

IDENTIFICATIE EN OPSPORING VAN INVASIEVE AZIATISCHE HOORNAARS DOOR MIDDEL VAN BEELDHERKENNING

Het VELU-PI prototype: een open-source automatisch monitoringssysteem

Wolf Missotten

Cedric Billiet

Wolf De Witte

Kobe Lambert

Promotor: Wim Van Criekinge

Tutor: Wim Van Criekinge

Bachelorproef voorgelegd voor het behalen van de graad Bachelor of Science in de Bio-ingenieurswetenschappen

Academiejaar: 2023-2024

De auteurs en promotor(s) geven de toelating deze scriptie voor consultatie beschikbaar te stellen en delen ervan te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting uitdrukkelijk de bron te vermelden bij het aanhalen van resultaten uit deze scriptie.

Gent, 6 mei 2024

De auteurs,

Wolf Missotten

Cedric Billiet

Wolf De Witte

Kobe Lambert

De promotor en tutor,

Wim Van Criekinge

Wim Van Criekinge

INHOUDSOPGAVE

Inhoudsopgave	3
1 Inleiding.....	5
2 Literatuurstudie.....	6
2.1 Introductie in Europa.....	6
2.2 Waarom moet de Aziatische hoornaar bestreden worden?	6
2.3 Levenscyclus	8
2.4 Gedrag.....	9
2.5 Herkenning	10
2.6 Bestrijdingsmethoden	11
2.7 Deep learning als nieuwe bestrijdingsmethode	15
3 Materialen en methodes	18
3.1 Beeldherkenning met YOLOv8	18
3.1.1 Verzameling van beelden	18
3.1.2 Annotatie.....	18
3.1.3 Preprocesing	19
3.1.4 Augmentatie.....	19
3.1.5 Training van YOLOv8 op de dataset.....	20
3.2 Gebruikte hardware, basisconfiguraties & opstelling.....	22
3.2.1 Sensoren	22
3.2.2 PiJuice	24
3.2.3 Opstelling.....	26
3.3 Aanmaken van een lokale MySQL database	27
3.4 Afbeeldingen en video's verzenden naar Dropbox.....	29
3.5 Finale Python script.....	30
4 Resultaten.....	31
4.1 Prestatiemetrieken.....	31
4.2 Resultaat van training YOLOv8.....	32
4.3 Werking finale script	36
5 Discussie	36
5.1 Verdere evaluatie van VELU-PI.....	36
5.2 Verbeteringen voor VELU-PI	37
6 Conclusie.....	38

7	<i>Duurzaamheidsreflectie</i>	39
7.1	Kadering	39
7.2	Reflectievraag 1: Relatie tot duurzame ontwikkeling	39
7.3	Reflectievraag 2: Relatie tot het maatschappelijke debat	40
8	<i>Referenties</i>	42
9	<i>Appendices</i>	49
	Appendix A: Training van YOLOv8 in de Colab-notebook	49
	Appendix B: De PiJuice automatisch afsluiten.....	50
	Appendix C: Resetten interne klok PiJuice verhelpen	51
	Appendix D: Finale Python script	52
	Appendix E: Verbinding op afstand met MariaDB	56
	Appendix F: Python script automatisch laten lopen vanaf opstart Raspberry Pi.....	59

1 INLEIDING

De opkomst van de Aziatische hoornaar (*Vespa velutina*) in Europa doet tal van imkers en natuurexperten in de haren wrijven, aangezien deze invasieve exoot zowel een gevaar voor bijenpopulaties als de algemene biodiversiteit vormt. Ondanks de ontwikkeling van diverse methoden om de opmars een halt toe te roepen, heeft geen enkele aanpak deze exoot de kop kunnen indrukken. Een belangrijke uitdaging hierbij is dat de huidige bestrijding van de Aziatische hoornaar voornamelijk berust op meldingen die gemaakt worden door imkers en burgers, waarbij het insect handmatig wordt geïdentificeerd, wat een arbeidsintensief proces is. Deze bachelorproef introduceert daarom het VELU-PI prototype, een open-source automatisch monitoringssysteem dat de krachten van deep learning en Citizen Science tracht te bundelen tot een efficiënt en toegankelijk systeem voor de bestrijding van de Aziatische hoornaar.

VELU-PI, een acroniem voor **V**espa **v**elutina **E**limination with deep **L**earning **U**sing Raspberry **P**I, is gebaseerd op een Raspberry Pi 4 die gebruik maakt van een beeldherkenningsmodel getraind met behulp van het YOLOv8-algoritme. De configuratie omvat een camera, een bewegingssensor en een sensor voor de omgevingscondities. Telkens wanneer de bewegingssensor beweging detecteert, wordt een foto vastgelegd en geanalyseerd door het beeldherkenningsmodel. Indien het model een Aziatische hoornaar detecteert, begint de camera met filmen. Zowel de afbeelding als de video die werden gemaakt, worden opgeslagen in een databank ter bewijs en controle.

Eerst en vooral volgt een literatuurstudie dat zich richt op verschillende aspecten, waaronder algemene informatie over de Aziatische hoornaar, de reeds bestaande bestrijdingsmethoden en het gebruik van deep learning als bestrijdingsstrategie. Vervolgens wordt een beschrijving gegeven van de gebruikte materialen en methoden waarbij duidelijkheid, reproduceerbaarheid en implementeerbaarheid centraal staat. Het doel is namelijk dat deze bachelorproef kan dienen als handleiding voor imkers en burgers om het VELU-PI prototype zelf te kunnen bouwen. Daarom worden stappen die vereist zijn voor het nabouwen van het VELU-PI systeem in detail beschreven, wat minder het geval is bij het proces van het trainen van het beeldherkenningsmodel. Het beeldherkenningsmodel hoeven geïnteresseerden namelijk niet zelf te trainen gezien het beschikbaar is, net als alle andere benodigde code, op de GitHub-repository: <https://github.com/VELU-PI/VELU-PI>. In de resultaten wordt de prestatie van het beeldherkenningsmodel geëvalueerd. De discussie richt zich op het identificeren van tekortkomingen van het VELU-PI systeem en het voorstellen van mogelijke verbeteringen.

2 LITERATUURSTUDIE

2.1 Introductie in Europa

Villemant et al. (2006) vermoeden dat de introductie van de Aziatische hoornaar zich voordeed in Europa in 2004, met het eerst geïdentificeerde nest in de nabijheid van Agen (Monceau et al., 2013b). De koninginnen maakten in dat jaar hoogstwaarschijnlijk hun intrede via transport van Chinees aardewerk naar het zuidwesten van Frankrijk. De Aziatische hoornaar heeft sindsdien een sterke opmars gekend in Europa. Van 2005 tot 2012 vergrootte de omvang van het gebied waarin ze voorkwamen met een gemiddelde snelheid van 78 kilometer per jaar, mede geholpen door hun dispersie via menselijk transport (Robinet et al., 2016).

In 2011 werd het eerste individu in België gespot (Schoonvaere et al., 2020). Het eerste nest werd in 2016 gemeld en sindsdien kwamen in Vlaanderen 400 meldingen van nesten binnen in de periode van 2017 tot en met 2021 (Meer Nesten van de Aziatische Hoornaar Ondanks Koude Lente en Natte Zomer | Natuurpunt, z.d.). De helft van die meldingen kwam uit 2021. In 2022 werden 1400 nesten gemeld en in 2023 steeg dit cijfer zelfs nog tot 5700 (Van Ransbeeck, 2024), al is dit onder voorbehoud, aangezien de data nog niet volledig geverifieerd is.

2.2 Waarom moet de Aziatische hoornaar bestreden worden?

De Aziatische hoornaar jaagt onder andere op bijen en kan volledige bijenkolonies uitroeien, waardoor deze soort berucht is onder imkers. De recentste cijfers van het Vlaams Bijeninstituut (Eerste meting van het effect van de Aziatische hoornaar op bijenvolken in Vlaanderen | Vlaams Bijeninstituut, 2023) zijn alarmerend en tonen aan dat er dringend actie ondernomen moet worden. Uit een bevraging in 2023 bij 300 Vlaamse imkers blijkt dat 85% dit jaar last heeft gehad van de Aziatische hoornaar. Bij 32% van de imkers is er minstens een van die invasieve exoten binnengetrepen in de bijenkorven. Als we kijken naar alle gecontroleerde bijenvolken (2404 in totaal) dan was 18% nog voor de winter leeg teruggevonden, waarbij de Aziatische hoornaar voor 70% de boosdoener was.

De inheemse Europese honingbij *Apis mellifera* is samen met de Europese Vespidae geëvolueerd, bijgevolg zijn hun afweermechanismen op deze soorten aangepast. Omdat de manier van jagen van *Vespa velutina* verschilt van die van *Vespa crabro* (Europese hoornaar) zijn de evolutionaire aanpassingen niet even efficiënt. Tan et al. (2007) beschrijven het verschil tussen de Europese honingbij *Apis mellifera* en de Aziatische soort *Apis cerana*, waarbij deze laatste van nature een verspreidingsgebied heeft dat overlapt met het verspreidingsgebied van *Vespa velutina*. Specifiek werd ingezoomd op het succes van de “bee-hawking” jachttechniek van de *Vespa velutina*. Bij deze jachttechniek vliegen de hoornaars voor de toegang van de bijenkorf, waar ze uitgaande of terugkerende werksters in de lucht vangen. Bij het vergelijken werd duidelijk dat *A. cerana* gewoon is om met deze jachttechniek om te gaan. Wanneer hoornaars voor de kast vliegen, halveert de tijd die *A. cerana* nodig heeft om de kast te betreden. *A. mellifera* daarentegen vertraagt, en probeert de hoornaars eerder te ontwijken. Dit maakt hen een makkelijkere prooi voor de snelle *Vespa velutina*. Daarnaast rekruteert *A. cerana* driemaal zoveel wachters als *A. mellifera*. De succesratio van *Vespa velutina* bij *A. mellifera* was driemaal hoger dan bij *A. cerana*. Daarbij was de kans dat de Aziatische hoornaar naar de *A. mellifera* kolonie ging acht keer groter wanneer de hoornaar zich op gelijke afstand van beide nesten bevonden.

Naast het directe verlies van werksters schaadt de Aziatische hoornaar ook indirect de bijenkast. Kolonies van de honingbijen die frequent worden aangevallen door hoornaars lijden aan ‘foraging paralysis’, wat een stressreactie is op de intense jachttechniek van de Aziatische hoornaar. De aanwezigheid van de predator zorgt ervoor dat de bestuivers tijdelijk stoppen met foerageren, waardoor de kolonie dus minder voedsel verzamelt (Requier et al, 2018.; Van Ransbeeck et al., 2024). Wanneer deze foraging paralysis te lang aanhoudt, legt de bijenkast onvoldoende voorraden aan. Dit doet de kans dat de bijenkast de winter overleeft significant dalen.

De Aziatische hoornaar jaagt naast honingbijen (*Apis mellifera*) ook op andere insecten. Volgens de cijfers van Vespa-Watch bestaat het dieet van de Aziatische hoornaar in stedelijke gebieden voor 66% uit honingbijen, maar in bos- en landbouwgebieden is dit slechts 35% (Wat is het probleem | Vespa-Watch, z.d.). Het andere deel van hun dieet bestaat voornamelijk uit andere wilde bestuivers, die ook belangrijk zijn voor de lokale flora. Wat voor impact de Aziatische hoornaar heeft op de overlevingskansen van andere bestuivers is voorlopig nog onzeker en moet nog verder onderzocht worden.

Wel is al bewezen dat de aanwezigheid van de predator zorgt voor minder bestuiving. De Aziatische hoornaar creëert tijdens het foerageren voor die bestuivers een ‘landscape of fear’, waarbij de inheemse bestuivers afgeschrikt worden om de locatie nog te bezoeken en dus te bestuiven (Van Ransbeeck et al., 2024). Vooral sociale bestuivers zoals bijen en hommels mijden opvallend de jachtgebieden. Ze hebben dan ook de mogelijkheid om met elkaar te communiceren, door bijvoorbeeld het gevaar met feromonen te markeren. Het onderzoek van Rojas-Nossa en Calviño-Cancela (2020) bevestigt het fenomeen van de ‘landscape of fear’. Volgens hun bevindingen leidt de aanwezigheid van de *Vespa velutina* bij bloemenperken tot lagere bezoekersaantallen voor elke bloem. Bovendien daalt de tijd die de bestuivers doorbrengen op de bloem (Rojas-Nossa en Calviño-Cancela, 2020). Hierdoor is de invasieve wespensoort dus niet enkel problematisch voor de fauna, maar ook voor de flora. De Aziatische hoornaar staat daarom ook op de lijst van de Europese Unie voor “zorgwekkende invasieve soorten” (Unielijst invasieve exoten | NVWA, 2024).

De invloed van de Aziatische hoornaar op lokale bestuivers vertaalt zich bovendien ook in economische schade. Door de reductie aan bestuivers en het aanvreten van vruchten door de hoornaar, lijdt de landbouwsector aan enorme verliezen. Fedele et al. (2019) schatten de jaarlijkse economische verliezen van fruitteelt in Spanje in op een waarde van 7.200.000 euro als gevolg van de reductie van bestuivingsdiensten door de hoornaar. Een EU gefinancierd project toont aan dat de bijdrage van bestuiving voor de Europese landbouw neerkomt op een richtwaarde van 15 miljard euro door hun invloed op de hoeveelheid, kwaliteit en zekerheid van de voedselvoorziening (Bescherming van wilde bestuivers in de EU | Europese rekenkamer, 2020). Daarenboven is de economische kost voor de imker ook aanzienlijk, want zo kost een nieuw bijenvolk al ongeveer 120 euro en dat is zonder het verlies van de honingooft en bestuivingsdiensten gerekend (Schoonvaere et al., 2020).

Het aantal gemelde nesten neemt elk jaar toe en het is zeer moeilijk om te voorspellen wanneer dit aantal zal stagneren. Als dit jaar eenzelfde stap wordt gezet zoals gerapporteerd door Van Ransbeeck et al. (2024) tussen 2022 (1400) en 2023 (5700), dan zou het aantal nesten met een factor groter dan drie kunnen

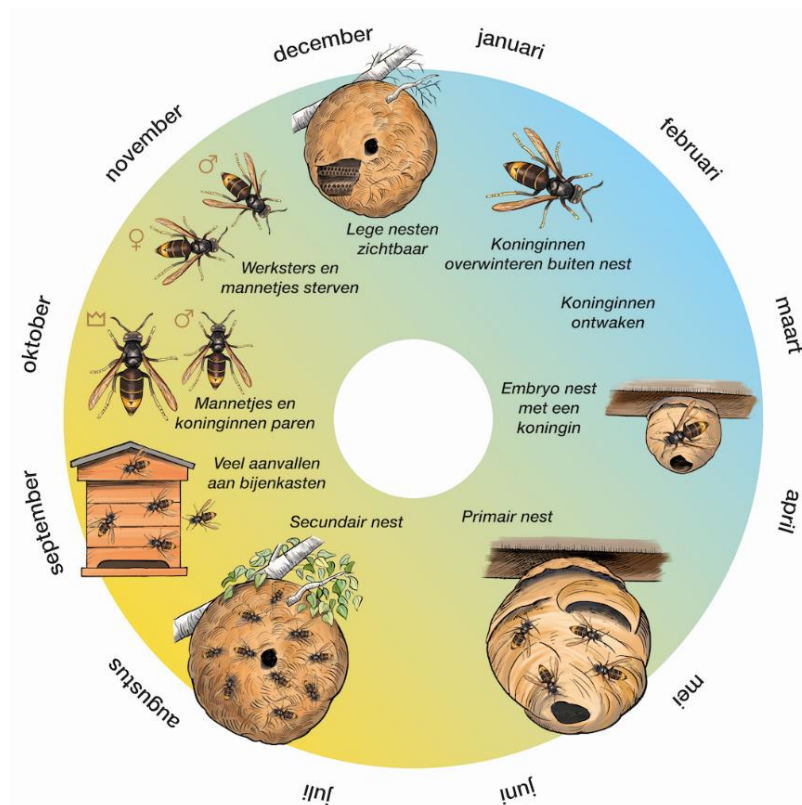
toenemen. Uiteraard is dit onzeker, en moet ook de extra bewustwording door de bevolking in rekening gebracht worden. Hoe dan ook ziet het er naar uit dat de cijfers de huidige trend blijven volgen. De omvang van het probleem lijkt voorlopig enkel toe te nemen.

2.3 Levenscyclus

Monceau et al. (2013b) beschrijven hoe de jaarlijkse levenscyclus (geïllustreerd op figuur 1) van geobserveerde kolonies in Frankrijk begint met de stichterskoninginnen. Deze komen eind maart uit overwintering, waarna ze op zoek gaan naar een plek om hun embryonest te bouwen. In het begin is de koningin zelf verantwoordelijk voor zowel de aanmaak van het nest, als het verzamelen van koolhydraten en eiwitten. Vanaf het moment dat de eerste larven uitgroeien tot werksters nemen zij deze taken over, waardoor de koningin zich kan terugtrekken in het nest. Daar richt ze zich enkel nog op reproductie. Op dat moment is het nest ook al gegroeid en is er geen sprake meer van een embryonest, maar van een primair nest.

De kolonie hoornaars zal over één jaar twee nesten maken, namelijk het primaire en het secundaire nest (Diéguez-Antón et al., 2022). Het primaire nest bevindt zich in beschutte plekken, vaak dicht bij menselijke beschaving, die de kolonie beschermen tegen de meteorologische omstandigheden. Wanneer de kolonie groeit, limiteert dit nest echter de mogelijkheden tot verdere expansie. Van eind juli tot begin augustus beginnen de werksters daarom met de constructie van een secundair nest, meestal in bomen op een hoogte van 20m. Wanneer dit secundaire nest groot genoeg is, zal de koningin haar intrek nemen. Het nest bereikt zijn maximale omvang rond midden september. Monceau et al. (2013b) beschrijven nesten met gemiddeld 6000 hoornaars, terwijl Diéguez-Antón et al. (2022) maxima tot 13000 hoornaars benoemen.

Vanaf eind september stopt het nest met het produceren van werkers en schakelt het over tot de productie van zogenaamde gynes en darren (Monceau et al., 2013b). Gynes zijn vrouwelijke hoornaars die de mogelijkheid hebben om zich te ontwikkelen tot koninginnen, terwijl darren mannelijke hoornaars zijn. Monceau et al. (2013b) bemerken dat de hoeveelheid geproduceerde darren driemaal zo groot is als de hoeveelheid gynes. Wanneer de gynes in de herfst tevoorschijn komen, kunnen ze namelijk door meerdere mannetjes bevrucht worden (polyandrie). Aangezien de koningin genetisch materiaal verkrijgt van meerdere mannetjes, zullen werksters meer genetische diversiteit vertonen onderling (Saga et al., 2020). Een meer genetisch diverse kolonie is namelijk meer bestand tegen pathogenen, omdat er meer werkers bij zijn die wel immuniteit vertonen tegenover bepaalde ziektebeelden. Na de bevruchting zoeken de bevruchte gynes een overwinteringsplaats op, terwijl de mannetjes en werksters sterven waardoor de secundaire nesten komen leeg te staan.



Figuur 1. Schematische voorstelling van de levenscyclus (Wat zie je doorheen het jaar? | Vespa-Watch, z.d).

2.4 Gedrag

Perrard et al. (2009) en Monceau et al. (2017) onderzochten de activiteit van Aziatische hoornaar kolonies doorheen de dag en doorheen het jaar. Beide onderzoeken onderscheidden foerageren en werken aan het nest als de belangrijkste activiteiten van de kolonie. Andere bezigheden omvatten het verkennen van de omgeving, het verdedigen van het nest, persoonlijke hygiëne en het zorgen voor verkoeling op warme dagen. De term foerageren refereert naar de activiteit waarbij werksters eiwitten, suikers en water verzamelen. Hierbij neemt de zoektocht naar suikers dubbel zoveel tijd in als de zoektocht naar eiwitten (Perrard et al., 2009). De Aziatische hoornaar is dagactief, zelfs bij voldoende hoge temperaturen is na zonsondergang vrijwel geen activiteit meer te zien. Daarnaast heeft ook de temperatuur een belangrijk effect. Wanneer de temperatuur onder de 10° Celsius dook, werd in de studie van Perrard et al. (2009) geen activiteit meer waargenomen. Naarmate de dag vordert, neemt de activiteit stelselmatig toe. De piek ligt in alle geobserveerde nesten rond de middag, alhoewel dit exacte moment verschilt op basis van de lokale meteorologische omstandigheden. Na deze piek begint de activiteit weer te minderen. Het is onzeker of deze piek ten gevolge van een piek in maximale temperatuur of een piek in UVB-straling van de zon komt.

De belangrijkste voedselbron voor de Aziatische hoornaar zijn koolhydraten, waarbij ongeveer 20% van de dagelijkse activiteit van werksters bestaat uit het verzamelen ervan (Monceau et al., 2013b; Perrard et al., 2009). De koolhydraten worden verzameld door gespecialiseerde werksters, die de suikers meenemen naar de kolonie om de andere hoornaars te voeden. In de embryonale fase van het nest halen de hoornaars hun koolhydraten uit boomsappen en de nectarklieren van voorjaarsbloeiërs. Van zodra de kolonie voldoende

groot is, worden de hoornaars vooral afhankelijk van orale excreties van hun larven voor hun dagelijkse suikeropname. Dit proces stopt wanneer de kolonie in zijn reproductieve fase terechtkomt. Vanaf dat moment zullen de hoornaars voornamelijk koolhydraten uit rijp fruit consumeren. Zo vond een studie van Nave et al. (2024) dat 89% van de fruitboeren aan de westkust van het Iberisch schiereiland een bepaalde vorm van schade aan hun vruchten hadden, vanwege de *Vespa velutina*. 66% van de boeren rapporteerden dat meer dan 5% van hun oogst aangetast was, 12% melden zelfs dat meer dan 75% van hun oogst schade had ondervonden door het foerageergedrag van de *Vespa velutina*. De meest geliefde vruchten waren druiven, gevolgd door appels en peren.

Naast koolhydraten verzamelen de werksters ook eiwitten, in de vorm van prooien. De zoektocht naar proteïnen neemt ongeveer 10% van de dagelijkse activiteiten in (Perrard et al., 2009). Aziatische hoornaars zijn zoals vele andere Vespidae opportunistische, generalistische jagers, die hun eetpatronen vooral laten afhangen van de meest abundante voedselbronnen in de buurt van het nest (Rome et al., 2021; Monceau et al., 2013a). De werksters verwerken de gevangen prooien tot zogenaamde “pellets” door alle overbodige lichaamsdelen af te bijten en de eiwitrijke, goed verteerbare delen samen te bundelen in een bal (Verdasca et al., 2021). Een studie door Rome et al. (2021) bestudeerde 16 nesten en de samenstelling van hun voedingspatroon. Tussen de nesten was er een vrij aanzienlijk verschil in de fractionering tussen bepaalde klassen van prooien op basis van de locatie van de nesten. Dit weerspiegelt het opportunistische karakter van hun dieet. Zo viel het bijvoorbeeld op dat het aandeel honingbijen sterk steeg in meer stedelijke gebieden. Gemiddeld gezien vond men dat het grootste aandeel van eiwitten in hun dieet bestond uit honingbijen (38,1%), gevolgd door vliegen (29,9%) en sociale wespen (19,7%). In totaal konden 159 verschillende organismen onderscheiden worden die deel uitmaakten van hun dieet.

Poidatz et al. (2018) ondervonden dat slechts 5% van hoornaars de weg naar het nest terugvonden wanneer ze op 5000 meter of meer van het nest werden vrijgelaten. De hoornaars werden hiervoor vervoerd in donkere dozen zodat ze geen visuele aanwijzingen hadden van de route. Hoe dichterbij het nest de hoornaars werden vrijgelaten, hoe groter de kans dat ze terug het nest bereikten. Van nature ligt de maximale operatieradius van de *Vespa velutina* rond de 2000 meter, maar dit hangt vooral af van de lokale abundantie van hulpbronnen (Poidatz et al., 2018). Typisch is de Aziatische hoornaar vooral actief in een radius van 500 tot 750 meter van het nest.

2.5 Herkenning

De Europese hoornaar (*Vespa crabro*) is de grootste inheemse wesp in België, maar deze groep vormt in tegenstelling tot de Aziatische hoornaar (*Vespa velutina*) geen probleem binnen het ecosysteem. Ze zijn vaak terug te vinden in boomholten en voeden hun kolonie met bijen, vliegen, wespen en andere insecten. De Europese hoornaar is de enige hoornaar die van nature voorkomt in West-Europa en is de enige soort die in competitie gaat met de Aziatische hoornaar (Schoonvaere et al., 2020). Het is dus belangrijk om de 2 hoornaars te kunnen onderscheiden. Figuur 2 vergelijkt beide organismen en geeft hun grootte weer ten opzichte van een 1 euro munt. De honingbij en Duitse wesp (*Vespula germanica*) staan voor de volledigheid ook afgebeeld en hieruit is af te leiden dat de hoornaars een maat groter zijn. Naast de grootte zijn er ook kleurkenmerken om de twee soorten hoornaars uit elkaar te halen. Ten eerste kleurt de Aziatische hoornaar donkerder met een enkele gele band ter hoogte van het achterlijf, terwijl de Europese zijn achterlijf

overwegend geel is. Het volgend opvallend kenmerk verkeert zich in de kleur van de poten met enerzijds de gele poten van de Aziatische en anderzijds de roodbruine van de Europese hoornaar.



Figuur 2. Illustratie van respectievelijk de honingbij, de Duitse wesp, de Aziatische hoornaar en de Europese hoornaar (Aziatische hoornaar (Vespa velutina) | Vlaams Bijeninstituut, 2023).

2.6 Bestrijdingsmethoden

Na de vaststelling van de destructieve impact van de Aziatische hoornaar op bijenkolonies in Europa, zijn er diverse methodes ontwikkeld om de oorzaak aan te pakken. De meest voor de hand liggende methodes zijn vallen (Turchi en Dérijard, 2018). Deze vallen kunnen zowel selectief als niet selectief zijn.

Niet selectieve vallen bestaan vaak uit een doos met een trechter op, zodat insecten er wel in, maar niet meer uit kunnen. In de val bevindt zich lokstof, meestal samengesteld uit een combinatie van een zoete substantie en een alcoholhoudende vloeistof zoals bier en/of wijn. De alcohol houdt bijen op afstand, maar heeft geen merkbaar effect op de hoornaars. Hoewel deze niet selectieve val zeer makkelijk zelf te maken is, heeft dit type val als groot nadeel dat een grote hoeveelheid andere, vaak nuttige insectensoorten ook in de val terecht komen.

Selectieve vallen daarentegen gaan bijna uitsluitend de exoot vangen. Turchi en Dérijard (2018) vermelden de belangrijkste techniek, die zich richt op de specifieke afmetingen van de Aziatische hoornaar. Een voorbeeld van zo'n zelfgemaakte selectieve val is te zien op figuur 3. In de wand van de val zit een kegel met aan de punt een ring (diameter 8,5 mm), waardoor alleen de Aziatische hoornaar of kleinere insecten binnen kunnen en de inheemse, grotere Europese hoornaar niet. In tegenstelling tot de invasieve exoot kunnen kleinere insecten, waaronder ook de honingbij, de val verlaten via rasters in de wanden (openingen met diameter 5,5 mm).



Figuur 3. Voorbeeld van een selectieve val, waarbij de selectiviteit gebaseerd is op de afmetingen van een insect (Aziatische hoornaar | Imkers Neteland, 2024).

Deze selectieve vallen kunnen als tijdelijke oplossing gebruikt worden bij bijenkasten, maar kunnen volgens Turchi en Dériard (2018) ook gebruikt worden in de lente, om koninginnen die een embryonaal nest aan het maken zijn, af te vangen. Dit wordt ook wel 'spring trapping' genoemd en is een controversieel onderwerp, omdat er volgens de auteurs ook een nadeel is aan deze methode. Bij het maken van een nest is er namelijk vaak competitie tussen koninginnen die de winter overleefd hebben. Koninginnen die er niet in slagen om een eigen embryonaal nest te maken, gaan op zoek naar embryonale nesten van andere koninginnen om deze over te nemen. Dit is een fenomeen dat gekend staat als 'queen usurpation' (Spring trapping: koninginnen Aziatische hoornaar vangen in het voorjaar | Imkers Neteland, 2023). Deze confrontatie gaat gepaard met een strijd op leven en dood, waarbij één koningin het leven laat. In sommige gevallen is de overblijvende koningin ook sterk verwond, waardoor deze ook niet overleeft. Bij het afvangen van de meest fitte koninginnen zal deze onderlinge competitie dus veel minder aanwezig zijn, wat tot de vraag leidt of de hoeveelheid koninginnen en nesten effectief verlaagt door deze vallen in de lente te plaatsen. Dit omgekeerde effect zou vooral van toepassing zijn in regio's waar de Aziatische hoornaar in grote getale aanwezig is (Spring trapping: koninginnen Aziatische hoornaar vangen in het voorjaar | Imkers Neteland, 2023). Er zijn nog veel vraagtekens over de effecten van spring trapping, waardoor verder wetenschappelijk onderzoek vereist is.

Naast de typische lokstoffen, biedt het gebruik van feromonen een extra mogelijkheid om Aziatische hoornaars te lokken. Wen et al. (2017) beschreven als eerste de twee seksferomonen van een vruchtbare Aziatische hoornaar koningin, namelijk 4-oxo-octaanzuur (4-OOA) en 4-oxo-decaanzuur (4-ODA). Darren komen heel sterk af op deze 2 stoffen, voornamelijk wanneer ze in een verhouding van 0.78 zijn. Het afvangen van darren biedt mogelijkheden om de voortplantingscycli te verstoren. Bovendien vermelden Wen et al. (2017) dat gevangen darren geen alarmferomonen verspreiden, waardoor ze andere soortgenoten niet kunnen verwittigen. Veel van deze vallen zijn nog niet commercieel beschikbaar, maar zijn veelbelovend.

Om de verdere verspreiding van de Aziatische hoornaar een halt toe te roepen, volstaat vangen en doden van individuele werksters niet. Het vroeg opsporen en vernietigen van nesten daarentegen is van cruciaal belang (Aziatische hoornaar | Imkers Neteland, 2024). Een veel gebruikte techniek voor het opsporen van nesten, maakt gebruik van wiekpotten. Zoals te zien is op figuur 4 vangen wiekpotten de hoornaars niet, maar worden ze gelokt naar een afgesloten confituurpot waar zich een mengsel van alcohol en suiker in bevindt. Een vochtige wiek trekt de vloeistof naar boven, waar de insecten op afkomen. Bij actieve controle van de wiekpotten kunnen Aziatische hoornaars gemerkt en terug vrijgelaten worden. Het merken kan aan de hand van een merkstift of met Opalith merkplaatjes, die eigenlijk bedoeld zijn om bijenkoninginnen te merken. Het merkplaatje wordt dan bevestigd met sterke lijm op het borststuk van de hoornaar, die onbeweeglijk in een merkbuisje met stamper zit (Vespawatchers-Wiekpotmethode | Imkers Neteland, 2023; Speur nesten op | Vespa-Watch, z.d.). Volgens de eerste auteur wordt een wiekpot het meest bezocht als deze tussen de 100 en de 500 meter van het nest hangt.






Figuur 4. Wiekpot met zowel Europese hoornaar (boven) als Aziatische hoornaar (onder) (Aziatische hoornaar | Imkers Neteland, 2024).

Eenmaal een hoornaar gemerkt is, kan de vliegtijd van dit individu opgemeten worden. Deze vliegtijd wordt gedefinieerd als de tijd die de hoornaar nodig heeft om van de wiekpot naar het nest te vliegen, het suikergoedje daar af te leveren (retentietijd) en terug naar dezelfde pot te vliegen (rekening houdend dat het insect geen omwegen maakt) (Van Ransbeeck et al., 2023). Schoonvaere et al. (2020) vermelden dat een vuistregel van 100 m lineaire afstand tot het nest per minuut vliegtijd, een redelijk goede schatting kan geven van de afstand van het nest. Het is een techniek die momenteel veel gebruikt wordt in Vlaanderen en Jersey (Verenigd Koninkrijk) om nesten van Aziatische hoornaar op te sporen (Schoonvaere et al., 2020; Speur nesten op | Vespa-Watch, z.d.). Door uit te gaan van de vliegtijd en deze vuistregel, kan de gemiddelde afstand tot het nest geschat worden. Door op meerdere plekken in een gebied wiekpotten te hangen en ditzelfde principe toe te passen, kan via triangulatie een goede schatting van de plaats van het nest gedaan worden. Dit houdt in dat je alle vliegrichtingen vanaf de verschillende wiekpotten op een kaart tekent (met een lengte die overeenkomt met de vliegtijd) waarbij de snijpunten van deze lijnen het zoekgebied voor het nest vormen.

Van Ransbeeck et al. (2023) onderzochten de vuistregel van 100 meter afstand per minuut vliegtijd, door in Vlaanderen 284 vliegtijden naar 35 nesten van 121 individuen te registreren (88 wiekpotten in totaal). Ze kwamen tot de conclusie dat de vuistregel standhoudt, maar dat er rekening gehouden moet worden met de nabijheid van een nest en de graad van verstedelijking. Bij nesten die dichterbij dan 382 meter van een wiekpot hangen, vliegen hoornaars trager dan verwacht en bij grotere afstanden sneller dan verwacht. Volgens de onderzoekers cirkelen hoornaars met een dichtbijgelegen nesten eerder traag omhoog, in plaats van in een rechte lijn naar het nest te vliegen, waardoor de vliegtijd toeneemt. Ook zagen Van Ransbeeck et al. (2023) variaties in de vliegtijden voor verschillende graden van urbanisatie. In landelijkere gebieden foerageerden de hoornaars namelijk sneller dan in stedelijke gebieden. Volgens de onderzoekers is dit toe te schrijven aan de neiging van hoornaars om straten te volgen in steden, wat leidt tot langere vliegroutes. Aan de hand van hun bevindingen hebben Van Ransbeeck et al. (2023) nieuwe vuistregels geformuleerd, die afhankelijk zijn van de graad van urbanisatie. In tabel 1 zijn de afstanden tot het nest weergegeven voor bepaalde vliegtijden. Voor een vliegtijd van 1 minuut is te merken dat de oorspronkelijke 100 meter vuistregel vooral goed wordt benaderd in een randstedelijk gebied.

Tabel 1. Vuistregels per graad van urbanisatie, met daaronder de geschatte afstand tot het nest voor bepaalde vliegtijden. Onzekerheid is weergegeven in de vorm van een standaardfout (Van Ransbeeck et al., 2023).

	Landelijk gebied	Randstedelijk gebied	Verstedelijkt gebied
			
Vuistregel:	$94 \times \text{vliegtijd (min)} + 107$	$94 \times \text{vliegtijd (min)} - 17$	$94 \times \text{vliegtijd (min)} - 27$
Vliegtijd (min)	Afstand tot nest \pm SE (m)	Afstand tot nest \pm SE (m)	Afstand tot nest \pm SE (m)
1	201 \pm 79	77 \pm 85	67 \pm 74
2	294 \pm 67	170 \pm 74	161 \pm 72
3	388 \pm 58	264 \pm 65	254 \pm 74
4	482 \pm 53	358 \pm 59	348 \pm 80
5	575 \pm 54	452 \pm 59	442 \pm 88
6	669 \pm 59	545 \pm 62	536 \pm 99
7	763 \pm 68	639 \pm 70	629 \pm 111
8	857 \pm 79	733 \pm 80	723 \pm 124

Het is duidelijk dat dusver besproken methodes, hoewel laagdrempelig, erg arbeidsintensief en weinig nauwkeurig zijn. Een veel accuratere methode is het gebruiken van radiotelemetrie. Hierbij krijgt de hoornaar een tag mee die radiogolven uitzendt. Deze radiogolven kunnen opgevangen worden via een antenne en bijhorende ontvanger, om zo de plaats van de hoornaar te lokaliseren (en dus hopelijk ook het nest). Kennedy et al. (2018) onderzochten deze methode door tags met een gewicht van 0.28 gram (batterij van 4 tot 12 dagen en een bereik van 800 meter) aan de ventrale zijde van het insect te binden. Uit hun onderzoek bleek namelijk dat de hoornaars nog goed konden vliegen, als de tag niet meer woog dan 80% van hun eigen lichaamsgewicht. Hierbij werd rekening gehouden dat een werkster minimum 0.35 gram weegt. Kim et al. (2019) concluderen uit hun onderzoek dat een sensor maximum 0.25 g mag wegen, iets wat goed aansluit bij Kennedy et al. (2018). Acht Aziatische hoornaars werden in het onderzoek van Kennedy et al. (2018) van een tag voorzien (6 in Frankrijk en 2 in Jersey), waarbij het gelukt is ze allemaal te kunnen volgen en vijf nesten te vinden. De drie hoornaars die de onderzoekers niet tot een nest leidden, waren in onbegaanbaar gebied gevlogen of waren gestopt met vliegen door een regenbui. Radiotelemetrie is dus wel degelijk een methode die zeer efficiënt is, niet alleen in het vinden van de nesten maar ook in tijd. De auteurs vermelden

namelijk dat het proces, van tag aanbrengen tot nest vinden, maximum 2 tot 3 uur duurt voor 1 of 2 personen. Een groot nadeel is het prijskaartje dat er aan vasthangt. Kennedy et al. (2018) gebruikten namelijk een antenne en ontvanger van samen £2000 (€ 2335) en elke tag kostte £140 (€163). Gezien het feit dat dergelijke tags meestal slechts eenmaal kunnen worden gebruikt, kan de prijs snel oplopen.

Naast radiotelemetrie zijn er de laatste jaren nog andere methodes ontworpen om individuele hoornaars te kunnen volgen naar hun nest. Eén daarvan is het gebruik van een harmonische radar, dat in tegenstelling tot radiotelemetrie een passief detectiesysteem is (Lioy et al., 2021). De tags die aangebracht zijn op de insecten sturen zelf geen signaal uit, maar reflecteren een signaal van de radar. Daardoor is in tegenstelling tot radiotelemetrie geen batterij nodig, waardoor het gewicht van de tag vermindert. Lioy et al. (2021) spreken over een betrouwbaar bereik van 750 meter voor deze techniek. De auteurs vermelden eveneens dat de efficiëntie van het opsporen van nesten (63%), hetzelfde is als het onderzoek met radiotelemetrie van Kennedy et al. (2018). Maar ook bij deze techniek is de initiële prijs, die hoger ligt dan bij radiotelemetrie, een grote beperking.

Lioy et al. (2021) onderzochten aan de hand van deze harmonische radar onder andere de gemiddelde vliesnelheid. Voor foeragerende hoornaars vonden ze een gemiddelde grondsnelheid van 6.66 ± 2.31 m/s ($n = 130$) en voor hoornaars die terug naar het nest vlogen een gemiddelde snelheid van 4.06 ± 1.34 m/s ($n = 186$). De onderzoekers concludeerden aan de hand van een Wilcoxon rank-sum test een significant verschil ($W = 19.830$, $P < 0.001$) tussen beide snelheden en verklaren dit door het feit dat terugkerende hoornaars vaak belast zijn met extra gewicht, afkomstig van gevangen prooien. Anderzijds kan het ook verklaard worden doordat hoornaars sneller vliegen wanneer ze aan het jagen zijn. Een gemiddelde van deze 2 waarden zou dus kunnen genomen worden, om te gebruiken bij afstandsbepalingen van nesten, maar dan moet de tijd die de hoornaar in het nest verblijft ook mee in rekening gebracht worden. Deze retentietijd kan sterk variëren afhankelijk van de temperatuur of de drukte rond het nest (Van Ransbeeck et al., 2023).

Een gevonden nest kan in Vlaanderen gemeld worden op de site van Vespa-Watch (Schoonvaere et al., 2020), waarna er actie ondernomen wordt om het nest te verdelgen. Deze auteurs vermelden dat het vernietigen van deze nesten enkel een werk is voor professionelen. Hoornaars reageren namelijk zeer agressief als hun nest gevaar loopt en kunnen personen steken met gif, als speciale wespenpakken en gelaatsbescherming niet gedragen worden. Omdat in de late herfst alle werksters en mannelijke hoornaars sterven en bevruchte koninginnen (die het volgende jaar een heel nieuwe cyclus kunnen starten) een overwinteringsplaats zoeken (Lioy et al., 2019), dient de nestverdelging idealiter voor deze periode te gebeuren (Schoonvaere et al., 2020).

2.7 Deep learning als nieuwe bestrijdingsmethode

Artificiële intelligentie (AI) is in ecologie en natuurbehoud sterk in opmars (AI Voor het Meten en Monitoren van Biodiversiteit en Fauna | WUR, 2024). Deep learning technieken, in het bijzonder, hebben zich bewezen als een krachtig hulpmiddel voor onderzoekers om biodiversiteit te meten en monitoren (Christin et al., 2019). In tegenstelling tot klassieke machine learning methoden, kunnen deep learning methoden zelfstandig kenmerken (features) identificeren uit de trainingsdata tijdens het trainingsproces. Handmatige feature-extraction door ontwikkelaars is dus overbodig. Hierdoor kunnen complexe features worden geïdentificeerd

die anders mogelijk over het hoofd zouden gezien worden. Deep learning maakt gebruik van neurale netwerken met meerdere lagen die in staat zijn complexe patronen te modelleren (Christin et al., 2019).

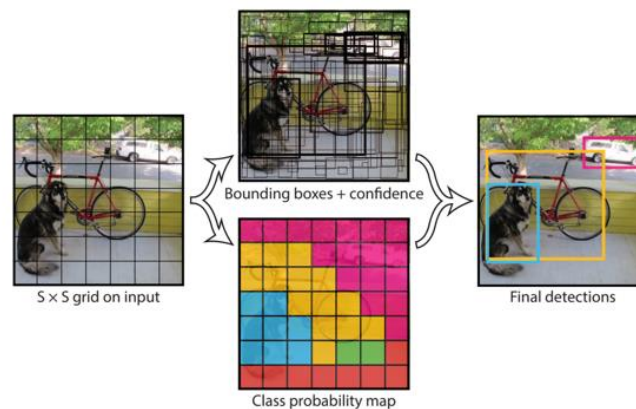
Objectdetectiemodellen trachten de aanwezigheid van relevante objecten in een afbeelding te detecteren en te classificeren in de juiste klassen (Solawetz, 2024a). Met de opkomst van convolutionele neurale netwerken (CNN), die gespecialiseerd zijn in beelddata (Høye et al., 2021), heeft deep learning de klassieke methoden overtroffen in objectherkenning (Jeon et al., 2023). Ook voor het automatisch detecteren en classificeren van insecten in beeldmateriaal zijn CNN-modellen veelbelovend. In de publicatie van Kasinathan et al. (2021) wordt de classificatienauwkeurigheid van insecten vergeleken tussen verschillende machine learning technieken, waarbij geconcludeerd wordt dat het CNN-model het best presteert. Het CNN-model overtrof de artificial neural networks, support vector machine, k-nearest neighbors en naive bayes.

Het onderzoek van Kasinathan et al. (2021) past machine learning-technieken toe om insectenplagen op landbouwgewassen te detecteren en te bestrijden, wat een veelvoorkomende toepassing van objectherkenning is op insecten. Het is vanzelfsprekend dat deze technologie ook kan worden toegepast voor veldstudies binnen het kader van de biodiversiteit van insecten (Christin et al., 2019). Bijvoorbeeld, het onderzoek van Bjerger et al. (2021) gebruikt een camerasysteem uitgerust met objectherkenningssoftware om de afname van abundantie, diversiteit en biomassa van insecten in kaart te brengen. Een groot voordeel van monitoring van insecten met objectherkenning is dat het non-destructief is, aangezien er geen directe fysieke interactie met de insecten of hun omgeving vereist is. Hierdoor wordt de verstoring van de insecten of hun omgeving tot een minimum beperkt. Bovendien kunnen grote datasets die ontstaan bij het gebruik van automatische 'camera traps' vermeden worden, omdat enkel ecologisch relevante data wordt opgeslagen (Bjerger et al., 2021).

Voor real-time monitoring van insecten is computationele snelheid van groot belang (Bjerger et al., 2021). Het open-source deep learning objectherkenning algoritme 'You Only Look Once' (YOLO) is momenteel state-of-art op dit gebied, en wordt hoog aangeprezen binnen de computer vision wereld. Zijn populariteit komt voort uit zijn aanzienlijke nauwkeurigheid en snelheid, terwijl het model compact is waardoor het eenvoudig kan geïmplementeerd worden op edge hardware zoals de Raspberry Pi (Solawetz, 2023).

YOLO is een single-stage algoritme dat in één keer voorspellingen doet van klasse-probabiliteiten en bounding boxes (Redmon et al., 2016). Dit verschilt van de aanpak van two-stage objectherkenning algoritmes. Terwijl two-stage algoritmes zoals Faster R-CNN en Mask R-CNN werken door eerst de 'regions of interest' te detecteren met behulp van het Region Proposal Network, en vervolgens afzonderlijk herkenning uit te voeren op die regio's, doet YOLO al zijn voorspellingen door één enkel regressieprobleem op te lossen. Two-stage algoritmes voeren hierdoor meerdere iteraties uit voor dezelfde afbeelding, terwijl single-stage algoritmes zoals YOLO slechts één iteratie nodig hebben, waardoor het algoritme computationeel zeer efficiënt is. (Redmon et al., 2016; Soviany en Ionescu, 2018). Hierdoor is de snelheid van het YOLO-algoritme uitzonderlijk hoog (Redmon et al., 2016), wat een vereiste is voor real-time monitoring (Bjerger et al., 2021). Sinds de oorspronkelijke release van YOLO in 2015 zijn er verschillende versies van dit model uitgekomen, waarvan YOLOv8 de meest recente is (Jocher et al., 2023).

Het regressieprobleem wordt door het YOLO-algoritme opgelost door de afbeelding eerst te verdelen in een rooster. Elke cel is verantwoordelijk voor het detecteren van de objecten waarvan het centrum zich in dat vakje bevindt. Voor elke cel voorspelt YOLO een aantal bounding boxes, rechthoekige kaders die het object omlijnen, en de zekerheden van deze bounding boxes. Deze zekerheid weerspiegelt hoe zeker het model is dat er zich een object in de box bevindt, en hoe accuraat de box is. Elk vakje berekent bovendien de klasse-probabiliteiten, die worden vermenigvuldigd met de bounding box-zekerheid om zo tot de finale detectie te bekomen (zie figuur 5) (Redmon et al., 2016). Een techniek die YOLO toepast bij post-processing is non-maximal suppression. Dit mechanisme vermijdt dat er meerdere bounding boxes voor één enkel object worden gegenereerd. Hierdoor is de mean Average Precision (mAP) (zie hoofdstuk 'Resultaten' voor gedetailleerde beschrijving) van YOLO 2-3% hoger (Redmon et al., 2016).



Figuur 5. Illustratie van het YOLO-model, waarbij het regressieprobleem in een enkele stap wordt opgelost (Redmon et al., 2016).

Beeldherkenning, dat analoog is aan objectherkenning, voor de specifieke detectie van Aziatische hoornaars is reeds onderzocht in verschillende studies: Jeon et al. (2023) bouwde een real-time monitoring systeem dat Aziatische hoornaars kan detecteren en de gebruiker hiervan een melding stuurt via een app. Hierbij werd het YOLOv5-model dat getraind werd op 2358 afbeeldingen geïmplementeerd op een Jetson Xavier NX GPU. Het systeem werd getest bij 3 bijenkasten, waarbij het in staat bleek om 83,3% van de Aziatische hoornaars te detecteren. Ook het onderzoek van O'Shea-Wheller et al. (2024), dat gepubliceerd werd tijdens het schrijven van deze bachelorproef, leverde veelbelovende resultaten. Ze trainden een YOLOv5s-model dat een mAP van > 99% behaalt voor het herkennen van Aziatische hoornaars. Het model wordt geïmplementeerd op een Raspberry Pi 4. Herrera et al. (2022) experimenteerde met optische sensoren en machine learning, waarbij door middel van vleugelslagopnames een gemiddelde nauwkeurigheid van 74,5% voor de classificatie van *Vespa velutina* werd vastgesteld.

Naast optische sensoren is er ook al geëxperimenteerd met akoestische sensoren, inclusief in België. Als onderdeel van een Citizen Science project zijn 35 Arnia-systemen geïnstalleerd bij imkers, waarbij een geluidssensor aan de ingang van de bijenkorf is geplaatst voor de detectie van hoornaars. Helaas blijkt het systeem niet betrouwbaar te zijn vanwege de frequente valse meldingen (Schoonvaere et al., 2020).

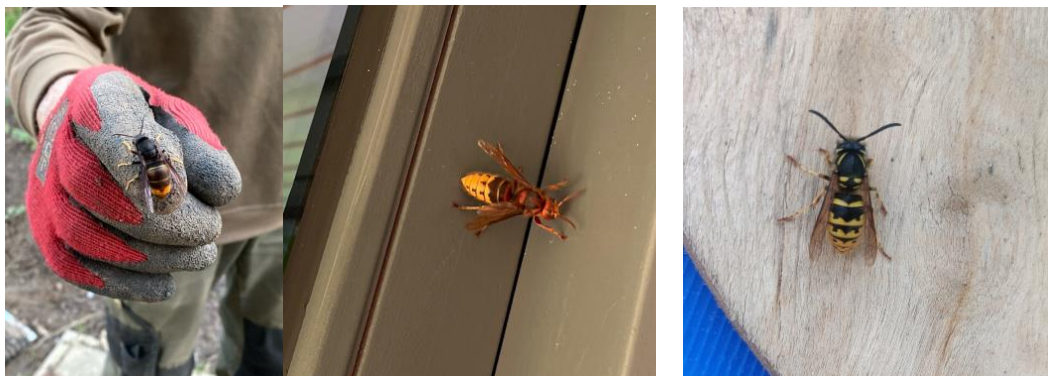
Tot op heden heeft geen van de bestaande studies voor het detecteren van hoornaars gebruik gemaakt van de state-of-art YOLOv8, dat volgens Ultralytics (de ontwikkelaar van YOLOv5 en YOLOv8) een betere prestatie levert dan YOLOv5 (Jocher et al., 2023). Deze bachelorproef brengt hier verandering in.

3 MATERIALEN EN METHODES

3.1 Beeldherkenning met YOLOv8

3.1.1 Verzameling van beelden

De dataset die gebruikt is voor het trainen van het YOLOv8-model bestaat uit afbeeldingen van drie soorten insecten: de Aziatische hoornaar, de Europese hoornaar en de gewone wesp (*Vespula vulgaris*). Deze dataset is samengesteld door handmatig afbeeldingen te selecteren van Waarnemingen.be, waarbij alleen afbeeldingen zijn gekozen waarop het insect duidelijk herkenbaar is. Omdat het model uiteindelijk wordt geïmplementeerd in een camerasysteem dat het insect van bovenaf bekijkt, zijn voornamelijk afbeeldingen gekozen met ditzelfde perspectief. Voorbeelden van zulke afbeeldingen zijn te zien op figuur 6.



Figuur 6. Voorbeelden van afbeeldingen geselecteerd van Waarnemingen.be (Aziatische hoornaar, Europese hoornaar en gewone wesp) (Aziatische hoornaar | Waarnemingen.be, z.d. ; Europese hoornaar | Waarnemingen.be, z.d. ; Gewone wesp | Waarnemingen.be, z.d.).

3.1.2 Annotatie

De annotatie van de geselecteerde foto's werd uitgevoerd op Roboflow (Dwyer et al., 2024). Dit is een platform voor computervisie dat het annoteren, preprocessen en augmenteren van beelden gebruiksvriendelijk maakt (Teoh, 2023). Na het uploaden van de beelden op Roboflow werd er rond elk insect een bounding box getekend (zie figuur 7), en werd het type insect als klasse toegevoegd.



Figuur 7. Bounding box rond een Aziatische hoornaar, met aanduiding van de klasse.

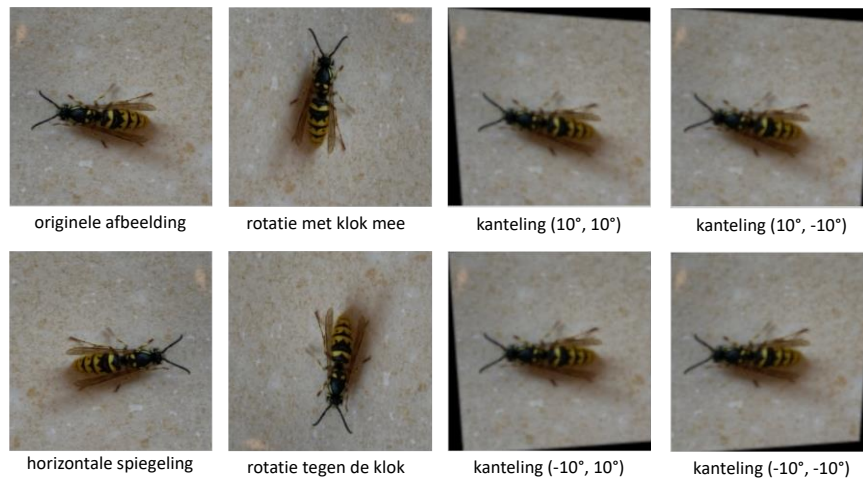
Roboflow beschikt over openbare datasets met reeds geannoteerde beelden van hoornaars en wespen. De beste daarvan zijn in dit onderzoek zorgvuldig geselecteerd en samengevoegd met de beelden van Waarnemingen.be. De uiteindelijke dataset die gebruikt is voor het beeldherkenningsmodel bevat 545 geannoteerde foto's, waarvan het overgrote deel afkomstig is van Waarnemingen.be.

3.1.3 Preprocessing

Bij preprocessing worden bewerkingen op de ruwe beeldgegevens uitgevoerd zodat deze beter geschikt zijn voor het trainen van het machine-learning model. Dit is een essentiële stap alvorens de beelden gebruikt kunnen worden. Roboflow past automatisch enkele basis preprocessing bewerkingen uit, waarbij alle afbeeldingen tot dezelfde grootte worden gestretcht (Preprocess Images | Roboflow Docs, z.d.)

3.1.4 Augmentatie

Over het algemeen leidt het gebruik van grotere datasets tot betere deep learning modellen (Shorten en Khoshgoftaar, 2019). Om het aantal afbeeldingen voor de training van het model te vermeerderen, is augmentatie toegepast. Dit is een techniek waarbij een reeks transformaties op de afbeeldingen wordt toegepast waardoor er 'nieuwe' afbeeldingen ontstaan waarom het model kan trainen (Shorten en Khoshgoftaar, 2019). Onze dataset is artificieel uitgebreid door sommige van de originele beelden te roteren, horizontaal te spiegelen of een kanteling van 10° toe te passen, zoals weergegeven in figuur 8.



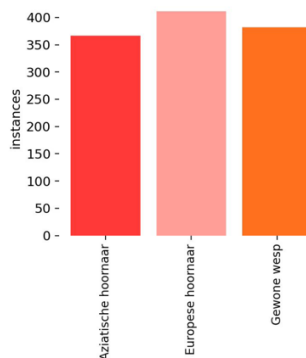
Figuur 8. Illustratie van augmentatie.

3.1.5 Training van YOLOv8 op de dataset

Voordat het YOLOv8 model getraind kan worden, moet de dataset gesplitst worden in een training, validatie- en testdataset. De training dataset wordt gebruikt om het model te trainen, waarbij het model wordt geoptimaliseerd om patronen te leren uit de trainingsgegevens. Tijdens het trainen wordt de validatiedataset gebruikt om in te schatten hoe goed het model werkt op afbeeldingen die niet zijn gebruikt tijdens het trainen. Dit stelt ons in staat om de prestaties van het model te monitoren en te controleren op overfitting. Overfitting treedt op wanneer een machine learning-model zich te sterk aanpast aan de trainingsbeelden, waardoor het model zeer nauwkeurig is op de trainingsbeelden, maar slecht presteert op nieuwe, niet eerder geziene beelden (Solawetz, 2024b).

Finaal wordt het model geëvalueerd op de testdataset, die niet eerder gebruikt is tijdens de training of validatie. Het doel is om de prestaties van het model te beoordelen. Door de dataset op deze manier te splitsen, kunnen we ervoor zorgen dat het model robuust is en goed presteert op nieuwe, ongeziene afbeeldingen (Solawetz, 2024b).

Roboflow voert de splitsing van de dataset automatisch uit bij het uploaden van de beelden op het platform. Onze dataset is verdeeld in drie sets: 384 trainingsbeelden, die zijn uitgebreid tot 1152 trainingsafbeeldingen door middel van data-augmentatie, 106 validatiebeelden en 55 testbeelden. Van elk van de insectenklassen zijn er ongeveer evenveel beelden, om over- of onderrepresentatie van een bepaalde klasse in de dataset te vermijden (zie figuur 9).



Figuur 9. Visualisatie van de trainingsset na data-augmentatie.

Het YOLOv8 model is getraind met behulp van een Google Colab Notebook, die geschreven werd door Roboflow (How to Train YOLOv8 Object Detection on a Custom Dataset | Google Colaboratory, z.d.). Een Colab-notebook biedt de mogelijkheid om Python-code uit te voeren op de cloudservers van Google. Dit stelt gebruikers in staat om gratis gebruik te maken van de rekenkracht van Google-hardware, waaronder GPU's en TPU's, met alleen een browser als vereiste. Om deze reden is het een veelgebruikte tool binnen de machine learning-community (Welcome To Colab | Google Colaboratory, z.d.). In **Appendix A** wordt een samenvatting gegeven van wat werd uitgevoerd in de Colab-notebook om het YOLOv8 model te trainen.

Voor de training van het YOLOv8 model zijn 110 trainingstappen uitgevoerd, bestaande uit afwisselende training- en validatiefasen. Tijdens de trainingfase wordt een willekeurige batch afbeeldingen uit de trainingdataset geëxtraheerd. Deze afbeeldingen worden door het model geleid, waarbij het model de bounding boxes en klassen in het beeld voorspelt. Daarna wordt het voorspelde resultaat vergeleken met het correcte resultaat met behulp van een verliesfunctie, die de grootte van de fout tussen het voorspelde en correcte resultaat berekent. Op basis van deze fout worden de gewichten van het model aangepast, zodat het model zich tijdens de training verbetert (Germanov, 2023).

In de validatiefase worden afbeeldingen uit de validatiedataset gebruikt. Ook deze worden door het model geleid, waarbij het model opnieuw de bounding boxes en klassen in het beeld voorspelt. Door het verschil tussen het voorspelde resultaat en het correcte resultaat te evalueren, kan de precisie van het model worden bepaald (Germanov, 2023).

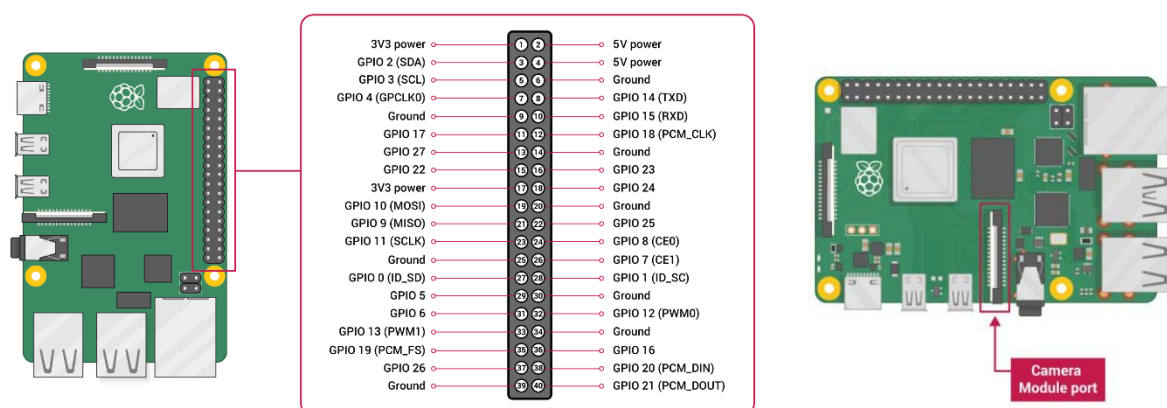
Na het voltooien van de training van het model, kan het geëvalueerd worden aan de hand van de testdataset en vervolgens geëxporteerd worden als bestand met de extensie '.pt'. Met dit bestand kan het model offline werken, bijvoorbeeld op een Raspberry Pi. Het model dat in deze bachelorproef is getraind, is beschikbaar in de GitHub-repository. De evaluatie van het model wordt beschreven in het hoofdstuk 'Resultaten'.

3.2 Gebruikte hardware, basisconfiguraties & opstelling

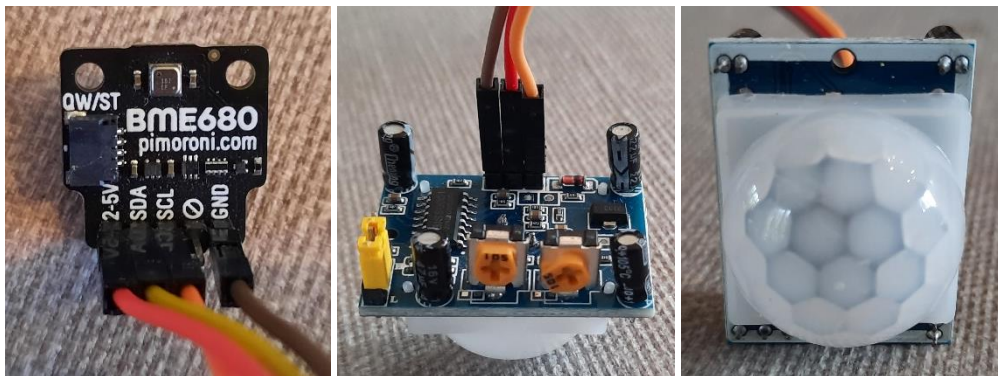
Om dit project tot een goed einde te brengen hebben we gebruik gemaakt van een Raspberry Pi 4 model B van 4 GB RAM (zie overzichtfiguur 10), gecombineerd met een 64 GB SD kaart waarop het Raspberry Pi operating system Legacy 64 bit (Raspbian Bullseye) geïnstalleerd is. Hoe dit operating system geïnstalleerd kan worden, staat uitvoerig besproken op de website van Raspberry Pi zelf (Raspberry Pi documentation: Getting started with your Raspberry Pi | raspberrypi.com, z.d.). Vanzelfsprekend dient de Raspberry Pi voorzien te worden van stroom doormiddel van een USB-C kabel. Om de Raspberry Pi gemakkelijk te kunnen programmeren, dient deze gekoppeld te worden met een scherm via een micro HDMI-poort (aanwezig op de Raspberry Pi). Het aansluiten van een muis en toetsenbord via de USB-poorten is een must. Verder is een Raspberry Pi Camera Module 2 gebruikt, in combinatie met een bewegingssensor HC-SR 501 en een omgevingssensor BME680 die de temperatuur, luchtvochtigheid en luchtdruk meet. 7 dupont jumper cables (female to female) zijn gebruikt om de laatste 2 sensoren te verbinden met de Raspberry Pi. Om onafhankelijk in het veld te kunnen werken is de Raspberry Pi ook nog uitgerust met een PiJuice HAT, aangesloten op een batterij van 12000 mAh en een PiJuice Solar Panel van 20 W.

3.2.1 Sensoren

Nadat de Raspberry Pi voor de eerste keer opgestart is, kunnen deze sensoren aangesloten worden en bijhorende packages gedownload worden. De BME680 moet, nadat het gesoldeerd is (Getting started with BME680 Breakout | Pimoroni, z.d.), aangesloten worden met 4 dupont kabels met de GPIO header van de Raspberry Pi (zie figuur 10). Belangrijk hierbij is dat de Raspberry Pi op geen enkele manier verbonden is met een stroombron, om elektrocutie te vermijden. De meest linkse kabel op figuur 11 (links), die dient voor de 2-5V stroomvoorziening, kan verbonden worden met pin 4 op figuur 10. De kabels voor SDA en SDL kunnen respectievelijk met pin 3 en 5 verbonden worden en de ground (GND) met pin 9 op figuur 10. Eenmaal correct aangesloten, kan in de terminal het benodigde Python package gedownload worden met `sudo pip3 install bme680`. Ten slotte moet I2C nog aangezet worden via de interface options van de pi, die te bewerken zijn via `sudo raspi-config`. Het camera lint (met het blauwe stukje plastic op het lint naar de grote USB poorten gericht), laat toe de camera eenvoudig te verbinden met de camera module poort op figuur 10 (De Boer, 2020).



Figuur 10. Links: functies van de GPIO pins op de Raspberry Pi (Raspberry Pi documentation: Getting started with your Raspberry Pi | raspberrypi.com, z.d.). Rechts: aanduiding van de camera aansluitingsplaats (De Boer, 2020).



Figuur 11. Eigen gesoldeerde BME680 met bedrading (links) en bewegingssensor HC-SR 501 (midden & rechts).

“import bme680” importeert het bme680 package in een script, waarbij de sensor aangeroepen wordt met “sensor = bme680.BME680()”. Omdat er soms geen apparaat wordt gedetecteerd op het primaire I2C-adres, kan een try-except blok gebruikt worden met het secundaire I2C-adres om te verzekeren dat de sensor op het juiste adres wordt uitgelezen (megatommy, 2021):

try:

```
sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)
```

except IOError:

#input/output fout

```
sensor = bme680.BME680(bme680.I2C_ADDR_SECONDARY)
```

Eveneens moeten een paar voorinstellingen gemaakt worden:

```
sensor.set_humidity_oversample(bme680.OS_2X)
```

```
sensor.set_pressure_oversample(bme680.OS_4X)
```

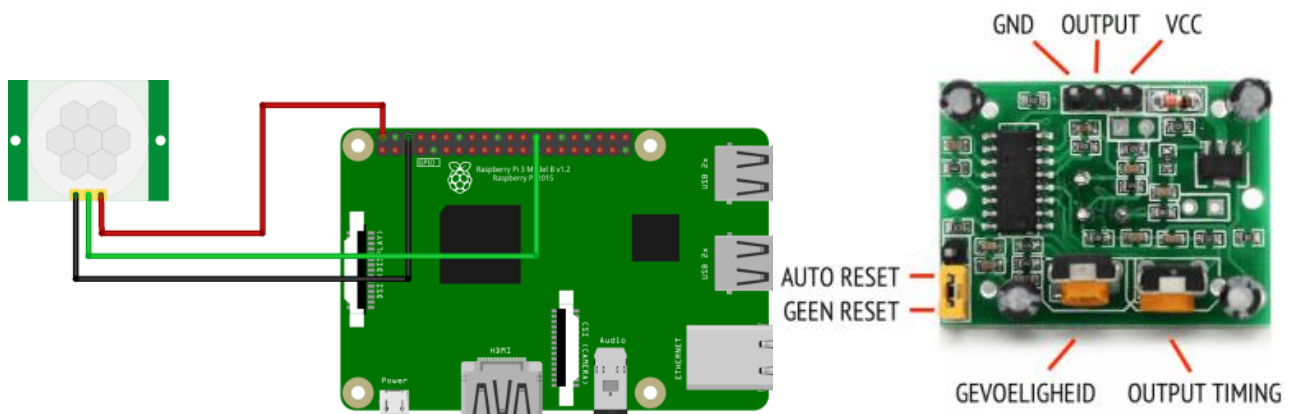
```
sensor.set_temperature_oversample(bme680.OS_8X)
```

```
sensor.set_filter(bme680.FILTER_SIZE_3)
```

De oversample instellingen voor de vochtigheid, druk en temperatuur zorgen voor een goede balans tussen signaal en ruis. Hoe groter de oversampling, hoe minder ruis, maar eveneens meer verlies aan accuraatheid. De filter zal er voor zorgen dat er geen korte, scherpe veranderingen in omgevingscondities worden uitgelezen. Bij het uitzetten van de data in een grafiek, zullen er dus minder grote pieken en dalen zijn (Getting started with BME680 Breakout | Pimoroni, z.d.). Een output van de data kan uiteindelijk bekomen worden met: `output = "{0:.2f} C,{1:.2f} hPa,{2:.2f} %RH".format(sensor.data.temperature, sensor.data.pressure, sensor.data.humidity)`. Hierbij geven we aan met .2f, dat elke waarde een float is met 2 cijfers na de komma.

De bewegingssensor op figuur 11 (midden & rechts) moet ook aangesloten worden met 3 female to female dupont kabels. De VCC pin op de sensor (figuur 12, rechts) moet verbonden worden met een 5V pin op de pi, zoals pin 2. De ground (GND) op de sensor mag alleen verbonden worden met een ground pin op de pi, een voorbeeld is pin 6. De output pin kan vervolgens met meerdere mogelijke GPIO pinnen verbonden worden

op de pi. Ijzermans (2016) gebruikt, zoals te zien op figuur 12 (links), pin 26 terwijl wij voor dit project pin 18 zullen gebruiken. Zoals te zien is op figuur 12 (rechts) zijn op de sensor zelf nog handmatige instellingen mogelijk, zoals de afstand van waarneming en dus de gevoeligheid (3 tot 7 m). Voor dit project gebruiken we de standaardinstellingen. De bewegingssensor werkt op basis van veranderingen in infrarood licht ten gevolge van beweging. Deze sensor heeft een brede kijkhoek, waardoor deze ook beweging detecteert die niet op de wiekpot plaatsvindt. Om dit probleem op te lossen hebben we simpelweg een stukje zwarte tape aan de randen van de sensor geplakt.



Figuur 12. Links: aansluitschema van de HC-SR501 op de Raspberry Pi. Rechts: functie van elke pin en mogelijke hardware instellingen (Ijzermans, 2016).

3.2.2 PiJuice

Voor de stroomvoorziening van de Raspberry Pi hebben we geopteerd voor een opstelling die off-grid kan werken. Op deze manier kan VELU-PI ook op sterk afgelegen locaties, die zich ver van een stroomvoorziening bevinden, functioneren. De opstelling is gebaseerd op een PiJuice HAT, wat staat voor Hardware Attached on Top. Zoals de naam doet vermoeden, wordt deze extra hardwaremodule bevestigd bovenop de Raspberry Pi. Om de PiJuice HAT te installeren dien je simpelweg de GPIO-pins van de HAT boven op de GPIO-pins de Raspberry Pi zelf te houden en voorzichtig, maar stevig neer te duwen. De sensoren kunnen verbonden worden met de pins op de HAT, net zoals ze rechtstreeks met de Raspberry Pi zouden worden verbonden. Vervolgens kan de standaardbatterij uitgeklikt worden. Om de mogelijke meettijd te verlengen is het aangewezen om een grotere batterij te gebruiken. De externe batterij van PiJuice zelf wordt aangesloten zoals voorgetoond is op figuur 13.

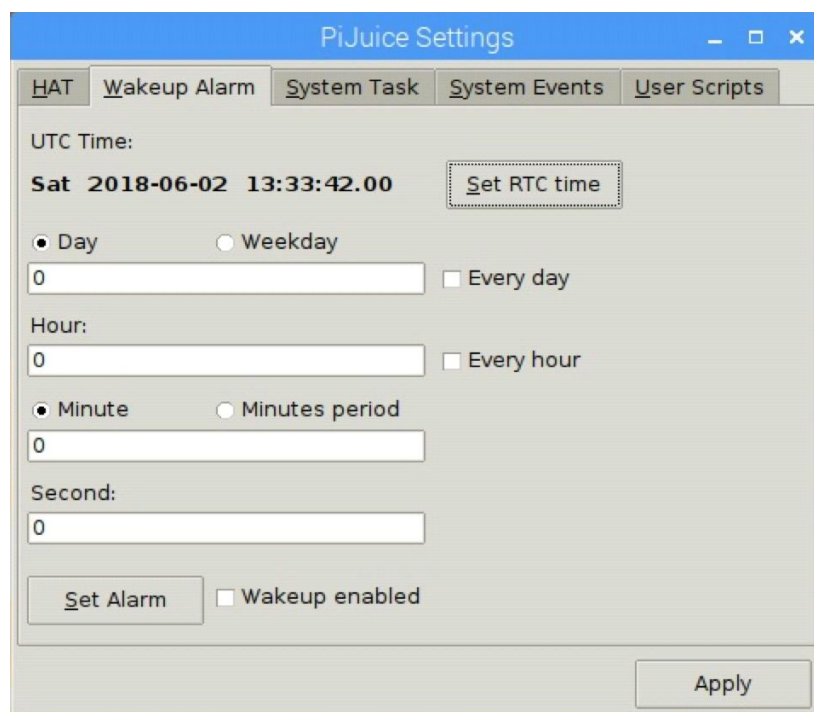


Figuur 13. Correcte aansluiting van de externe batterij op de PiJuice HAT.

Vervolgens kan via de meegeleverde kabel het zonnepaneel aangesloten worden op de PiJuice HAT via de micro USB. Vanaf dit moment zal het zonnepaneel de batterij beginnen opladen. De batterij zal vanaf dit moment in staat zijn om de Raspberry Pi van stroom te voorzien. Een nadeel hiervan is dat de Raspberry Pi niet vanzelf opstart, wat anders wel gebeurt wanneer hij aangesloten wordt op netstroom. Om dit probleem op te lossen hebben de ontwikkelaars van de PiJuice een aan/uit-knop toegevoegd op de zijkant van de PiJuice HAT.

Een voordeel van de PiJuice HAT is dat de instellingen aanpasbaar zijn aan de noden van de gebruiker. Om dit eenvoudig te kunnen doen moet je eerst het softwarepakket downloaden via `sudo apt-get install pijuice-gui`. GUI staat hier voor 'graphical user interface'. Na het downloaden van de software en het heropstarten van de Raspberry Pi, kan je via Menu > Preferences > Pij Juice settings naar de interface. Van hieruit kan je je persoonlijke instellingen aanduiden. Via het tabblad System Task kan je ten eerste al een minimale batterijlading instellen. Wanneer de batterij onder deze drempelwaarde duikt, zal de Raspberry Pi afgesloten worden. Dit is noodzakelijk om de SD-kaart en de verzamelde data te beschermen, aangezien beide negatieve gevolgen ondervinden van een plotseling verval van de stroom.

Via de PiJuice settings is het ook mogelijk om de Raspberry Pi op voorafbepaalde momenten op te starten. Via het tabblad Wakeup Alarm kan je dagelijkse of uurlijkse momenten vastleggen waarop de opstelling moet opstarten. Op figuur 14 staat een voorbeeld van hoe dit er uit ziet. Het enige probleem is dat er geen mogelijkheid is om de Raspberry Pi automatisch uit te schakelen via de interface. Dit probleem is echter snel opgelost door een script te laten lopen, dat de Raspberry Pi laat afsluiten na een bepaalde tijd. Voor een voorbeeld verwijzen we naar **Appendix B**. Het script is ook terug te vinden in de VELU-PI Github-repository. Extra mogelijkheden kunnen steeds teruggevonden worden op de Github van PiJuice (PiSupply, z.d.).

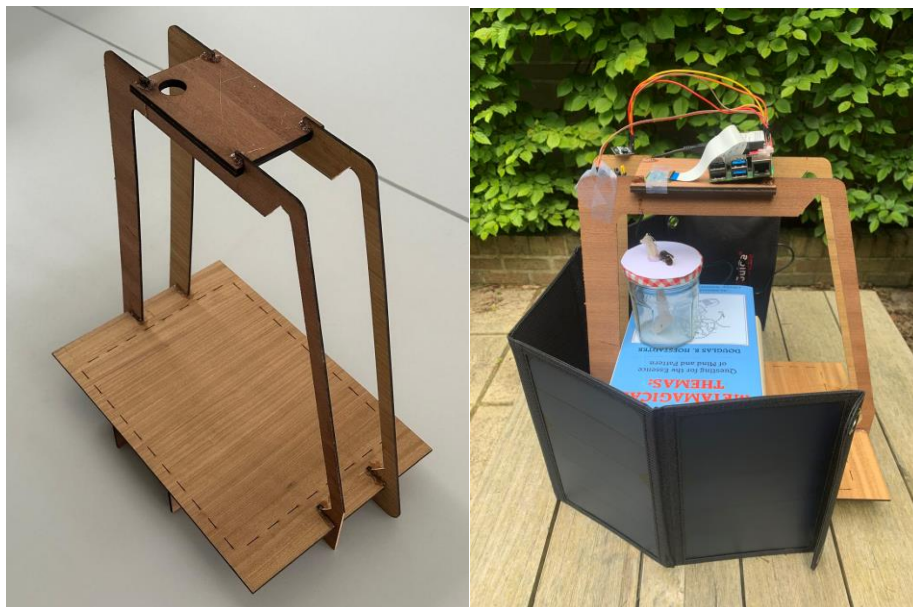


Figuur 14. Voorbeeld van de graphical user interface op het tabblad Wakeup Alarm (PiSupply, z.d.).

Als je dit gewoon zo instelt zal het echter nog niet direct werken. Na een reboot zal de Raspberry Pi de interne klok van de PiJuice resetten, waardoor deze niet meer in staat is om uit zichzelf wakker te worden. Om dit probleem te verhelpen moet je het script uit **Appendix C** in de cron table zetten. Het script is ook terug te vinden in de VELU-PI Github-repository. Als het script in de cron table staat zal het script automatisch doorlopen worden bij reboot, en zal de interne klok van PiJuice HAT beschermd worden tegen reset. Je kan aan de cron table geraken via de command `crontab -e`, waarna je de lijn `@reboot /usr/bin/python3 /home/pi/PiJuice/Software/Test/wakeup_enable.py` toevoegt om het script automatisch te doorlopen.

3.2.3 Opstelling

Om ervoor te zorgen dat de camera en bewegingssensor op een vaste afstand van de wiekpot blijven, is een robuuste opstelling essentieel. Terwijl verschillende budgetvriendelijke opties denkbaar zijn, zoals bijvoorbeeld een zelfgemaakte houten constructie, hebben wij gekozen voor een opstelling vervaardigd met behulp van een lasercutter. Dit specifieke ontwerp (te zien in figuur 15, links) werd oorspronkelijk ontwikkeld door Koch (z.d.) voor het scannen van documenten met een iPhone. Het bestaat uit een basis en een houder waar de Raspberry Pi stevig op kan worden gemonteerd. In deze houder is een opening voorzien voor de lens van de camera. Op de basisplaat staat de wiekpot, gevuld met een mengsel van suiker en alcohol of Trappit. Om een zo scherp mogelijk beeld te bekomen kan de hoogte van deze wiekpot aangepast worden of kan de lens van de Raspberry Pi camera scherp gesteld worden door deze licht te draaien. Om de bewegingssensor te richten op de bovenkant van de wiekpot, kan deze vastgemaakt worden aan een been van de gelasercutte constructie. Het zonnepaneel die met de PiJuice verbonden is, kan zoals op figuur 15 (rechts) rond de opstelling gezet worden in de richting van de zon, eventueel onder een hoek voor grotere efficiëntie. Dit prototype is op zichzelf nog niet bestand tegen weer en wind, waardoor het best geplaatst wordt onder een waterdichte beschutting.



Figuur 15. Links: Houten basisopstelling vervaardigd met een lasercutter. Rechts: Eigen finale prototype, waarbij de wiekpot op de basisplaat komt (hier met een verhoging) en de Raspberry Pi op de houder bevestigd wordt. De bewegingssensor en camera zijn vastgehecht met plakband. Het zonnepaneel staat rond de opstelling naar de zon gericht.

3.3 Aanmaken van een lokale MySQL database

Data is een kostbaar goed en we willen het in ons geval dan ook niet verloren zien gaan. Het opslaan van deze data in een lokale relationele databank zoals MySQL is dan een eerste stap om actief aan de slag te kunnen gaan met deze data. Er zijn veel verschillende relationele databanken die kunnen gebruikt worden, maar voor een Raspberry Pi wordt vaak gebruik gemaakt van MariaDB. Om een databank te kunnen laten werken op een web server (LAMP server) moeten er naast de databank zelf, nog andere software pakketten geïnstalleerd worden. Een eerste is Apache2, wat de meest gebruikte web server interface is. Dit zal de verzoeken, die in een web browser worden ingegeven, behandelen en een gepaste pagina in de browser terugsturen. Daarnaast moet PHP geïnstalleerd worden, dit is een programmeertaal die gebruikt wordt voor het maken van websites (Santos R. en Santos S., z.d.). Nadat ook de databank zelf geïnstalleerd is, kan je de administratie van zo'n databank enorm vereenvoudigen door het gebruik van een gratis tool als phpMyAdmin. Via een web server als phpMyAdmin krijg je een meer visueel overzicht van alle tabellen en kolommen en zijn bewerkingen makkelijker uit te voeren dan in de Raspberry Pi terminal. Om de lokale databank op een optimale manier te laten werken dienen de installaties van de software pakketten best te gebeuren in dezelfde volgorde als net besproken. Volgende gebruikte stappenplan is gebaseerd op Spinner A. (2021) en op Santos R. en Santos S. (z.d.).

Allereerst dient de recentste informatie van de te downloaden software pakketten opgehaald te worden door volgend commando in te geven in de terminal van de Raspberry Pi: `sudo apt-get update`. Installatie van Apache2 gebeurt met het commando `sudo apt-get install apache2 -y`. Dit zou geen problemen mogen opleveren, waarna kan overgegaan worden naar de installatie van PHP. Het commando `sudo apt install libapache2-mod-php` voert de installatie uit. Als de vraag 'Do you want to continue?' komt, moet het bevestigende antwoord `y` gegeven worden. Na succesvolle installatie, moet genavigeerd worden naar een directory via `cd /var/www`, om in deze directory het commando `sudo chown pi: html` in te geven en zo de gebruiker 'pi' toegang te geven tot alle bestanden in de publieke html folder. Hierbij dient 'pi' vervangen worden door de gebruikersnaam van de Raspberry Pi, al is dit in veel gevallen nog standaard 'pi'. Finaal wordt Apache2 best herstart met het commando `sudo /etc/init.d/apache2 restart`. Vanaf nu kan terug gewerkt worden in de homedirectory via `cd`.

MySQL (in de vorm van MariaDB server) kan gedownload worden via `sudo apt install mariadb-server php-mysql -y`, waarna een user van de databank moet aangemaakt worden. Na het commando `sudo mysql` kan dit gebeuren via `CREATE USER 'admin'@'%' IDENTIFIED BY 'password';` waar admin en password vervangen kunnen worden naar een eigen keuze. Met het percent teken geven we de gebruiker toegang van elk IP-adres. Vervolgens laten we toe dat deze user alle privileges krijgt: `GRANT ALL PRIVILEGES ON *.* to 'admin'@'%' WITH GRANT OPTION;`, waarbij admin opnieuw vervangen wordt door de eigen gekozen admin naam. De user is nu aangemaakt, waarna de database terminal kan afgesloten worden via "exit".

Uiteindelijk kan ook phpMyAdmin geïnstalleerd worden via `sudo apt-get install phpmyadmin`. Op de vraag welke web server gebruikt moet worden om phpMyAdmin te configureren, moet de eerder geïnstalleerde apache2 gekozen worden. Klik vervolgens op 'yes' om phpMyAdmin te configureren met dbconfig-common. De volgende vraag, om een paswoord in te stellen voor phpMyAdmin, kan maar hoeft niet ingevuld te worden aangezien al een user aangemaakt is voor de databank. Belangrijk is om op [ok] te drukken. Finaal moeten we Apache2 nog laten weten van de configuratie van phpMyAdmin, door een configuratiebestand van Apache2 aan te passen: `sudo nano /etc/apache2/apache2.conf`. Helemaal onderaan het bestand moet Include

`/etc/phpmyadmin/apache.conf` bijgevoegd worden, om te garanderen dat er geen problemen zullen opduiken bij het inloggen in phpMyAdmin. Nadat het bestand opgeslaan en afgesloten is, moet apache2 heropgestart worden met een al eerder gebruikt commando: `sudo /etc/init.d/apache2 restart`. Alles is nu klaar voor gebruik, maar een heropstart van de Raspberry Pi is aanbevolen: `sudo reboot`.

Om te testen als alles werkt, kan geprobeerd worden in te loggen op phpMyAdmin met de gekozen admin en paswoord door in de webbrowser van de Raspberry Pi te surfen naar `http://localhost/phpmyadmin`. Via een ander apparaat dat op hetzelfde lokale netwerk als de Raspberry Pi aangesloten is, kan je phpMyAdmin ook bereiken door localhost te vervangen door het IP-adres van de Raspberry Pi (deze is makkelijk te achterhalen door `hostname -I` in te geven in de terminal van de Raspberry Pi).

Om uiteindelijk data te kunnen opslaan in MariaDB, moet een database en een tabel aangemaakt worden. Telkens een Aziatische hoornaar herkend wordt via beeldherkenning, willen we de tijd, temperatuur, luchtdruk en luchtvochtigheid opslaan en bovendien een foto en video bewaren. De eerste variabelen kunnen makkelijk onder een float datatype opgeslaan worden. Een MySQL databank kan echter een foto of video niet rechtstreeks opslaan onder de vorm van een JPG of MP4 datatype. Zulke bestanden moeten eerst vertaald worden naar een binair datatype, vooraleer het opgeslaan kan worden als een binair bestand in de databank. Zo'n binair bestand krijgt de naam BLOB (Binary Large Object) en kent in een databank meerdere vormen: TINYBLOB (max. 255 bytes per bestand), BLOB (max. 64 KB per bestand), MEDIUMBLOB (max. 16 MB per bestand), LONGBLOB (max. 4 GB per bestand) (Blob and Text Data Types | MariaDB, z.d.). Een afbeelding met de Raspberry Pi camera heeft een bestandsgrootte die zeker onder de 16 MB ligt, waardoor een MEDIUMBLOB datatype aangewezen is. De bestandsgrootte van een video hangt dan weer af van de duur en de resolutie, waardoor een MEDIUMBLOB datatype soms niet volstaat. Als de video onder de 10 seconden blijft kan geopteerd worden voor MEDIUMBLOB, anders is LONGBLOB aangeraden. Zo is het mogelijk om met onderstaande code (gebaseerd op megatommy (2021)) in de database terminal eerst en vooral een database genaamd 'Sensordata' aan te maken, waarin een tabel genaamd 'image_video_bme680' komt met de kolommen datum, naam van de host die de data in de database plaatst (variabele met max. 50 karakters met meegegeven collation (bepaald hoe karakters worden vergeleken bij het sorteren)), afbeelding, video, temperatuur, druk en vochtigheid. Als er geen input is voor een bepaalde variabele, krijgt deze de default waarde NULL. Belangrijk is dat de kolomnamen niet tussen aanhalingstekens staan, maar tussen backticks. Onderstaande drie queries zijn ook opgenomen in de VELU-PI Github-repository.

```
CREATE DATABASE Sensordata;

USE Sensordata;

CREATE TABLE image_video_bme680 (
  `date` DATETIME DEFAULT NULL,
  `host` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `image` MEDIUMBLOB DEFAULT NULL,
  `video` LONGBLOB DEFAULT NULL,
  `temperature` float DEFAULT NULL,
  `pressure` float DEFAULT NULL,
```

```
`humidity` float DEFAULT NULL  
);
```

De mysql connector laat toe data vanuit een Python script te stockeren in deze tabel. Deze tool legt de link tussen het script en de lokale databank. Volgende methodiek is gebaseerd op megatommy (2021). Na installeren van het mysql.connector package (via `pip3 install mysql-connector-python`) kan het geïmporteerd worden in het script: `import mysql.connector`. Verbinding met de database kan op volgende manier, waarbij user, password en database correct gedefinieerd moeten worden: `mydb = mysql.connector.connect(host=localhost, user=user, password=password, database=database)`, waarna een cursor object wordt aangemaakt via: `cursor = mydb.cursor()`. De sql query om de data in een tabel in te voegen kan gedefinieerd worden met: `sql = "INSERT INTO image_video_bme680 (date,host,image,video, temperature, pressure, humidity) VALUES (%(date)s,%(host)s,%(image)s,%(video)s,%(temperature)s,%(pressure)s,%(humidity)s)".` Bij query's is het belangrijk dat er geen enkele schrijffout aanwezig is. De kolomnamen in het begin van de query en op de variabelen op het einde moeten extra aandacht krijgen. De data in het finale script (zie **Appendix D**) kan toegewezen worden aan de variabelen, die gedefinieerd zijn op het einde van de query met:

```
image_data = {  
    'date': datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'),  
    'host': host,  
    'image': image_binary,  
    'video': video_binary,  
    'temperature': '{0:.2f}'.format(sensor.data.temperature),  
    'pressure': '{0:.2f}'.format(sensor.data.pressure),  
    'humidity': '{0:.2f}'.format(sensor.data.humidity) }
```

Uiteindelijk kan de query uitgevoerd worden met een execute command: `cursor.execute(sql, image_data)`, waarna `mydb.commit()` de wijzigingen permanent maakt. Uiteindelijk beëindigt `mydb.close()` de connectie met de database.

Het uitlezen van de lokale databank met een andere computer op afstand is mogelijk, mits aanpassingen aan de beveiligingsinstellingen van MariaDB. Deze aanpassingen, samen met een voorbeeldscript voor het uitlezen van alle foto's en video's in de databank, staat uitvoerig beschreven in **Appendix E**.

3.4 Afbeeldingen en video's verzenden naar Dropbox

Een databank is zeer nuttig en interessant, maar niet altijd even praktisch om snel een nieuwe foto of video te kunnen bekijken op bijvoorbeeld je gsm. Er bestaan meerdere tools waarmee je data kunt verzenden naar een online databank. Een er van is Dropbox, waar je een gratis account kan aanmaken. Een nadeel is dat er bij dit gratis account een datalimiet is van 2 GB, dus als je deze service wilt blijven gebruiken moeten er regelmatig bestanden verwijderd worden. Volgende methodiek is gebaseerd op NeuralNine (2023). Nadat

een gratis Dropbox account is aangemaakt, kan een developer app in je account gemaakt worden via www.dropbox.com/developers. Na het kiezen van de API 'Scoped acces' kan je kiezen als je de app toegang wilt verlenen aan één bepaalde map of aan je volledige Dropbox account. Wij kozen voor het tweede, waarna je de app een gepaste naam kan geven. Om toe te laten dat de app data kan schrijven en lezen in je Dropbox account, moet na het klikken op 'Create app', er genavigeerd worden naar het tabblad 'permissions' op de website. Na het aanklikken van de hokjes: `files.content.write`, `files.metadata.read`, `files.content.write` en `files.content.read`, is het belangrijk om onderaan het scherm te klikken op 'Submit'. Terug in het tabblad 'settings' kan de belangrijke acces token gegenereerd worden. Deze laat toe om vanuit een Python script data te verzenden naar je Dropbox account.

Terug in de terminal van de Raspberry Pi kan het Dropbox package gedownload worden met het commando `pip3 install dropbox`. Na het importen van de Dropbox packages `import dropbox` en `import dropbox.files` en het toewijzen van je acces token (`dbx = dropbox.Dropbox(TOKEN)`) in een Python script, laat volgende code toe elke binaire foto te uploaden naar Dropbox (waar het te zien is in JPG formaat): `dbx.files_upload(image_binary,"/raspi/Pictures/img_{timestamp}.jpg")`. Voor een video geldt hetzelfde principe: `dbx.files_upload(image_video,"/raspi/Videos/video_{timestamp}.mp4")`. De binaire files zijn al aangemaakt voor de lokale databank, dus kunnen gerecupereerd worden. De directory die je meegeeft voor elke afbeelding of video, is de directory die in Dropbox gebruikt wordt. Als deze directory nog niet bestaat, wordt deze automatisch aangemaakt bij de eerste upload.

3.5 Finale Python script

Het finale Python script, dat zich in de map 'Documenten' van onze Raspberry Pi bevindt, is te raadplegen in **Appendix D** en de VELU-PI GitHub-repository. Om dit finale script automatisch te laten werken vanaf het opstarten van de Raspberry Pi, is gebruik gemaakt van een service bestand in de systeemcontrole. In **Appendix F** staat beschreven hoe dit ingesteld kan worden. Het bestand is ook beschikbaar in de GitHub-repository.

Nadat de benodigde packages zijn geïmporteerd, waaronder het ultralytics package (dat moet worden geïnstalleerd met het commando `pip install ultralytics`), moet naast het initialiseren van de lokale databank, Dropbox, BME680 en de bewegingssensor, ook het beeldherkenningsmodel worden ingeladen met het commando `model = YOLO('best.pt')`. Het is noodzakelijk dat het beeldherkenningsmodel 'best.pt' zich in dezelfde map bevindt als het Python script.

Met behulp van een 'while'-lus wordt er na bewegingsdetectie en een wachttijd van 5 seconden een foto genomen. Deze wachttijd is ingesteld om een insect de kans te geven om op de wiekpot te landen voordat de foto wordt genomen, om zo te voorkomen dat er een foto wordt gemaakt van een insect in volle vlucht. De foto wordt genomen door het commando `libcamera-still`. De foto krijgt het tijdstip van opname als naam. Vervolgens wordt op deze foto de beeldherkenning uitgevoerd. Pas als de beeldherkenning met meer dan 66.8% zekerheid een Aziatische hoornaar herkent, wordt met het `picamera2` package van Raspberry Pi een video opgenomen (Padrón, 2023). De video en eerder genomen foto worden vervolgens opgeslagen in de lokale databank en Dropbox, op de manier zoals in vorige paragrafen beschreven staat. De Dropbox token is voor het gemak in een apart tekst bestand geplaatst (aanwezig in dezelfde directory als het finale script), waarbij dit bestand in het begin van het script wordt uitgelezen.

Nadat VELU-PI met behulp van het beeldherkenningsmodel een Aziatische hoornaar detecteert, wordt de 'while'-lus gedurende 5 minuten gepauzeerd. Dit is een poging om te voorkomen dat dezelfde Aziatische hoornaar meerdere keren wordt gedetecteerd. De duur van dit tijdsinterval is een schatting, aangezien de verblijftijd van een hoornaar op een wiekpot sterk variabel is en er geen sluitende gemiddelde waarde beschikbaar is in de literatuur.

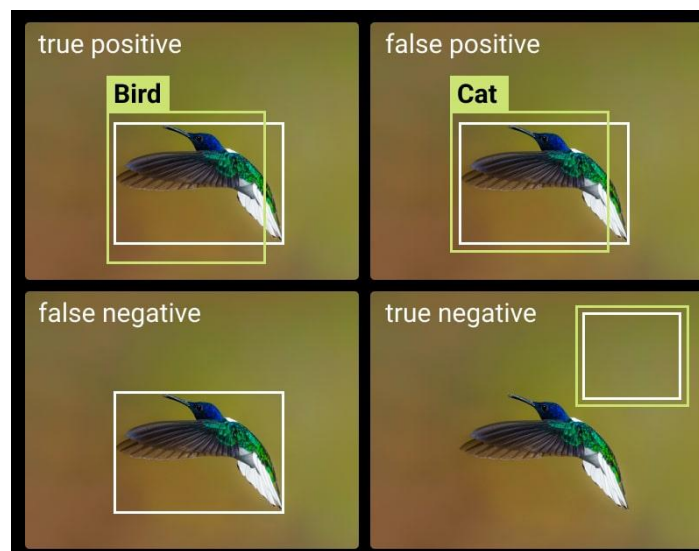
Wanneer er door VELU-PI geen wespachtige op de afbeelding wordt herkend, wordt deze afbeelding verwijderd op de Raspberry Pi, om opslag te besparen. Indien er wel een wespachtige wordt waargenomen dat geen Aziatische hoornaar is, volgt een wachtperiode van een minuut. Hierdoor wordt niet steeds dezelfde wespachtige door VELU-PI herkend, maar is de wachtperiode ook niet te lang om eventueel een Aziatische hoornaar, die in de tussentijd naast de wespachtige is geland, te missen.

4 RESULTATEN

4.1 Prestatiemetriecken

Om de prestatie van het beeldherkenningsmodel te evalueren worden er 4 metrieken gebruikt: precision, recall, F1-score en mAP. Alvorens we dieper ingaan in de berekening en formules van deze metrieken is het nodig om eerst een duiding te geven bij de concepten van true positive (TP), false positive (FP), true negative (TN) en false negative (FN).

Indien een insect in een afbeelding correct wordt geïdentificeerd, wordt dit aangeduid als een TP, is de voorspelling foutief dan is het een FP. Wanneer het model voorspelt dat er geen insect in de afbeelding is, en dit ook niet het geval is, is het een TN. Was er in dit laatste geval echter wel een insect in de afbeelding dan is het een FN. Figuur 16 verduidelijkt deze concepten.



Figuur 16. Visuele voorstelling van TP, FP, TN, FN (Kniazieva, 2023).

- **Precision:**

De precision meet de nauwkeurigheid van positieve voorspellingen van het model. Het is de ratio van het aantal TP's tegenover de som van de TP's en de FP's. Met andere woorden, de precision geeft aan in hoeverre correcte voorspellingen gebeuren tegenover het totale aantal positieve voorspellingen. (zie vergelijking 1)

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- **Recall:**

De recall meet de gevoeligheid van het model voor het detecteren van positieve gevallen. Het is de ratio van het aantal TP's tegenover de som van het aantal TP's en het aantal FN's. Met andere woorden, de recall geeft aan hoeveel van de werkelijke positieve gevallen correct zijn geïdentificeerd door het model. (zie vergelijking 2)

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- **F1:**

De F1-score is het harmonisch gemiddelde van de precision en recall, en hoge F1-score duidt dus op zowel een hoge precision als recall. Deze metriek geeft met andere woorden een gebalanceerde meting van de prestatie van het model en is daarom een veelgebruikte metriek. (zie vergelijking 3)

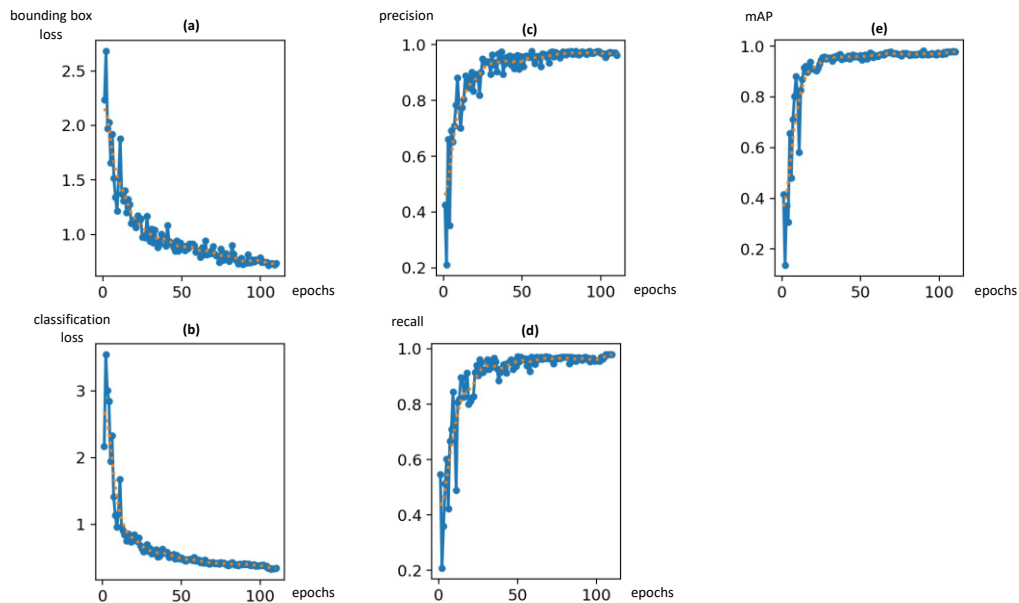
$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

- **mAP:**

De average precision (AP) komt overeen met de oppervlakte onder de precision-recall curve, wat een enkele waarde oplevert die zowel de precision als de recall van het model samenvat. De mean average precision (mAP) berekent het gemiddelde van de AP-waarde over alle klassen, waardoor deze metriek bijzonder geschikt is om modellen die meerdere klassen bevatten te evalueren.

4.2 Resultaat van training YOLOv8

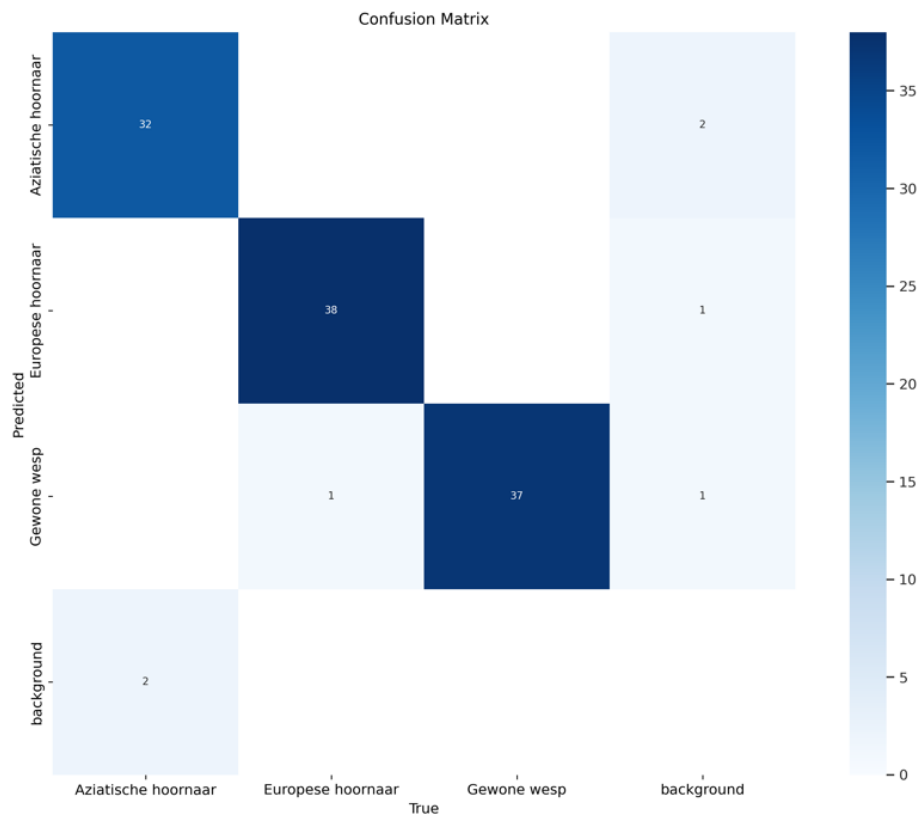
De trainingsresultaten van het YOLOv8-model voor het detecteren van de Aziatische hoornaar zijn veelbelovend. Na 110 epochs (trainingsstappen) heeft het model een mAP, precision en recall van respectievelijk 97.9%, 97.1% en 98.0% bereikt (zie figuur 17). Deze statistieken suggereren een uitstekende nauwkeurigheid van het model voor het identificeren en classificeren van de Aziatische hoornaar.



Figuur 17. Resultaat training YOLOv8: a) bounding box loss, b) classification loss, c) precision, d) recall en e) mAP

In figuur 17 geeft de 'bounding box loss' de grootte van de fout aan in de gedetecteerde bounding boxes, terwijl 'class loss' de grootte van de fout aangeeft in de gedetecteerde klassen. Naarmate het trainingsproces vorderde, namen beiden termen af, terwijl de precision, recall en mAP steeds groter werden. Het besluit om de training na 110 epochs te stoppen, lijkt gerechtvaardigd gezien de hoge prestatie-indicatoren.

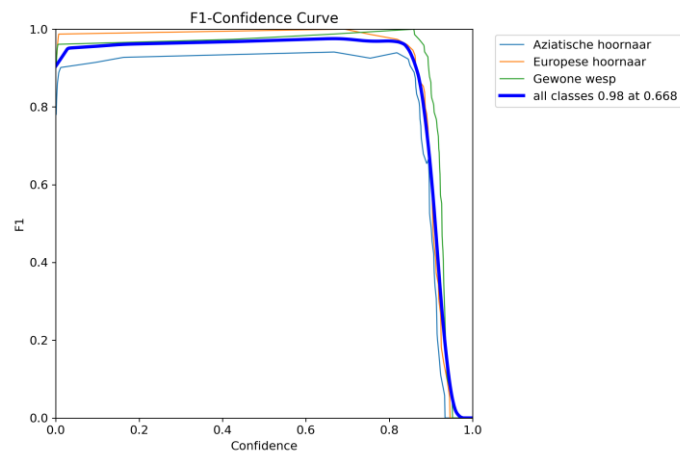
De confusion matrix in figuur 18 illustreert de relatie tussen de werkelijke klasse en de door het model voorspelde klasse, en biedt op die manier inzicht in de prestatie van het model. Hoge waarden op de diagonaal van de matrix duiden op een betere prestatie van het model, aangezien elke waarde op de diagonaal wijst op een correcte voorspelling van het model. Voor het opstellen van de confusion matrix de validatiedataset van 106 afbeeldingen gebruikt. Bovendien valt uit figuur 18 af te leiden dat het aantal TP's, FP's, FN's en TN's respectievelijk 107, 4, 3 en 0 bedraagt. Het is belangrijk op te merken dat object detectie modellen geen aandacht besteden aan het classificeren van regio's waar zich géén objecten bevinden, waardoor het aantal TN's vanzelfsprekend nul is. Dit aspect wordt daarom niet in beschouwing wordt genomen bij de evaluatie van object detectie modellen (Solawetz, 2024a).



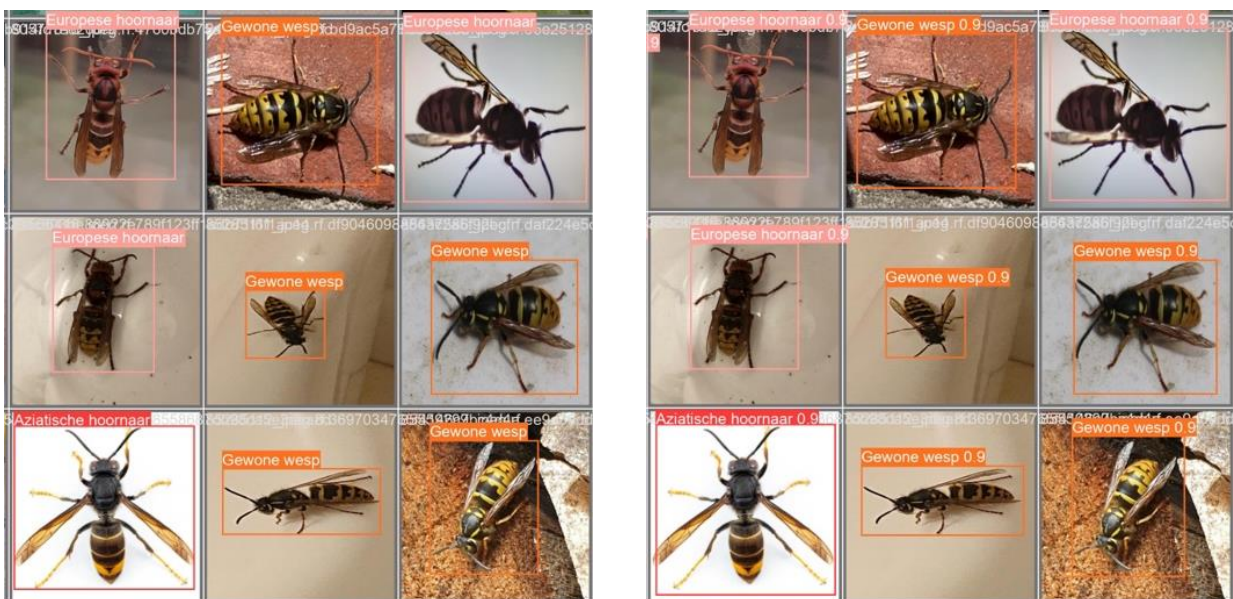
Figuur 18. Confusion matrix die werd bepaald door de 106 validatie afbeeldingen.

In figuur 19 wordt de F1-score weergegeven in relatie tot de zekerheid van het model. Deze grafiek biedt inzicht in de balans tussen de precision en recall van het model bij verschillende zekerheidsthresholds. Uit de grafiek blijkt dat de F1-score niet optimaal is bij een zeer lage of hoge zekerheidsthreshold. De F1-score is maximaal bij een zekerheidsthreshold van 0.668, de F1-score is hierbij 0.98. Hieruit volgt dat 0.668 de optimale threshold is om voorspellingen met het model uit te voeren, gezien de balans tussen de precision en recall hier optimaal is.

Figuur 20 is een visualisatie van de prestatie van het model op enkele testafbeeldingen. Links op de figuur is te zien welke bounding boxes en klassen er werden toegekend bij het annoteren van de beelden (ground truth), rechts zijn de voorspellingen van het YOLOv8 model. Het model levert een opvallend goede prestatie.



Figuur 19. F1-score in relatie tot de zekerheid van het YOLOv8 model, de F1-score is maximaal bij een threshold van 0.668.



Figuur 20. links: ground truth (testdataset), rechts: voorspellingen van YOLOv8 model.

Aangezien het YOLOv5-model werd gebruikt in het onderzoek van Jeon et al. (2023) en O'Shea-Wheller et al. (2024) was het relevant voor ons om, naast het trainen van YOLOv8, ook YOLOv5 te trainen op de dataset. Dit resulteerde in een mAP, precision en recall van respectievelijk 94,3%, 94,7% en 91,7%. Alhoewel deze scores nog steeds hoog zijn, liggen ze toch beduidend lager dan de scores van het YOLOv8-model.

Om het beeldherkenningsmodel in de praktijk te testen, hebben we een experiment uitgevoerd met een dode Aziatische hoornaar. Het betreffende exemplaar, afgebeeld op figuur 21, vertoonde door zijn dood enkele afwijkingen van de typische morfologie, zo waren bijvoorbeeld de poten ingetrokken. Deze afwijkingen zullen de herkenbaarheid van de hoornaar beïnvloeden. Ondanks de verminderde representativiteit, slaagde het model erin om het insect met 89% zekerheid correct te identificeren. Dit suggereert dat het model in realistische omstandigheden succesvol is, hoewel verdere veldtesten nodig zijn om dit met zekerheid vast te stellen.



Figuur 21. Eigen dode Aziatische hoornaar werkster met bijhorende bounding boxes en zekerheid van het model.

4.3 Werking finale script

In het finale script werkt alles zoals het bedoelt is. Bij langdurig testen blijkt echter dat er na vier uur een fout optreedt bij het uploaden van foto's en videos naar Dropbox, als gevolg van de acces token die vervalst. Pas na het invoeren in het script van een nieuw gegenereerde acces token, werkt het uploaden naar Dropbox opnieuw. Dit probleem zou kunnen opgelost worden door het gebruik van refresh tokens, die elk moment gebruikt kunnen worden om een nieuwe acces token te genereren, al vereist dit wat extra codeerwerk (How to get long lived acces (refresh) tokens for Dropbox developer apps | Codemzy, 2022).

5 DISCUSSIE

5.1 Verdere evaluatie van VELU-PI

De resultaten van het beeldherkenningsmodel zijn veelbelovend, met vergelijkbare prestaties op de prestatie-metrieken in vergelijking met analoge monitoringsystemen zoals VespAI. Voor een grondige evaluatie van het model in de praktijk, en om de prestatie met zekerheid te kunnen vergelijken met andere modellen, is het noodzakelijk dat veldproeven worden uitgevoerd. Het is mogelijk dat de dataset vooral afbeeldingen bevat waarop het insect gemakkelijk te herkennen is, waardoor de nauwkeurigheid van het model mogelijks wordt overschat. Tijdens de afronding van dit onderzoek, eind april 2024, waren slechts een beperkt aantal Aziatische hoornaars actief, waarvan hoofdzakelijk koninginnen. Dit resulteerde in een tekort aan testmateriaal voor uitgebreide veldtesten.

Bovendien moet ook de zelfvoorzienigheid van VELU-PI met behulp van de PiJuice en het zonnepaneel worden getest in het veld. Deze autonomie is essentieel om VELU-PI in afgelegen gebieden te laten functioneren. We schatten in dat dit geen probleem zal zijn, gezien enkel de bewegingssensor actief is zo lang er zich geen insect op de wiekpot bevindt. Bovendien kan een volledig opgeladen PiJuice-batterij meer dan 12u stroom voorzien bij gebruik van sensoren met lage energievereisten en deze batterij wordt overdag opgeladen door het zonnepaneel. De veldtesten in een volgend onderzoek kunnen een beter inzicht geven in het stroomgebruik (gekoppeld aan de rekenkracht) bij het uitvoeren van het script en of de PiJuice dit kan opvangen, zeker op momenten dat veel Aziatische hoornaars worden waargenomen.

5.2 Verbeteringen voor VELU-PI

Momenteel is het beeldherkenningsmodel getraind op afbeeldingen afkomstig van Waarnemingen.be. Idealiter zou het model verder geoptimaliseerd kunnen worden door het te trainen op basis van de afbeeldingen die werden vastgelegd bij het gebruik van de VELU-PI. Dit zou de prestatie van het systeem in het veld verhogen.

Het beeldherkenningsmodel is momenteel in staat om met hoge accuraatheid een Aziatische hoornaar te herkennen, waardoor we informatie krijgen wanneer een wiekpot bezocht wordt. Helaas is het nog niet mogelijk om een individuele hoornaar te onderscheiden van een andere hoornaar, waardoor het lastig is om vliegtijden te meten gezien we niet zeker zijn of het wel hetzelfde individu is dat terugkeert naar de wiekpot. Een mogelijke oplossing voor dit probleem kan worden geboden in toekomstig onderzoek door de hoornaars te merken. Meestal worden insecten gemerkt door een handmatig aangebracht merkplaatje op hun borststuk, waarop een uniek cijfer staat. Het beeldherkenningsmodel zou kunnen getraind worden om deze merkplaatjes uit te lezen, zodat de vliegtijden van de individuele hoornaars kunnen worden gekwantificeerd door het VELU-PI systeem. Deze methode stapt, door de nood aan het merken van de hoornaars, af van volledige automatisatie, maar biedt wel een oplossing voor het uitlezen van de vliegtijd, wat cruciaal is bij lokalisatie van nesten. Echter zou ook een systeem kunnen worden uitgedacht waarbij de hoornaars automatisch worden gemerkt bij het bezoeken van de wiekpot.

Naast de vliegtijd is ook de vliegrichting cruciaal voor de toepassing van triangulatie. Dit zou in een volgend onderzoek opgelost kunnen worden door de gemiddelde vliegrichtingen waarin een individu wegvliegt te berekenen. Geautomatiseerde video-analyse van de laatste seconden van video's, waarop een individuele hoornaar wordt herkend, zou hiervoor een oplossing kunnen bieden. Om te achterhalen in welke windrichting de hoornaar vliegt, zou er een kompas kunnen worden ingebouwd in de wiekpot, of zou het VELU-Pi systeem kunnen worden uitgebreid met een kompas-module (zoals CMPS14) zodat de vliegrichting automatisch kan gekoppeld worden aan een kaart. Een belangrijke kanttekening hierbij is dat de gemiddelde richting waarin een hoornaar wegvliegt van een wiekpot aan grote onzekerheid onderworpen is. Een hoornaar kan boven de wiekpot (uit het zicht van de camera) eerst nog wat cirkels draaien, om dan pas de juiste richting uit te vliegen naar het nest. Ook is die vliegrichting niet altijd de richting waar het nest gezocht moet worden. Een hoornaar gebruikt het landschap zoals een boom of een huis om zich te oriënteren waardoor deze niet in een rechte lijn vliegt (Speur nesten op | Vespa-Watch, z.d.). Praktijkervaring is dus vereist in welke mate deze fenomenen optreden, al zal dit ook van individu tot individu verschillen.

Zolang het probleem van de acces tokens (die na slechts vier uur vervallen) zich voordoet, is het bij langdurige monitoring beter de optie van het uploaden naar Dropbox achterwege te laten. Het is ook belangrijk om rekening te houden met de gratis 2 GB opslaglimiet van Dropbox. Regelmatig verwijderen van bestanden in Dropbox is dus een must, wat niet ideaal is. Er kan geopteerd worden om mits betaling een hogere limiet te verkrijgen, maar dan valt de betaalbaarheid van het systeem voor een deel in het water. De vraag kan dus gesteld worden als Dropbox wel de ideale cloudopslag vormt voor dit project. Het uittesten van andere cloud services en het gebruik van refresh tokens voor het automatisch genereren van nieuwe acces tokens, kan in een volgend onderzoek gebeuren.

Naast beeldgegevens kan er ook in een volgend onderzoek meer aan de slag gegaan worden met de gegevens van de BME680, die de temperatuur, luchtdruk en luchtvochtigheid meet. Zo zou met deze data een verband kunnen gezocht worden tussen de omgevingscondities en de bezoeken op de wiekpotten (of andere plaatsen, afhankelijk van de opstellingsplaats) van Aziatische hoornaars. De conclusies van Perrard et al. (2009) (die in paragraaf 2.1 reeds werden aangehaald), namelijk dat Aziatische hoornaars niet foerageren onder de 10°C en dat de piek in de middag valt, zouden kunnen heronderzocht worden. Deze informatie zou kunnen gebruikt worden om de VELU-PI automatisch uit te schakelen wanneer de omgevingscondities niet volstaan voor de activiteit van de Aziatische hoornaar, waardoor energie kan bespaard worden en valse waarnemingen worden vermeden.

6 CONCLUSIE

Het VELU-PI prototype, ontwikkeld in deze bachelorproef, is een veelbelovende vooruitgang in de identificatie en monitoring van de Aziatische hoornaar. Door middel van deep learning kan het systeem Aziatische hoornaars identificeren, waarbij foto's en video's worden vastgelegd en opgeslagen in zowel een lokale databank als in een cloud. Het systeem kan bovendien worden geconfigureerd om offline te werken, wat het geschikt maakt voor gebruik in diverse omstandigheden. De gebruiker moet bij offline gebruik fysiek naar de VELU-PI gaan om de vastgelegde beelden te bekijken, het systeem gebruikt namelijk enkel een internetverbinding om de foto's en video's te versturen naar de cloud. Hoewel een eerste evaluatie van het beeldherkenningsmodel wijst op een hoge nauwkeurigheid, zijn verdere veldtesten noodzakelijk om dit te bevestigen, evenals om de autonomie van het systeem te verifiëren.

Het VELU-PI systeem kan een belangrijke rol spelen bij het melden van Aziatische hoornaars aan Vespa-Watch, waarbij de verzamelde foto en video als bewijs kunnen dienen. Zij kunnen vervolgens ter plaatsen komen om het nest op te sporen en te verwijderen. Om echter volledig gebruik te kunnen maken van het potentieel van het systeem, zijn verdere verbeteringen nodig. Het systeem moet worden verfijnd om individuele hoornaars te kunnen onderscheiden, zodat vliegtijden en vliegrichtingen automatisch kunnen worden gekwantificeerd. Dit zou triangulatie vergemakkelijken, dat het proces van nestdetectie en -verwijdering aanzienlijk zou versnellen.

De impact van het VELU-PI systeem is afhankelijk van de breedte van de implementatie ervan. Het is noodzakelijk dat grote aantallen van dit systeem worden verspreid over heel Vlaanderen, zodat een uitgebreid beeld kan worden verkregen van de verspreiding van deze invasieve soort, wat op zijn beurt kan helpen bij een doeltreffende bestrijding. Op basis van deze bachelorproef kunnen burgers en imkers zelf de benodigde materialen aankopen en het systeem nabouwen. Om de toegankelijkheid van het systeem te vergroten, zou ook een zelfbouwpakket kunnen worden ontwikkeld. Dit pakket zou alle benodigde materialen bevatten, evenals een voorgeprogrammeerde SD-kaart en een handleiding gebaseerd op deze scriptie. Deze handleiding zou enkel nodig zijn om de hardware met bedrading te koppelen aan de Raspberry Pi. Eventueel zouden er verschillende versies van het pakket kunnen worden aangeboden om de kosten te verlagen, gezien de PiJuice HAT, de 12000mAh batterij en het zonnepaneel de grootste kost zijn en deze niet in alle gevallen noodzakelijk is. Dit zou de totale kost van ±€300 kunnen verlagen tot slechts ±€125. Het pakket zou kunnen verdeeld worden via gespecialiseerde websites zoals bijenhof.be, of gratis worden

verspreid via gemeenten, vergelijkbaar met de huidige distributie van selectieve vallen. Op deze manier kan een brede acceptatie en implementatie worden bevorderd.

In conclusie heeft dit onderzoek belangrijke inzichten opgeleverd met betrekking tot de problematiek van de Aziatische hoornaar, en introduceert het daarom een concrete bestrijdingsstrategie. Het voorgestelde prototype zou kunnen bijdragen aan de bescherming van de bijenpopulaties en de gehele biodiversiteit. Verder onderzoek en implementatie van dit prototype zijn essentieel om de bedreiging effectief aan te pakken.

7 DUURZAAMHEIDSREFLECTIE

7.1 Kadering

De Aziatische hoornaar (*Vespa Velutina*) begeeft zich al vanaf 2011 in België en sinds de afgelopen jaren neemt het aantal nesten van de invasieve wespensoor enorm toe (Aantal nesten Aziatische hoornaars in Vlaanderen verviervoudigd in 2023 | Imkers Neteland, 2024). De soort vormt niet alleen een bedreiging voor de honingbijen, maar ook voor de rest van het ecosysteem. De imkerverenigingen doen massaal hun best om de nesten op te sporen en te verdelgen, maar deze aanpak volstaat helaas niet en is arbeidsintensief. Daarom stelt deze bachelorproef het VELU-PI prototype voor, een open-source automatisch monitoringssysteem voor de identificatie en bestrijding van de Aziatische hoornaar.

Het prototype is gebaseerd op de Raspberry Pi, een kleine computer met veel mogelijkheden. Deze Raspberry Pi is uitgerust met een camera, een bewegingssensor en een sensor voor de omgevingscondities. Onder de camera staat een lokmiddel die de hoornaar aantrekt. De Raspberry Pi kan op basis van beeldherkenning de Aziatische hoornaars waarnemen, waarna een foto en video van het insect genomen wordt. Het VELU-PI prototype kan in de toekomst evolueren naar een systeem dat bovendien de vliegrichting van de hoornaar bepaalt. Met deze vliegrichting kan de locatie van het nest bepaald worden, door ook de vliegtijd in rekening te brengen. De vuistregel geldt dat het nest 100 meter van het lokmiddel verwijderd is, per minuut vliegtijd van de hoornaar (Van Ransbeeck et al., 2023).

De bachelorproef heeft als doel het monitoringssysteem massaal te verspreiden zodat de kracht van Citizen Science kan worden ingeschakeld. Daarom staat bij de ontwikkeling van VELU-PI toegankelijkheid centraal, zo kan deze bachelorproef gebruikt worden als een handleiding en staat alle gebruikte code beschikbaar online alsook een lijst van het gebruikte materiaal.

7.2 Reflectievraag 1: Relatie tot duurzame ontwikkeling

Dit project draagt bij aan het behoud van biodiversiteit binnen het ecosysteem en heeft dus een duidelijke betrekking op SDG 15 (leven op het land), gezien deze doelstelling zich voornamelijk richt op het herstellen en beschermen van biodiversiteit in ecosystemen. De problematiek van de Aziatische hoornaar sluit hier nauw bij aan, want zoals onder andere Clavero et al. (2009) ondervonden, is de intrede van invasieve uitheemse

soorten een belangrijke bron voor het verlies aan biodiversiteit. Moncau et al. (2013) verwijst naar het feit dat de inheemse honingbij en andere prooien zich nog onvoldoende hebben aangepast aan de hoornaar in dit vroege stadium van co-evolutie, waardoor de predator meer impact verricht op de lokale prooien dan de inheemse predatorsoorten.

Directe actie ondernemen is noodzakelijk. Target 15.8 (Prevent Invasive Alien Species On Land And In Water Ecosystems) van SDG 15 luidt: “maatregelen invoeren om de invoering van invasieve uitheemse soorten in land- en waterecosystemen te beperken en hun impact op aanzienlijke wijze te beperken”. Dit prototype kan, na verdere ontwikkeling, bijdragen aan deze target.

Naast SDG 15 draagt VELU-PI onrechtstreeks bij aan SDG 2 (geen honger), aangezien het opsporen en bestrijden van de Aziatische hoornaars bescherming biedt voor insecten die een belangrijke functie hebben bij bestuiving. Onder deze SDG valt onder andere het bereiken van voedselzekerheid en duurzame, veerkrachtige landbouwpraktijken die de productiviteit en productie verhogen. Het is algemeen geweten dat de bestuivers cruciaal zijn voor het bevruchten van essentiële landbouwgewassen. Meer dan 75% van de 115 belangrijkste gewassoorten ter wereld zijn afhankelijk van, of hebben op zijn minst voordeel aan, dierlijke bestuiving. Op gewichtsbasis vertaalt dit zich naar $\pm 35\%$ van de wereldwijde productie van gewassen. Het zijn vooral bestuivingsafhankelijke gewassen zoals fruit die bijdragen aan een gezonde menselijke voeding door bijzonder hoge hoeveelheden essentiële voedingsstoffen zoals vitamines, antioxidanten en vezels te leveren (Klein et al., 2006).

Klatt et al. (2014) tonen met hun onderzoek de gevolgen van bijenbestuiving voor de kwaliteit, kwantiteit en marktwaarde van fruit. Bijenbestuiving bevordert niet alleen de productiviteit, maar ook de handelskwaliteiten, de houdbaarheid en de vermindering van fruitverlies. Deze convergentie van ecologische diensten en economische voordelen benadrukt de bedreiging van de duurzaamheid en winstgevendheid van de fruitteelt door afnemende bijenpopulaties.

7.3 Reflectievraag 2: Relatie tot het maatschappelijke debat

De opkomst van de Aziatische hoornaar vormt een veel besproken onderwerp, zowel op nationaal als internationaal niveau. De urgentie om effectieve strategieën te vinden voor de bestrijding van deze exoot is hoog, aangezien steeds meer imkers kampen met aanvallen van hoornaars op hun bijenkasten. René De Backer van het Vlaams Bijeninstituut stelt vast dat de wintersterfte van bijen aanzienlijk is toegenomen, van 10 naar 27%, in vergelijking met voorgaande decennia (Dewever, 2024).

Aanvankelijk uitte het Vlaamse Bijeninstituut (VBI) bedenkingen over de beperkte bijdrage van de Vlaamse overheid bij de bestrijding van de Aziatische hoornaar, bevestigt De Backer in de Standaard (Debeuckelaere, 2023). Meer specifiek wordt het departement Landbouw en Visserij ervan beschuldigd dat het zijn verantwoordelijkheid ontliet, waardoor de imkergemeenschap op weinig financiële steun kon rekenen voor verdelgingskosten. Gelukkig beginnen steeds meer lokale besturen subsidies toe te kennen voor de vergoeding van verdelgingskosten en delen ze vallen uit (Auman, 2024), en begint ook de Vlaamse overheid

in actie te schieten. De Vlaamse overheid erkent steeds meer de ernst van de situatie en neemt samen met organisaties zoals het Agentschap Natuur en Bos het initiatief voor de coördinatie van een beheerplan. Zo is het Agentschap Natuur en Bos momenteel aan het werken zijn aan een beheerplan dat binnenkort goedgekeurd zou worden (Deweever, 2024). De bestrijding van de Aziatische hoornaar in Vlaanderen gebeurt momenteel in samenwerking tussen het VBI, het departement Landbouw en Visserij en het Agentschap Natuur en Bos.

In een recente vergadering van het Vlaams Parlement werd er kritiek geuit op het nieuwe beheerplan voor de bestrijding van de Aziatische hoornaar, dat gepland staat voor uitvoering in de zomer. (Commissieleden Bespreken Bestrijding Aziatische Hoornaar | Vlaams Parlement, 2023). Het VBI benadrukt dat er in het plan een gebrek is aan coördinatie en informatie, en dat het te laat komt. Het VBI vreest dat het nieuwe beheerplan zal zorgen voor de minimalisatie van de betrokkenheid van dit instituut bij de bestrijding. Jo Brouns (CD&V), minister van Landbouw, verwijst naar minister Demir (N-VA) en haar administratie (het Agentschap voor Natuur en Bos) voor de aanpak van invasieve soorten. Hij stelt dat het departement Landbouw en Visserij zich voornamelijk zal focussen op het de imkerij, die binnen deze globale impact een groot deel van de schade ondervindt. Daarnaast weerlegt hij de onzekerheid van het VBI door te stellen dat de organisatie weldegelijk een rol zal spelen in het nieuwe strategische plan, hoewel het nog onduidelijk is of zij de coördinatie op zich zullen mogen nemen. Er is ook kritiek op het wegvallen van subsidies voor het verwijderen van nesten, die nu door inwoners of lokale besturen moet worden gefinancierd terwijl dit voordien federaal gesubsidieerd werd. Tot slot betwijfelt Chris Steenwegen (Groen) de timing van het nieuwe beheerplan, dat gepland staat voor de zomer, terwijl actie volgens hem aan het begin van het seizoen moet worden genomen. In het begin van het seizoen zijn er enkel primaire nesten, die makkelijker kunnen verwijderd worden dan de secundaire nesten die later in het seizoen voorkomen.

In het kader van duurzame ontwikkeling en het bredere maatschappelijke debat is de bestrijding van de Aziatische hoornaar van aanzienlijk belang. Het aantal nesten van deze soort in Vlaanderen vertoont een exponentiële groei, wat de urgentie van een effectief beheerplan benadrukt. Het beschermen van bestuivers is cruciaal voor het behoud van biodiversiteit en heeft ook indirecte gevolgen voor de voedselvoorziening. Daarom is algemene bezorgdheid en massale betrokkenheid bij de aanpak van deze invasieve soort gerechtvaardigd.

8 REFERENTIES

- Aantal nesten Aziatische hoornaars in Vlaanderen vervijfvoudigd in 2023. (2024). Imkers Neteland. [Online] Beschikbaar: <https://www.imkersneteland.be/aantal-nesten-aziatische-hoornaars-in-vlaanderen/#:~:text=Meer%20dan%206000%20nesten%20van,nog%20een%20stuk%20hoger%20liggen>. [Geraadpleegd op 3/03/2024]
- AI Voor het Meten en Monitoren van Biodiversiteit en Fauna. (2024). Wageningen Research & Development. [Online] Beschikbaar: <https://www.wur.nl/nl/en/onderzoek-resultaten/onderzoeksinstituten/plant-research/show-wpr/ai-voor-het-meten-en-monitoren-van-biodiversiteit-en-fauna.htm>. [Geraadpleegd op 9/04/2024]
- Auman, J. (2024). Groen en CD&V vragen actieplan om Aziatische hoornaar te bestrijden: “Lokvallen en terugbetaling kosten nestverwijdering. *Gazet van Antwerpen*. [Online] Beschikbaar: https://www.gva.be/cnt/dmf20240227_93589849. [Geraadpleegd op 4/04/2024]
- Aziatische hoornaar. (2024). Imkers Neteland. [Online] Beschikbaar: <https://www.imkersneteland.be/aziatische-hoornaar/>. [Geraadpleegd op 16/02/2024]
- Aziatische hoornaar (*Vespa velutina*). (2023). Vlaams Bijeninstituut. [Online] Beschikbaar: <https://vlaamsbijeninstituut.be/expertisecentra/aziatische-hoornaar/>. [Geraadpleegd op 26/02/2024]
- Aziatische hoornaar. (z.d.). Waarneming.be. [Online] Beschikbaar: <https://waarnemingen.be/species/8807/photos/?>. [Geraadpleegd op 20/02/2024]
- Bescherming van wilde bestuivers in de EU — de initiatieven van de Commissie hebben geen vruchten afgeworpen. (2020). Europese rekenkamer. [Online] Beschikbaar: <https://op.europa.eu/webpub/eca/special-reports/pollinators-15-2020/nl/index.html>. [Geraadpleegd op 29/04/2024]
- Bjerge, K., Mann, H. M. R., & Høye, T. T. (2021). Real-time insect tracking and monitoring with computer vision and deep learning. *Remote Sensing in Ecology And Conservation*, 8, 315–327.
- Blob and Text Data types. (z.d.). MariaDB. [Online] Beschikbaar: <https://mariadb.com/kb/en/blob-and-text-data-types/>. [Geraadpleegd op 2/04/2024]
- Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology And Evolution*, 10, 1632–1644.
- Clavero, M., Brotons, L., Pons, P., & Sol, D. (2009). Prominent role of invasive species in avian biodiversity loss. *Biological Conservation*, 142(10), 2043–2049.
- Commissieleden bespreken bestrijding Aziatische hoornaar. (2023). Vlaams Parlement. [Online] Beschikbaar : <https://www.vlaamsparlement.be/nl/actueel/nieuws-uit-het-vlaams-parlement/commissieleden-bespreken-bestrijding-aziatische-hoornaar>. [Geraadpleegd op 27/04/2024]
- Configuring MariaDB for Remote Client Access. (z.d.). MariaDB. [Online] Beschikbaar: <https://mariadb.com/kb/en/configuring-mariadb-for-remote-client-access/>. [Geraadpleegd op 2/04/2024]

- Converting BLOB file data to image from sql database. (2016). forums.raspberrypi. [Online] Beschikbaar: <https://forums.raspberrypi.com/viewtopic.php?t=149142>. [Geraadpleegd op 2/04/2024]
- De Boer, E. (2020). Basis Raspberry Pi project: camera. [Online] Beschikbaar: <https://elektronicavoorjou.nl/basis-raspberry-pi-project-camera/>. [Geraadpleegd op 2/04/2024]
- Debeuckelaere, H. (2023). Versnipperde aanpak geeft Aziatische hoornaar vleugels. *De Standaard*. [Online] Beschikbaar: https://www.standaard.be/cnt/dmf20230521_97479415?&articlehash=kGo06m0XI2JCGdcYPij%2F5qP2FHeznwwWacpVLrjLDxrSKf34VXSbKWifUxuvY022kmLnAPRfWJrRI40laBsPPD0ZqXgE3Y7F3rg29c2yI%2Bg5cuDrESwq1wQIHvH9dhg%2FEtUrnnyH8KbUqE9E8RUD90Qxxjghave8nlch%2F4mFIPjOGkpAac%2Bg9%2F6%2Bz6kd8ynjW8TWMcdZT7pH78yRbFGQKpm2Yi7Rc7VY0EZ8cq6FYiPPrNUfz0voIJmqKXHZTTvD4v2N9G95HbX%2BGk3%2BonqksRsclQFkjt3KzIEv5opcGbFALWypW0mSL8NRQP0gRrS7XZgekb8M2%2Br4S0z14ajQ%3D%3D. [Geraadpleegd op 4/04/2024]
- Deweever, I. (2024). Vijf keer meer nesten Aziatische hoornaars in een jaar tijd: “Kost verdelging loopt te hoog op”. *Gazet van Antwerpen*. [Online] Beschikbaar: https://www.gva.be/cnt/dmf20240301_97636209?&articlehash=YzK0tcz4XG%2BcusUf839drbBZFfL0P4xjvy5RcRvCEYY8H%2BFTYHSwnIglvrMc9hzDxuDMPXx%2FxrGOg7dN4d0w95cwXB56lwEGs4%2F16zrjObcLx%2BDdbBrIHxnsKMXA6jScXmL1LJksimdCFWuDe2Ti4D7vgnw7cxFxWk5kvres9ET0xhkoGd%2By%2Fk%2BDcqOF8NEylsudLmeXoJfp8jMYSj1dbiw01B6PwBkUivKPoUhBOyPY%2B3pNBdFzs%2Bx9upzoRbLC0ujmYygkobiV5JZOEfafaqY7%2BQNm%2FVUK8zFVxEHv103pAVuYc0Rk4Aodni6KcyLqAR18ZaCQCwOGnMviuSIClGw%3D%3D. [Geraadpleegd op 3/04/2024]
- Diéguez-Antón, A., Escuredo, O., Seijo, M. C., & Rodríguez-Flores, M. S. (2022). Embryo, Relocation and Secondary Nests of the Invasive Species *Vespa velutina* in Galicia (NW Spain). *Animals (Basel)*, 12(20), 2781.
- Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [software]. <https://roboflow.com>
- Eerste meting van het effect van de Aziatische hoornaar op bijenvolken in Vlaanderen. (2023). Vlaams Bijeninstituut. [Online] Beschikbaar: <https://vlaamsbijeninstituut.be/eerste-meting-van-het-effect-van-de-aziatische-hoornaar-op-bijenvolken-in-vlaanderen/>. [Geraadpleegd op 11/12/2023]
- Europese hoornaar. (z.d.). Waarneming.be. [Online] Beschikbaar: <https://waarnemingen.be/species/1672/photos/?>. [Geraadpleegd op 20/02/2024]
- Fedele, E., Gervasini, E., Cardoso, A.C., La Notte, A., Vallecillo, S., Tsiamis, K., & Maes, J. (2019). Invasive alien species impact on ecosystem services : Asian hornet (*Vespa velutina nigrithorax*) case study. *Publications Office Of The EU*.
- Germanov, A. (2023). How to Detect Objects in Images Using the YOLOv8 Neural Network. freeCodeCamp.org. [Online] Beschikbaar: <https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8>. [Geraadpleegd op 6/04/2024]
- Getting started with BME680 Breakout. (z.d.). Pimoroni. [Online] Beschikbaar: <https://learn.pimoroni.com/article/getting-started-with-bme680-breakout>. [Geraadpleegd op 2/04/2024]

- Gewone wesp. (z.d.). Waarneming.be. [Online] Beschikbaar: <https://waarnemingen.be/species/1709/photos/?>. [Geraadpleegd op 20/02/2024]
- Herrera, C., Williams, M. R. J., Encarnação, J., Roura-Pascual, N., Faulhaber, B., Jurado-Rivera, J. A., & Leza, M. (2022). Automated detection of the yellow-legged hornet (*Vespa velutina*) using an optical sensor with machine learning. *Pest Management Science*, 79(3), 1225–1233.
- Housen, R. (2023). Imkers uit Stevoort sporen op eigen houtje Aziatische hoornaars op met lokpotten. *vrtnws.be*. [Online] Beschikbaar: <https://www.vrt.be/vrtnws/nl/2023/10/31/imkers-op-zoek-naar-hoornaars/>. [Geraadpleegd op 2/04/2024]
- How to enable Remote acces to your MariaDB/MySQL database on Ubuntu Bionic or MariaDB < v10.6. (2022). Webdock. [Online] Beschikbaar: <https://webdock.io/en/docs/how-guides/database-guides/how-enable-remote-access-your-mariadbmysql-database>. [Geraadpleegd op 2/04/2024]
- How to get long lived acces (refresh) tokens for Dropbox developer apps. (2022). Codemzy. [Online] Beschikbaar: <https://www.codemzy.com/blog/dropbox-long-lived-access-refresh-token>. [Geraadpleegd op 30/04/2024]
- How to Train YOLOv8 Object Detection on a Custom Dataset. (z.d.). Google Colaboratory. [Online] Beschikbaar: <https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb?authuser=1#scrollTo=QDZUt2WUYdEy>. [Geraadpleegd op 6/04/2024]
- Høye, T. T., Årje, J., Bjerger, K., Hansen, O. I., Iosifidis, A., Leese, F., Mann, H. M. R., Meissner, K., Melvad, C., & Raitoharju, J. (2021). Deep learning and computer vision will transform entomology. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 118(2).
- Ijzermans, R. (2016). HC-SR501 aansluiten op een Raspberry Pi. [Online] Beschikbaar: <https://raspberrytips.nl/hc-sr501-bewegingssensor-pir-raspberry-pi/>. [Geraadpleegd op 2/04/2024]
- Jeon, M., Jeong, Y., Lee, J., Sui, Y., Kim, S., Kim, D., Kim, K., Lee, S., Lee, C., & Choi, I. (2023). Deep Learning-Based Portable Image Analysis System for Real-Time Detection of *Vespa velutina*. *Applied Sciences*, 13, 7414.
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [software]. <https://github.com/ultralytics/ultralytics>
- Kasinathan, T., Singaraju, D., & Reddy, U. S. (2021). Insect classification and detection in field crops using modern machine learning techniques. *Information Processing in Agriculture*, 8(3), 446–457.
- Kennedy, P. J., Ford, S. M., Poidatz, J., Thiéry, D., & Osborne, J. L. (2018). Searching for nests of the invasive Asian hornet (*Vespa velutina*) using radio-telemetry. *Communications biology*, 1, 88.
- Kim, S., Ju, C., Kim, J., & Son, H. I. (2019). A tracking method for the invasive asian hornet: A brief review and experiments. *IEEE Access*, 7, 176998-177008.
- Klatt, B. K., Holzschuh, A., Westphal, C., Clough, Y., Smit, I., Pawelzik, E., & Tscharrntke, T. (2014). Bee pollination improves crop quality, shelf life and commercial value. *Proceedings of the Royal Society B: Biological Sciences*, 281(1775), 20132440.

- Klein, A., Vaissière, B., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., & Tscharntke, T. (2006). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*, 274(1608), 303–313.
- Kniazieva, Y. (2023). How to Use Object Detection Metrics for Computer Vision Excellence?. Label Your Data. [Online] Beschikbaar: <https://labeledyourdata.com/articles/object-detection-metricsa>. [Geraadpleegd op 7/04/2024]
- Koch, K. A. (z.d.). iPhone Document scanner. [Online] Beschikbaar: <https://www.ponoko.com/free-laser-cutting-files-templates>. [Geraadpleegd op 27/04/2024]
- Lioy, S., Manino, A., Porporato, M., Laurino, D., Romano, A., Capello, M., & Bertolino, S. (2019). Establishing surveillance areas for tackling the invasion of *Vespa velutina* in outbreaks and over the border of its expanding range. *NeoBiota*, 46, 51-69.
- Lioy, S., Laurino, D., Maggiora, R., Milanesio, D., Sacconi, M., Mazzoglio, P. J., Manino, A. & Porporato, M. (2021). Tracking the invasive hornet *Vespa velutina* in complex environments by means of a harmonic radar. *Scientific Reports*, 11(1), 12143.
- Megatommy. (2021). bme680-collector. [Online] Beschikbaar: <https://github.com/megatommy/bme680-collector/tree/main>. [Geraadpleegd op 2/04/2024]
- Moens, L. (2024). Gemeente verdeelt honderd vallen om Aziatische hoornaars te vangen: “Grote bedreiging voor onze bijen”. *Het Nieuwsblad Mobile*. [Online] Beschikbaar: https://m.nieuwsblad.be/cnt/dmf20240318_95821411. [Geraadpleegd op 4/04/2024]
- Monceau, K., Arca, M., Leprêtre, L., Mougél, F., Bonnard, O., Silvain, J., Maher, N., Arnold, G., & Thiéry, D. (2013a). Native Prey and Invasive Predator Patterns of Foraging Activity: The Case of the Yellow-Legged Hornet Predation at European Honeybee Hives. *PloS One*, 8(6), e66492.
- Monceau, K., Bonnard, O., & Thiéry, D. (2013b). *Vespa velutina*: a new invasive predator of honeybees in Europe. *Journal Of Pest Science*, 87(1), 1–16.
- Monceau, K., Tourat, A., Arca, M., Bonnard, O., Arnold, G., & Thiéry, D. (2017). Daily and Seasonal Extranidal Behaviour Variations in the Invasive Yellow-Legged Hornet, *Vespa velutina* Lepeletier (Hymenoptera: Vespidae). *Journal Of Insect Behavior*, 30(2), 220–230.
- Meer nesten van de Aziatische hoornaar ondanks koude lente en natte zomer. (z.d.). Natuurpunt. [Online] Beschikbaar: <https://www.natuurpunt.be/nieuws/meer-nesten-van-de-aziatische-hoornaar-ondanks-koude-lente-en-natte-zomer>. [Geraadpleegd op 25/04/2024]
- Nave, A., Godinho, J., Fernandes, J., Arias, A. I. G., Golpe, M. A. F., & Branco, M. (2024). *Vespa velutina*: a menace for Western Iberian fruit production. *Cogent Food & Agriculture*, 10(1).
- NeuralNine. (2023). Dropbox Automation for Backups in Python. [Online] Beschikbaar: https://youtu.be/-jPLlgms18I?si=5_5txaJxjht1ehOK. [Geraadpleegd op 2/04/2024]
- O’Shea-Wheller, T. A., Corbett, A. C., Osborne, J. L., Recker, M., & Kennedy, P. (2024). VespAI: a deep learning-based system for the detection of invasive hornets. *Communications Biology*, 7.
- Padrón, E. (2023). Camera setup on Raspberry Pi 4 2023. [Online] Beschikbaar:

<https://fulldataalchemist.medium.com/camera-setup-on-raspberry-pi-4-2023-46d02305afa4>.
[Geraadpleegd op 22/04/2024]

- Pérez-Méndez, N., Andersson, G. K., Réquier, F., Hipólito, J., Aizen, M. A., Morales, C., García, N., Gennari, G. P., & Garibaldi, L. A. (2020). The economic cost of losing native pollinator species for orchard production. *Journal Of Applied Ecology*, 57(3), 599–608.
- Perrard, A., Haxaire, J., Rortais, A., & Villemant, C. (2009). Observations on the colony activity of the Asian hornet *Vespa velutina* Lepeletier 1836 (Hymenoptera: Vespidae: Vespinae) in France. *Annales de La Société Entomologique de France*, 45(1), 119–127.
- PiSupply. (z.d.). GitHub - PiSupply/PiJuice: Resources for PiJuice HAT for Raspberry Pi - use your Pi Anywhere. [Online] Beschikbaar: <https://github.com/PiSupply/PiJuice>. [Geraadpleegd op 10/03/2024]
- PIJuice Battery Discharge Time Calculator (z.d.) PI Supply. [Online] Beschikbaar: <https://learn.pi-supply.com/battery-levels/>. [Geraadpleegd op 2/05/2024]
- Poidatz, J., Monceau, K., Bonnard, O., & Thiéry, D. (2018). Activity rhythm and action range of workers of the invasive hornet predator of honeybees *Vespa velutina*, measured by radio frequency identification tags. *Ecology And Evolution*, 8(15), 7588–7598.
- Preprocess Images. (z.d.). Roboflow Docs. [Online] Beschikbaar: <https://docs.roboflow.com/datasets/image-preprocessing>. [Geraadpleegd op 6/04/2024]
- Raspberry Pi documentation. (z.d.). Getting started with your Raspberry Pi. [raspberrypi.com](https://www.raspberrypi.com). [Online] Beschikbaar: <https://www.raspberrypi.com/documentation/computers/getting-started.html#install-an-operating-system>. [Geraadpleegd op 2/04/2024]
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- Requier, F., Rome, Q., Chiron, G., Decante, D., Marion, S., Ménard, M., Müller, F., Villemant, C., & Henry, M. (2018). Predation of the invasive Asian hornet affects foraging activity and survival probability of honey bees in Western Europe. *Journal Of Pest Science*, 92(2), 567– 578.
- Robinet, C., Suppo, C., & Darrouzet, É. (2016). Rapid spread of the invasive yellow-legged hornet in France: the role of human-mediated dispersal and the effects of control measures. *Journal Of Applied Ecology*, 54(1), 205–215. <https://doi.org/10.1111/1365-2664.12724>
- Robotics Back-End. (2021). How to make a Raspberry Pi program start on boot (systemd). [Online] Beschikbaar: https://youtu.be/DUGZC-tNm2w?si=YAf_5H7Vbcl6QyFZ. [Geraadpleegd op 30/04/2024]
- Rojas-Nossa, S. V., & Calviño-Cancela, M. (2020). The invasive hornet *Vespa velutina* affects pollination of a wild plant through changes in abundance and behaviour of floral visitors. *Biological Invasions*, 22(8), 2609–2618.
- Rome, Q., Perrard, A., Müller, F., Fontaine, C., Quilès, A., Zuccon, D., & Villemant, C. (2021). Not just honeybees: predatory habits of *Vespa velutina* (Hymenoptera: Vespidae) in France. *Annales de La Société Entomologique de France*, 57(1), 1–11.

- Saga, T., Okuno, M., Loope, K. J., Tsuchida, K., Ohbayashi, K., Shimada, M., & Okada, Y. (2020). Polyandry and paternity affect disease resistance in eusocial wasps. *Behavioral Ecology*, 31(5), 1172–1179.
- Santos, R., & Santos S. (z.d.). Raspberry Pi: Install Apache + MySQL + PHP (LAMP Server). [Online] Beschikbaar: https://randomnerdtutorials.com/raspberry-pi-apache-mysql-php-lamp-server/?fbclid=IwAR3K58IbRFimxN9qon7eDKxKimhSphCkqkSzFIH2GGJIVlpYY_4dUCi0XZA. [Geraadpleegd op 11/03/2024]
- Schoonvaere, K., Laget, D., Adriaens, T., Desmet, P., Villers, V., & de Graaf, D. (2020). Vespa-Watch: Invasiemonitoring van de Aziatische hoornaar met hobbyimkers en het publiek. Eindrapport in het kader van de oproep Citizen Science van het Departement Economie, Wetenschap en Innovatie (EWI) van december 2017. Honeybee Valley en Instituut voor Natuur- en Bosonderzoek, Brussel. [Online] Beschikbaar: https://purews.inbo.be/ws/portalfiles/portal/19229441/Schoonvaere_et_al_2020_VespaWatchInvasiemonitoringVanDeAziatischeHoornaarMetHobbyimkersEnHetPubliek.pdf. [Geraadpleegd op 16/02/2024]
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal Of Big Data*, 6.
- Solawetz, J. (2023). What is YOLOv8? The Ultimate Guide. [Online] Beschikbaar: <https://blog.roboflow.com/whats-new-in-yolov8/>. [Geraadpleegd op 9/04/2024]
- Solawetz, J. (2024a). What is Mean Average Precision (mAP) in Object Detection?. [Online] Beschikbaar: <https://blog.roboflow.com/mean-average-precision/#:~:text=guess%20correctly%3F%22-,Recall,->. [Geraadpleegd op 9/04/2024]
- Solawetz, J. (2024b). Train, Validation, Test Split for Machine Learning. Roboflow Blog. [Online] Beschikbaar: <https://blog.roboflow.com/train-test-split/>. [Geraadpleegd op 6/04/2024]
- Soviany, P., & Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 209–214.
- Speur nesten op. (z.d.). Vespa-Watch. [Online] Beschikbaar: <https://vespawatch.be/get-involved/>. [Geraadpleegd op 27/04/2024]
- Spinner, A. (2021). Raspberry Pi: Web Server Setup (Apache, PHP, MySQL, PHPMyAdmin). [Online] Beschikbaar: <https://youtu.be/OBbTkBy2bII?si=8HT-oRpkt8Lc8oQG>. [Geraadpleegd op 11/03/2024]
- Spring trapping: koninginnen Aziatische hoornaar vangen in het voorjaar. (2023). Imkers Neteland. [Online] Beschikbaar: <https://www.imkersneteland.be/spring-trapping-koninginnen-aziatische-hoornaar-vangen-in-het-voorjaar/>. [Geraadpleegd op 5/04/2024]
- Tan, K., Radloff, S. E., Li, J. J., Hepburn, H. R., Yang, M., Zhang, L. J., & Neumann, P. (2007). Bee-hawking by the wasp, *Vespa velutina*, on the honeybees *Apis cerana* and *A. mellifera*. *Naturwissenschaften*, 94(6), 469–472.
- Teoh, J. W. (2023). Revolutionizing Object Annotation For Computer Vision: A Comprehensive Guide to Roboflow. *Medium*. [Online] Beschikbaar: <https://medium.com/@jengweiteoh/revolutionizing->

object-annotation-for-computer-vision-a-comprehensive-guide-to-roboflow-ad0dfb833fc4.
[Geraadpleegd op 6/04/2024]

Turchi, L., & Dériard, B. (2018). Options for the biological and physical control of *Vespa velutina* nigrithorax (Hym.: Vespidae) in Europe: A review. *Journal of Applied Entomology*, 142(6), 553-562.

Unielijst invasieve exoten. (2024). NVWA. [Online] Beschikbaar:
<https://www.nvwa.nl/onderwerpen/invasieve-exoten/unielijst-invasieve-exoten>. [Geraadpleegd op 26/02/2024]

Van Ransbeeck, H., Adriaens, T., Batsleer, F., Bonte, D., Devisscher, S., Gelaude, E., Van Donink, S., Neukermans, A., Cartuyvels, E., Wauters, J., & Hillaert, J. (2023). De vliegkunst van de Aziatische Hoornaar. *Natuurfocus*, 23(1), 13-21.

Verdasca, M. J., Godinho, R., Rocha, R. G., Portocarrero, M., Carvalheiro, L. G., Rebelo, R., & Rebelo, H. (2021). A metabarcoding tool to detect predation of the honeybee *Apis mellifera* and other wild insects by the invasive *Vespa velutina*. *Journal Of Pest Science*, 95(2), 997–1007.

Vespawatchers-Wiekpotmethode. (2023). Imkers Neteland. [Online] Beschikbaar:
<https://www.imkersneteland.be/wp-content/uploads/2023/02/vespawatchers-wiekpotmethode.pdf>. [Geraadpleegd op 16/02/2024]

Villemant, C., Haxaire, J. & Streito, J. C. (2006). The discovery of the Asian hornet *Vespa velutina* in France. *Insectes* 143, 3–7.

Wat is het probleem? (z.d.). Vespa-Watch. [Online] Beschikbaar: <https://vespawatch.be/issue/>.
[Geraadpleegd op 26/02/2024]

Wat zie je doorheen het jaar? (z.d.). Vespa-Watch. [Online] Beschikbaar: <https://vespawatch.be/>.
[Geraadpleegd op 26/02/2024]

Welcome to Colab!. (z.d.). Google Colaboratory. [Online] Beschikbaar: <https://colab.research.google.com/>.
[Geraadpleegd op 6/04/2024]

Wen, P., Cheng, Y.-N., Dong, S.-H., Wang, Z.-W., Tan, K., & Nieh, J. C. (2017). The sex pheromone of a globally invasive honey bee predator, the Asian eusocial hornet, *Vespa velutina*. *Scientific Reports*, 7, 12956.

9 APPENDICES

Appendix A: Training van YOLOv8 in de Colab-notebook

In onderstaande wordt een samenvatting gegeven van wat werd uitgevoerd in de Colab-notebook om het YOLOv8 model te trainen. De volledige Colab-notebook die werd gebruikt is beschikbaar via: <https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb>.

Als eerste installeren we YOLO.

```
!pip install ultralytics==8.0.196
```

```
import ultralytics
```

```
ultralytics.checks()
```

```
from ultralytics import YOLO
```

Hierna importeren we de dataset uit Roboflow in de notebook-sessie.

```
!mkdir {HOME}/datasets
```

```
%cd {HOME}/datasets
```

```
!pip install roboflow --quiet
```

```
from roboflow import Roboflow
```

```
rf = Roboflow(api_key="mhBkqjSO6WCt9xRKhtd")
```

```
project = rf.workspace("autobee").project("vespa-2")
```

```
dataset = project.version(1).download("yolov8")
```

Nadat de dataset geïmporteerd is in de Colab Notebook, kan de training van start gaan. Het aantal ‘epochs’ is hier cruciaal omdat dit het aantal trainingsstappen bepaald. In de originele Colab Notebook staat dit aantal op 30, maar we hebben dit verhoogd naar 110 omdat dan pas de hoogste accuraatheid van het model werd bereikt (zie hoofdstuk ‘Resultaten’).

```
%cd {HOME}
```

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=110 imgsz=800  
plots=True
```

Appendix B: De PiJuice automatisch afsluiten

Je kan via de GUI op eenvoudige veel dingen instellen. Zo kan je bijvoorbeeld de Raspberry Pi elke dag wakker laten worden op hetzelfde uur. Je kan echter geen tijd geven wanneer hij zichzelf dan ook weer uitschakelt. Dit is eenvoudig op te lossen met behulp van volgend script. Eerst maken we een nieuwe script aan via `nano /home/pi/runandturnoff.py` In dat script plakken we volgende regels code:

```
#!/usr/bin/python3

import os
import logging
from time import sleep
from pijuice import PiJuice

logging.basicConfig(
    filename = '/home/pi/pistatus.log',
    level = logging.DEBUG,
    format = '%(asctime)s %(message)s',
    datefmt = '%d/%m/%Y %H:%M:%S')

pj = PiJuice(1, 0x14)

pjOK = False
while pjOK == False:
    stat = pj.status.GetStatus()
    if stat['error'] == 'NO_ERROR':
        pjOK = True
    else:
        sleep(0.1)

# Indien op batterij, schakel uit na 8 uur
data = stat['data']
if data['powerInput'] == "NOT_PRESENT" and data['powerInput5vIo'] == 'NOT_PRESENT':
```

```

# Schrijf een statement naar de log
logging.info('Raspberry Pi op batterij. Schakel uit na 8 uur.')

# Hou de Raspberry Pi draaiende voor 8 uur
sleep(28800)

# Zorg dat wakeup_enabled en wakeup_on_charge de juiste waardes hebben
pj.rtcAlarm.SetWakeupEnabled(True)
pj.power.SetWakeUpOnCharge(0)

# Geen stroom meer naar de Raspberry Pi laten gaan om batterij niet te draineren
pj.power.SetSystemPowerSwitch(0)
pj.power.SetPowerOff(30)

# Schakel het systeem uit
os.system("sudo shutdown -h now")

else:
    # Schrijf een statement naar de log
    logging.info('Raspberry Pi aangesloten op stroom, zal niet automatisch uitschakelen')

```

Vervolgens maken we het script uitvoerbaar met `chmod +x /home/pi/runandturnoff.py`. En voegen we het bij het opstartproces via `sudo nano /etc/rc.local` In de nano-editor voegen we volgende lijn ten slotte voor de lijn `exit 0` in: `python3 /home/pi/runandturnoff.py &`

Appendix C: Resetten interne klok Pijuce verhelpen

Onderstaand script moet doorlopen worden om ervoor te zorgen dat de interne klok van de Pijuce HAT niet gereset wordt. De exacte werkwijze staat beschreven in sectie 3.2.2.

```

import pijuce, time, os

while not os.path.exists('/dev/i2c-1'):
    time.sleep(0.1)

```

```
pj = pijuice.PiJuice(1, 0x14)
pj.rtcAlarm.SetWakeupEnabled(True)
```

Appendix D: Finale Python script

```
import bme680
import os
import time
import datetime
import mysql.connector
import dropbox
import dropbox.files
import RPi.GPIO as GPIO
from ultralytics import YOLO
import glob
from picamera2 import Picamera2
from picamera2.encoders import H264Encoder
from picamera2.outputs import FfmpegOutput

#Vooraf definiëren van beeldherkenningsmodel (aanwezig in zelfde map als dit script):
model = YOLO('best.pt')

print("Started script!")

# CONNECTIEGEGEVENS
host = 'localhost'
user = '...' # Naam van de database user
password = '...' # paswoord van de user
database = 'Sensordata'

# Verbinding maken met de database
mydb = mysql.connector.connect(
    host=host,
```



```

user=user,
password=password,
database=database
)
cursor = mydb.cursor()

# Token voor Dropbox klaarzetten:
with open("dropbox_token.txt", "r") as f:
    TOKEN = f.read()
dbx = dropbox.Dropbox(TOKEN)

# bme680 instellen
try:
    sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)
except IOError:
    sensor = bme680.BME680(bme680.I2C_ADDR_SECONDARY)

sensor.set_humidity_oversample(bme680.OS_2X)
sensor.set_pressure_oversample(bme680.OS_4X)
sensor.set_temperature_oversample(bme680.OS_8X)
sensor.set_filter(bme680.FILTER_SIZE_3)

# Motionensor activeren
GPIO.setmode(GPIO.BOARD)
PIN = 18          # data pin van de bewegingssensor
GPIO.setup(PIN, GPIO.IN)
print("Start sensor...")
time.sleep(2)
print("Sensor geactiveerd...")

while True:

```

```

if GPIO.input(PIN): # Wanneer beweging gedetecteerd wordt
    # Tijdstempel genereren voor de afbeeldingsnaam
    timestamp = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    print("Beweging gedetecteerd! " + (time.strftime("%H:%M:%S")))
    time.sleep(5) # Wacht 5 seconden om een foto van een bewegend insect te vermijden

try:
    # Nieuwe afbeeldingsnaam met tijdstempel:
    image_name = f"/home/Autobeehive/Pictures/img_{timestamp}.jpg"
    os.system(f"libcamera-still -n -o {image_name}") # Foto maken
    print("Afbeelding gemaakt")
except Exception as e:
    print("Fout bij het maken van de foto:", e)

list_of_files = glob.glob('/home/Autobeehive/Pictures/*') # Namen van alle foto's in de Pictures map
latest_file = max(list_of_files, key=os.path.getctime) # Naam van de laatste foto selecteren
# Beeldherkenningsmodel laten lopen op laatst getrokken foto
results = model.predict(source=latest_file, show=False, conf=0.668, save=True)
for result in results:
    if result.bboxes:
        box = result.bboxes[0]
        class_id = int(box.cls)
        # Naam van de gedetecteerde klasse toekennen aan object name
        object_name = model.names[class_id]
        print(object_name)

        # Als het een Aziatische hoornaar is wordt er een video gemaakt
        if object_name == 'Aziatische hoornaar':
            video_name = f"/home/Autobeehive/Videos/video_{timestamp}.mp4"
            #Video maken:
            picam2 = Picamera2()
            video_config = picam2.create_video_configuration()

```

```

picam2.configure(video_config)

encoder = H264Encoder(10000000)      #Het aantal bits per seconde
output = FfmpegOutput(video_name)

picam2.start_recording(encoder, output)
time.sleep(10) #duur van video
picam2.stop_recording()
picam2.close()

print('Het is een hoornaar, dus er werd een video gemaakt!')

#Afbeelding en video opslaan in databank en dropbox
try:
    # Afbeelding en video openen en lezen als binaire gegevens
    with open(image_name, 'rb') as image_file, open(video_name, 'rb') as video_file:
        image_binary = image_file.read()
        video_binary = video_file.read()

    # Uploaden van afbeelding en video naar Dropbox
    dbx.files_upload(image_binary, f"/Raspi/Pictures/img_{timestamp}.jpg")
    dbx.files_upload(video_binary, f"/Raspi/Videos/video_{timestamp}.mp4")

    # INSERT-query uitvoeren om afbeelding en video op te slaan als een BLOB in de database
    sql = "INSERT INTO image_video_bme680 (date, host, image, video, temperature, pressure,
humidity) VALUES (%(date)s,%(host)s,%(image)s,%(video)s,%(temperature)s,%(pressure)s,%(humidity)s)"
    image_data = {
        'date': datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
        'host': host,
        'image': image_binary,
        'video': video_binary,
        'temperature': '{0:.2f}'.format(sensor.data.temperature),
        'pressure': '{0:.2f}'.format(sensor.data.pressure),

```

```

        'humidity': '{0:.2f}'.format(sensor.data.humidity)}

    cursor.execute(sql, image_data)
    mydb.commit()

    print("Afbeelding en video opgeslagen in de database")
except Exception as e:
    print("Fout bij het uitvoeren van de query:", e)
    mydb.rollback()

    time.sleep(300)          # Als een Aziatische hoornaar gedetecteerd is, wacht 5 minuten
else:                      # Bij detectie van een wespachtige, maar geen Aziatische hoornaar:
    time.sleep(60)          # Wacht een minuut

else:                      # Bij geen enkele detectie wordt de genomen afbeelding verwijderd
    os.remove(image_name)
    print("Geen detectie: afbeelding verwijderd")
    time.sleep(30)          # Bij geen detectie: wacht een halve minuut

mydb.close()

```

Appendix E: Verbinding op afstand met MariaDB

Door de nodige configuraties te doen aan MariaDB, kan een verbinding op afstand worden gelegd tussen de lokale databank op de Raspberry Pi en een computer, aan de hand van een Python script op deze computer. Belangrijk hierbij te vermelden is dat dit de databank openstelt naar de buitenwereld toe, aangezien een poort wordt geopend. Het gebruik van een sterk wachtwoord voor de databank is dus zeker vereist. Deze verbinding op afstand lijkt heel goed op de verbinding die de Raspberry Pi legt met de lokale databank om data te stockeren, alleen moet er nog een poort toegewezen worden. Voor een MySQL database is die standaard 3306. Soms kan het zijn dat die poort niet geopend staat door een firewall. De commando's `sudo apt-get install ufw` gevolgd door `sudo ufw allow 3306` openen deze poort, waarna `sudo ufw reload` de firewall herlaadt. (How to enable Remote acces to your MariaDB/MySQL database | Webdock, 2022)

Standaard laat MariaDB enkel connectie toe van de localhost, dus moet MariaDB nog zo geconfigureerd worden dat het luistert naar alle IP adressen die verbinding willen maken en de juiste inloggegevens kennen. Het MariaDB default configuratiebestand kan bewerkt worden met `sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf` (in sommige gevallen met `nano /etc/mysql/my.cnf`). Standaard staat in dit bestand het bind-adress ingesteld op 127.0.0.1, het standaard IP-adres van de local host. Door dit bind-adress te veranderen in 0.0.0.0 of door simpelweg voor het hele statement "bind-adress = 127.0.0.1" een # te plaatsen, laat MariaDB connecties toe van meerdere IP adressen. Na het opslaan en sluiten van dit bestand is het nodig de MariaDB

server te herstarten: `sudo systemctl restart mariadb`. (Configuring MariaDB for Remote Client Access | MariaDB, z.d.) Cruciaal bij het verbinden met een database vanop een ander IP adres dan de localhost, is dat een gebruiker (user) die kan bewerken vanop meerdere IP adressen er verbinding mee maakt. Dit werd in paragraaf 3.3 bekomen door de procent tekens in “CREATE USER ‘admin’@’%’ IDENTIFIED BY ‘password’;” en “GRANT ALL PRIVILEGES ON *.* TO ‘admin’@’%’ WITH GRANT OPTION;”. Als een gebruiker voor een of andere reden deze rechten niet heeft, kan dit nog steeds aangepast worden in de MySQL terminal met `GRANT ALL PRIVILEGES ON *.* TO ‘admin’@’%’ IDENTIFIED BY ‘password’`; waarbij admin en password juist worden ingevuld. Om deze wijzigingen door te voeren moet je afsluiten met `FLUSH PRIVILEGES`. Als slechts 1 specifiek IP adres gewenst is die verbinding kan maken (uit veiligheidsoverwegingen), kan `@’%’` vervangen worden door `@’<ip-adres>’`, waarbij `<ip-adres>` het juiste IP adres voorstelt. (How to enable Remote acces to your MariaDB/MySQL database | Webdock, 2022)

Een manier om uiteindelijk verbinding te maken maakt opnieuw gebruik van de mysql connector (dat opnieuw via `pip3 install mysql-connector-python` moet geïnstalleerd worden op de computer) in een Python script. Met een query kunnen alle gegevens uit de tabel opgevraagd worden (zoals in onderstaand voorbeeldscript), of een deel ervan. Met een for-lus kan vervolgens de tabel uitgelezen worden, waarbij per rij de afbeelding en video worden opgeslaan op de computer.

Voorbeeldscript in Python om alle afbeeldingen en video’s in de databank op te slaan op de computer, die verbinding maakt met de databank op de Raspberry Pi (Converting BLOB file data to image from sql database | forums.raspberrypi, 2016):

```
# Importeren van packages

import os

import mysql.connector

# Verbindingsgegevens voor de MySQL-database

HOST = #ip-adres van de Raspberry Pi
PORT = 3306
USER = #naam van de database user
PASSWORD = #paswoord van de user
DB = 'Sensordata'

# Verbinding maken met de database

db = mysql.connector.connect(host=HOST, port=PORT, user=USER, password=PASSWORD,
database=DB)

# Cursor-object voor het uitvoeren van SQL-query's
```

```

cursor = db.cursor()

# SQL-query om alle gegevens uit de tabel op te halen
sql = "SELECT * FROM image_video_bme680"

# SQL-query uitvoeren
cursor.execute(sql)

# Resultaten ophalen
myresults = cursor.fetchall()

# Mappen aanmaken (als deze nog niet bestaan) in de directory waar python project zich in bevindt
if not os.path.exists("images"):
    os.makedirs("images")

if not os.path.exists("videos"):
    os.makedirs("videos")

# Verwerken van de resultaten: de functie enumerate maakt een iterable object, dat met een for-lus wordt
uitgelezen volgens de rijen.

# Voor elke tuple of rij worden de kolommen uitgelezen en aan een variabele toegewezen.
for tup in enumerate(myresults):
    date = tup[0]          # datum (eerste kolom)
    host = tup[1]          # Naam van de host (tweede kolom)
    image_data = tup[2]    # afbeeldingsdata (derde kolom)
    video_data = tup[3]    # videodata (vierde kolom)
    temperature = tup[4]   # temperatuur (vijfde kolom)
    pressure = tup[5]      # luchtdruk (zesde kolom)
    humidity = tup[6]      # luchtvochtigheid (zevende kolom)

# Binaire afbeelding opslaan als jpg bestand
image_filename = f"{date.strftime('%Y-%m-%d_%H-%M-%S')}.jpg"
image_filepath = os.path.join("images", image_filename)

```



```

with open(image_filepath, 'wb') as g:
    g.write(image_data)

# Binaire video opslaan als mp4 bestand
video_filename = f"{date.strftime('%Y-%m-%d_%H-%M-%S')}.mp4"
video_filepath = os.path.join("videos", video_filename)
with open(video_filepath, "wb") as f:
    f.write(video_data)

# Verbinding met de database verbreken
cursor.close()

```

Appendix F: Python script automatisch laten lopen vanaf opstart Raspberry Pi

Omdat er niet altijd een scherm en muis voor handen is om het finale script op te starten, is het handig om het script te laten werken vanaf dat de Raspberry Pi opgestart wordt. Er zijn meerdere mogelijkheden om dit in te stellen, maar de meest betrouwbare optie is via een servicebestand in de systeemcontrole. Volgende werkwijze is gebaseerd op Robotics Back-End (2021). Opdat het finale script (in ons geval “final1.py” genaamd) uitvoerbaar is voor het systeem, dient deze veranderd te worden in een executabel via het commando `chmod +x final1.py`. Dit zal natuurlijk enkel werken als de terminal zich in dezelfde directory bevindt als het finale script. Vervolgens moet een nieuw service bestand aangemaakt worden via `sudo nano /lib/systemd/system/final1.service`, waarin onderstaande code terecht moet komen. Enkel de directory naar het finale script (bij ExecStart) moet hierin aangepast worden en de gebruiker (User) van de Raspberry Pi ook, al heeft die standaard de naam pi.

```

[Unit]
Description=Run script final1 from boot
After=multi-user.target

[Service]
ExecStart=/usr/bin/python3 /home/pi/Documents/final1.py      # Pas aan naar juiste directory
User=pi                                                       # Pas aan naar juiste User

[Install]
WantedBy=multi-user.target

```

Na afsluiten en opslaan van dit service bestand, dient de systeemcontrole herladen te worden via `sudo systemctl daemon-reload`. Als laatste stap activeren we dit aangemaakte servicebestand, wat ertoe zal leiden dat het meegegeven finale script zal beginnen werken vanaf de opstart van de Raspberry Pi. `sudo systemctl enable final1.service` voert deze laatste stap uit. Via `sudo systemctl disable final1.service` kan deze optie ook weer uitgeschakeld worden.