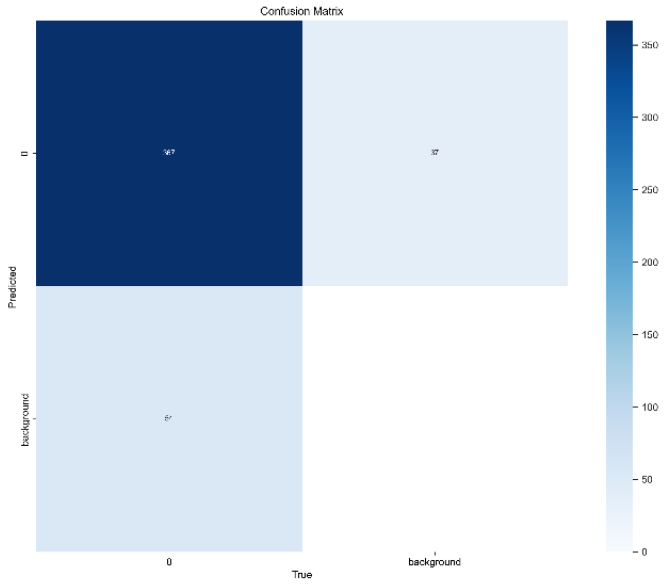
Na de validatie van alle modellen, werd de prestatie van het beste model verder geëvalueerd op de testdataset. De testdataset is een onafhankelijke dataset die het model tijdens de training en validatie niet heeft gezien, waardoor het mogelijk is om de algemene prestaties van het model te meten op ongeziene gegevens.

In tabel 4 zie je de metrics die uit de testset komen. Wat opvalt is dat bij het testen van het model op de testset, de precision is gedaald van 0.942 naar 0.871 in vergelijking met de resultaten van de validatieset. Tegelijkertijd is de recall gestegen van 0.888 naar 0.908 en is de F1-score gedaald van 0.941 naar 0.889. Deze verschillen tussen de testset- en validatieset-resultaten kunnen wijzen op een zekere mate van variabiliteit in de prestaties van het model wanneer het wordt geconfronteerd met ongeziene gegevens. In Figuur 16 wordt de confusion matrix van de testset weergegeven, waarin verschillende evaluatiemetrics zijn afgeleid.

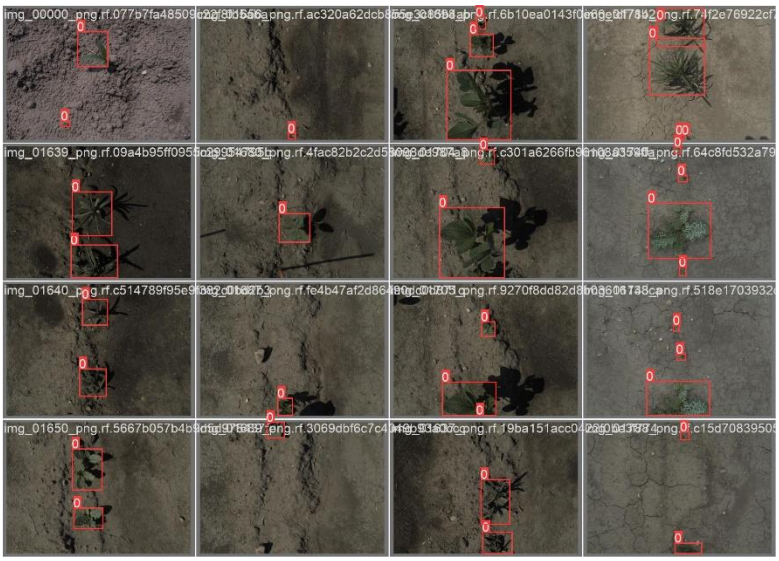
Tabel 4: Resultaten YOLOv8-S op de testset

Dataset	Model	Precision	Recall	F1-score
Testset	YOLOv8-S	0.871	0.908	0.889

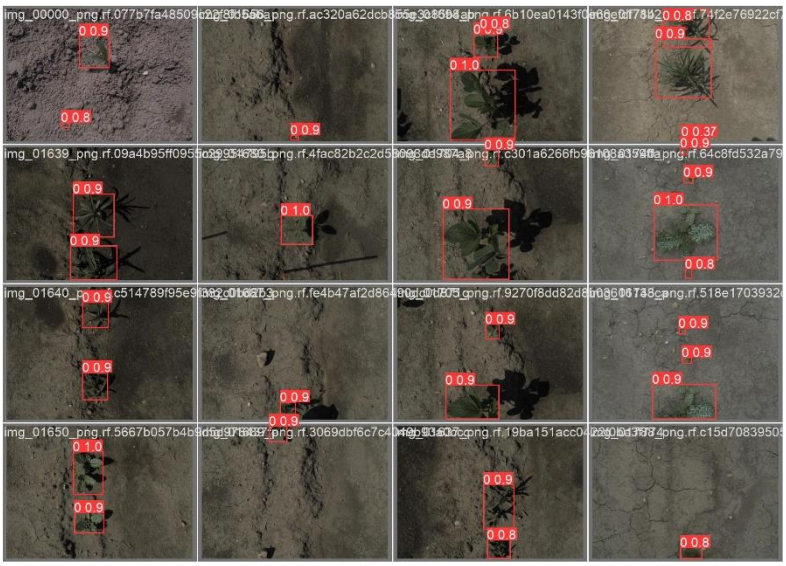
Op Figuur 17 en 18 worden de originele labels vergeleken met de voorspelde labels. Een opmerkelijk aspect is dat het model tijdens het trainingsproces geen kennis heeft genomen van de originele labels. Dit biedt de mogelijkheid om de werkelijke prestaties van het model op ongeziene gegevens te beoordelen. Opmerkelijk is dat de voorspelde labels allemaal een hoge mate van vertrouwen (confidence) vertonen, variërend van 0.8 tot 1.0.



Figuur 16: Confusion matrix van de testset



Figuur 17: Originele labels



Figuur 18: Predicted labels

5. Conclusie en advies

In dit onderzoek heb ik geëxperimenteerd met verschillende modellen en augmentatietechnieken om het meest effectieve model te identificeren voor het detecteren van vegetatie in afbeeldingen. Mijn doel was om te achterhalen welke combinatie van modelarchitectuur en augmentatietechnieken resulteert in optimale prestaties. Uit de resultaten blijkt dat het YOLOv8-S-model met de default augmentatie waardes over het algemeen het beste presterende model is.

Voor toekomstige studies en voortzetting van dit werk, zijn er verschillende aspecten die verdere aandacht verdienen en waarop verbeteringen kunnen worden aangebracht:

Meer data toevoegen: Het opvallende verschil in prestaties tussen de YOLOv8-S en de grotere modellen, zoals YOLOv8-M en YOLOv8-L, suggereert dat de dataset mogelijk niet voldoende complexiteit heeft om de voordelen van grotere modellen te benutten. Het vergroten van de dataset en diversifiëren van de afbeeldingen kan leiden tot betere prestaties van grotere modellen.

Aanpassen van Train-Val-Test Verhoudingen: In dit onderzoek is de gebruikelijke 70/20/10 train-val-test verhouding gehanteerd. Toekomstige studies kunnen de effecten onderzoeken van het aanpassen van deze verhoudingen en de impact op de prestaties van de modellen analyseren.

Toenemende Aantal Epochs: Hoewel de modellen in dit onderzoek tot 120 epochs zijn getraind, kan het interessant zijn om de impact te onderzoeken van het verhogen van het aantal epochs om te bepalen of er verdere prestatieverbeteringen kunnen worden bereikt naarmate het trainingsproces vordert.

Meer hyperparameter Tuning: Het optimaliseren van de hyperparameters, zoals de leersnelheid, batchgrootte en andere parameters, kan een aanzienlijke invloed hebben op de prestaties van de modellen. Toekomstige studies kunnen geavanceerde hyperparameteroptimalisatietechnieken gebruiken om de beste combinaties van parameters te identificeren.

Deze suggesties bieden een basis voor toekomstig onderzoek om de nauwkeurigheid, robuustheid en toepasbaarheid van vegetatiedetectiemodellen verder te verbeteren. Het continue onderzoek op dit

gebied draagt bij aan de ontwikkeling van effectieve en praktisch inzetbare oplossingen voor vegetatiedetectie.

6. Bronnenlijst:

- What is object detection? (z.d.). MATLAB & Simulink. https://www.mathworks.com/discovery/object-detection.html
- Solawetz, J. (2023). What is YOLOV8? The ultimate guide. Roboflow Blog. https://blog.roboflow.com/whats-new-in-yolov8/
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for Deep learning. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0
- The full Guide to Data augmentation in Computer Vision. (z.d.). https://encord.com/blog/data-augmentation-guide/
- Kitzler, F., Barta, N., Neugschwandtner, R. W., Gronauer, A., & Motsch, V. (2023). WE3DS: An RGB-D image Dataset for semantic segmentation in agriculture. Sensors, 23(5), 2713. https://doi.org/10.3390/s23052713
- Song, H., Kim, M., Park, D., & Lee, J. (2019). Prestopping: How does early stopping help generalization against label noise? arXiv (Cornell University). https://arxiv.org/pdf/1911.08059.pdf
- Wang, K., Fang, B., Qian, J., Yang, S., Zhou, X., & Zhou, J. (2020). Perspective Transformation data augmentation for object detection. IEEE Access, 8, 4935–4943. https://doi.org/10.1109/access.2019.2962572
- Nelson, J. (2020). Introducing grayscale and Hue/Saturation augmentations. Roboflow Blog. https://blog.roboflow.com/introducing-grayscale-and-hue-augmentations/
- Dodge, S., & Karam, L. J. (2016). Understanding how image quality affects deep neural networks. arXiv (Cornell University). http://export.arxiv.org/pdf/1604.04004
- Brems, M. (2022). The Power of Image Augmentation: an experiment. Roboflow Blog. https://blog.roboflow.com/the-power-of-image-augmentation-experiment/
- Ultralytics. (z.d.). GitHub - Ultralytics/Ultralytics: NEW - YOLOV8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite. GitHub. https://github.com/ultralytics/ultralytics
- https://blog.roboflow.com/train-test-split-with-roboflow/
- Sportelli, M., Apolo-Apolo, O., Fontanelli, M., Frasconi, C., Raffaelli, M., Peruzzi, A., & Ruiz, M. P. (2023). Evaluation of YOLO object detectors for weed detection in different turfgrass scenarios. Applied sciences, 13(14), 8502. https://doi.org/10.3390/app13148502
- Glue, R. (2023, 3 mei). YOLOV8, EfficientDET, Faster R-CNN or YOLOV5 for remote sensing. Medium. https://medium.com/@rustemgal/yolov8-efficientdet-faster-r-cnn-or-yolov5-for-remote-sensing-12487c40ef68