

KKC Technasium Team2

# SolarBot

Eindverslag

Jivraj Singh, Jono Wink, Xinxin Wang, Kevin Kang  
25/26

## Inhoud

Voorwoord .....	4
Inleiding .....	5
Oriëntatie .....	6
Het bedrijf.....	6
Opdrachtgever .....	6
Experts .....	7
Huidige situatie.....	8
De probleemstelling .....	8
De opdracht .....	8
Navigatie op hellende PV-oppervlakken .....	8
Effectieve reiniging van hardnekkige vervuiling .....	8
Programma Van Eisen .....	9
Veldonderzoek.....	10
Theoretisch achtergrond .....	12
Wat voor soort zonnepaneel reinigungsrobots zijn er? .....	12
Wat zijn de huidige reinigingstechnieken van robots? .....	12
Hoe werkt een zonnepaneel? .....	12
Uit welke materialen bestaat het contactoppervlak van een zonnepaneel?.....	13
Waarom is reiniging belangrijk voor zonnepanelen? .....	14
Hoe worden huidige robots voorzien van energie?.....	14
Hoe draagt een robot bij aan duurzamer onderhoud van zonnepanelen? .....	14
Wat is machinaal leren (machine learning)? .....	15
Wat zijn de gevolgen van viezigheden op zonnepanelen? .....	15
Hoe worden zonnepanelen tegenwoordig schoon gemaakt?.....	18
Handmatig schoonmaken .....	18
Watersproeiers (Water Spray Nozzle Systems) .....	18
Reiniging met vrachtwagens .....	18
Elektrodynamisch scherm (EDS) .....	18

“Wash Panel” schoonmaakrobot .....	19
Drones .....	20
Zelfreinigende & zonvolgende panelen.....	20
Concept .....	21
Software .....	21
Fase 1 Mappen van het paneel .....	21
Fase 2 Live schoonmaak en detectie .....	21
Fase 3 Nacontrole en herbehandeling .....	21
Mogelijke methoden.....	21
Schoonmaak proces .....	22
Concept Onderzoek .....	24
Welke programmeertaal is het meest effectief? .....	24
Machine learning (ML) voor visuele inspectie .....	25
Keuze voor de moderne aanpak .....	27
Covolutionele Neurale Netwerken.....	27
Siamese Neuraal Netwerken .....	28
Image Embeddings en Afstandsgebaseerde Detectie .....	29
Data Preprocessing en Normalisatie .....	30
Tile-Based Image Analysis voor Geavanceerde Inspectie.....	30
Computer Vision met OpenCV .....	32
Pad Planning/ Mapping .....	32
Ultrasone Sensoren .....	33
Raspberry pi.....	33
Unity Engine .....	34
Hoe kan deze schade op zonnepanelen voorkomen worden? .....	35
Uitwerking Concept Software .....	38
C++ “gedachtes” gedeelte .....	38
Bewegingsmodel van de robot .....	38
Navigatie en richtingscorrectie.....	39

Proportioneel controller (P-Controller) .....	39
Mapping en schoonmaakpad (spiraal) .....	40
Vuildetectie .....	41
Schoonmaak route-optimalisatie .....	41
Conclusie C++ “gedachtes” gedeelte .....	42
AI/ML Gedeelte .....	43
Hoe het AI systeem werkt .....	43
Training van het model .....	43
Detectie met embeddings.....	44
Tile based analyse.....	44
Gebruik op Raspberry Pi .....	44
Beslissingslogica .....	44
Conclusie AI/ML Gedeelte .....	45
Uitwerking Concept Hardware .....	46
PCB.....	46
Robot .....	49
Iteratief proces .....	51
Borstel en osmosewater systeem.....	54
Batterij.....	56
Eindproduct .....	58
Simulaties.....	58
Simulatie Versie 1 .....	58
Simulatie Versie 2 .....	58
Simulatie eind versie .....	59
C++ bewegingssoftware .....	59
Robot .....	60
Eindconclusie.....	62
Logboek .....	64
Literatuurlijst .....	69

## Voorwoord

Wij zijn Kevin Kang, Jivraj Singh, Xinxin Wang en Jono Wink. Wij zitten op het Keizer Karel College in 6vwo Technasium en volgen hierbij het vak Onderzoeken en Ontwerpen (O&O). Tijdens het examenjaar hebben wij ons laatste project voor O&O: de meesterproef. Wij hebben dit project gekozen omdat het over energie en duurzaamheid gaat, waar wij veel interesse in hebben. Ook omdat er veel ontwerpen en programmeren bij zit, dat voor ons ook interessant is, en waar wij al veel in gedaan hebben.

Ook willen wij bij dit project speciaal onze dankbaarheid tonen voor meneer de Graaf en mevrouw Roos die ons geholpen hebben met ondersteuning en feedback voor dit project.

# Inleiding

Sinds de uitvinding van het eerste zonnepaneel in 1954 is zonne-energie uitgegroeid tot een van de belangrijkste duurzame energiebronnen ter wereld. Zonnepanelen spelen een grote rol in de energietransitie en worden steeds vaker ingezet in grote zonneparken. Ondanks deze groei is er nog steeds een fundamenteel, praktisch probleem op het gebied van efficiëntie van de zonnepanelen.

De efficiëntie wordt namelijk beperkt, want na verloop van tijd raken zonnepanelen vervuild door stof, pollen en vogelpoep. Deze vervuiling zorgt ervoor dat minder zonlicht de zonnecellen bereikt, waardoor het rendement daalt. Om dit te voorkomen moeten zonnepanelen regelmatig worden schoongemaakt. In de praktijk gebeurt dit vaak handmatig of met relatief eenvoudige reinigingssystemen. Dit kost veel tijd, arbeid en geld, vooral bij grote zonneparken. Hierdoor wordt onderhoud soms minder vaak uitgevoerd dan eigenlijk nodig is, wat leidt tot veel, onnodig energieverlies.

Het gebruik van zonne-energie blijft doorgroeien en de behoefte aan slimmere en efficiëntere onderhoudsmethoden neemt alleen maar toe. Dankzij nieuwe technologieën zijn er mogelijkheden om dit proces te verbeteren.

In deze meesterproef wordt een robot onderzocht en gemaakt om met gebruik van machine learning en AI de zonnepanelen efficiënter schoon te maken.

# Oriëntatie

## Het bedrijf

Het bedrijf dat de opdracht geeft is een beleggingsholding bedrijf met de naam Concord New Energy (Vroeger China New Energy) Het bedrijf houdt zich bezig met engineering, procurement, constructie, gebruik en onderhoud van wind/zonne-energie-installaties, de productie van wind/zonne-energieapparatuur en andere wind/zonne-energie gerelateerde activiteiten. (Concord New Energy)

Hong Kong Pharmaceutical heeft China New Energy in 2007 gekocht en in 2015 de naam veranderd naar Concord New Energy. Ze begonnen ook rond 2011 zonneparken te kopen, en sinds 2014 werden zonneparken meer gebruikt dan hun windparken, vandaar de naamverandering. (Mancheva, 2015)

## Opdrachtgever

Dr. Bo Zhang kreeg zijn Ph.D. in robotica op de universiteit van Florida en zijn meester diploma op Universiteit van Tsinghua. Dr. Bo Zhang heeft al 20 jaar ervaring met Onderzoeken en ontwerpen (R&D), robotica, AI en groene energie.

Van 2014 tot 2018 richtte hij een startup op, die zich bezighield met de ontwikkeling van robotreinigingssystemen voor zonnepanelen.

Dr. Zhang was eerder AIoT-leider voor de hernieuwbare sector bij Envision Digital (Universon) en senior wetenschapper bij A\*STAR in Singapore. Momenteel leidt hij het Concord Clean Energy Research Center, waar hij zich richt op robotintelligentie en AI gestuurde automatisering voor duurzame, koolstofvrije energiesystemen.



Figuur 1 Dr. Bo Zhang

## Experts

Voor ons project hebben we hulp gehad van twee experts. De eerste is Harveer Singh, een derdejaars BIT-student aan de Universiteit Twente. Hij heeft ons vooral geholpen met de programmeerkant en met machine learning. Hij gaf advies over hoe we ons model het beste konden opbouwen, hoe embeddings werken binnen een Siamese Neural Network en hoe we hyperparameters konden afstellen voor betere prestaties.

Daarnaast hebben we hulp gekregen van Alec Li, een derdejaars student Electrical Engineering. Hij heeft ons ondersteund bij de hardware-kant van het project. Hij hielp ons met het ontwerpen van het schema, het kiezen van de juiste elektrische componenten en het zorgen voor een stabiele spanningsvoorziening. Ook heeft hij meegeholpen met het ontwerpen van de PCB, waarbij hij lette op een goede layout en een logische plaatsing van de onderdelen. Dankzij hun hulp konden we zowel de software als de hardware van ons project op een hoger niveau uitwerken.

## Huidige situatie

Op dit moment worden zonnepanelen in Nederland ongeveer eens per zes maanden schoongemaakt. In drogere gebieden gebeurt dat zelfs elke maand, en in industriële regio's elke twee weken. Nu wordt dit onderhoud meestal handmatig gedaan: mensen lopen met waterrugtassen rond om de panelen schoon te spuiten, of er wordt gebruikgemaakt van tractoren met watersystemen. Dit kost enorm veel fysieke inspanning, of ontzettend veel geld, vooral in extreem droge gebieden. Ook in andere plaatsen waar onze opdrachtgever zulke operaties voortzet waar het nog droger is zoals Zuidoost-Azië.

## De probleemstelling

Stof en vuil op zonnepanelen (PV-modules) verlaagt de energieopbrengst aanzienlijk. Zelfs lichte vervuiling zoals stof, pollen of vogelpoep kan al tot meer dan 10% rendementsverlies leiden, vooral in droge of stoffige gebieden waar dit vuil er minder snel af wordt gespoeld door regen. Wanneer panelen sterk vervuild raken, kan de opbrengst zelfs met tientallen procenten dalen. Om deze verliezen te voorkomen is regelmatige en effectieve reiniging erg van belang, maar huidige methoden zijn te duur, arbeidsintensief en inefficiënt of voldoen op een andere manier niet aan de eisen van de opdrachtgever.

## De opdracht

Ontwikkel een autonome reinigungsrobot die het onderhoud van zonneparken kan verbeteren door het oplossen van twee belangrijke technische uitdagingen:

### Navigatie op hellende PV-oppervlakken

Ontwerp een robot die zich veilig en stabiel kan voortbewegen over zonnepanelen die op hellende of ongelijke constructies zijn geplaatst, zonder gebruik van rails. De robot moet valdetectie en -preventie hebben en zich kunnen aanpassen aan verschillende hellingshoeken en vormen van panelen.

### Effectieve reiniging van hardnekkige vervuiling

Ontwerp een reinigingssysteem dat verschillende soorten vervuiling kan verwijderen, zoals stof, vervuiling in kieren, pollen en vogelpoep, terwijl schade aan de panelen wordt voorkomen en waterverbruik minimaal is.

# Programma Van Eisen

## Functionele eisen (opdrachtgever)

1. De robot moet kunnen werken zonder rails of een extern geleidingssysteem.
2. De robot moet stabiel kunnen bewegen op hellende of oneffen oppervlakken.
3. De robot moet zelfstandig kunnen navigeren en voorkomen dat hij van de panelen afvalt.
4. De robot moet verschillende soorten vervuiling kunnen verwijderen, zoals stof, pollen, aangekoekte vervuiling in kieren en vogelpoep.
5. De robot moet na het opstarten zo min mogelijk menselijke controle of begeleiding nodig hebben.

## Prestatie-eisen

1. De robot mag max 35kg wegen (om schade aan panelen en frames te voorkomen). (Asewe, 2023)
2. De robot moet een gebruiksduur van 55 minuten per acculading hebben om een aanzienlijk deel van het zonnepark te reinigen, 30 m<sup>2</sup> per lading als robot op accu rijdt.
3. De robot moet het vermogen van vervuilde zonnepanelen kunnen herstellen met een meetbare efficiëntiewinst. Na reiniging moet de robot minimaal 80% van het rendementsverlies terugwinnen.
4. De robot moet bestand zijn tegen warme omgevingen van boven 40°C, de robot moet ook compleet waterdicht zijn.

## Eisen rondom onderhoud en duurzaamheid

1. De robot moet zodanig ontworpen zijn dat minimaal 80% van de onderdelen zonder speciaal gereedschap binnen 30 minuten vervangen kan worden door één persoon met basis technische kennis.
2. De robot mag in volledig operationele toestand (inclusief watertank) niet meer dan 40 kg wegen, en het waterverbruik per schoonmaakcyclus mag maximaal 20 liter per 10 m<sup>2</sup> zonnepanelen bedragen.

## Optionele of toekomstige uitbreidingen

1. Mogelijkheid voor afstandsmonitoring of statusrapportage via een netwerkverbinding.
2. Mogelijkheid voor automatisch opladen via een dockingstation.
3. Ondersteuning voor samenwerking tussen meerdere robots.
4. De robot moet volledig waterloos kunnen reinigen

## Veldonderzoek

Voor het veldonderzoek is er een excursie geweest naar een zonnepark in Utrecht waar momenteel geen zonnerobots in gebruik zijn. Uit dit onderzoek is gebleken dat het park Astronergy 585-605W (ASTRO N5) panelen gebruikt (astronergy, n.d.), de mechanische specificaties van dit paneel zijn online te vinden (astron5, n.d.). Hierbij gegeven staat de “maximum mechanical test load”, waarbij gemeten is hoeveel druk voor een langdurige tijdsperiode op het vooroppervlak van de zonnepanelen uitgeoefend kan worden zonder dat deze schade verkrijgen in de zonnecellen of het beschermende glas. De druk die gegeven is voor de Astronergy panelen is 5400 Pa (pascal).



*Figuur 2 Astronergy zonnepanelen*

Omdat er niet vanuit gegaan kan worden dat alle zonnepanelen in het bezit van opdrachtgever, Concord New Energy, dezelfde krachten kunnen weerstaan, moet er onderzoek gedaan worden naar de soorten zonnepanelen die de Opdrachtgever gebruikt, zodat de robot zonder schade te verrichten op elk paneel kan rijden.

Er is verder gevraagd naar de maximale temperatuur die de oppervlakte van zonnepanelen in dit park in de laatste paar maanden bereikt is. Dit was 35 graden Celsius, en zal bepalend zijn voor de soorten materialen die gebruikt kunnen worden op de robot die in contact komen met de panelen.

Ook door het kijken naar vuil van de zonnepanelen heeft het team gezien dat meeste van het vuil op een zonnepaneel bestaat uit: Vogelpoep, zand, modder en andere stoffen. Dit vormt een groot risico voor het rendement van de zonnepanelen.



*Figuur 3 Vuil op zonnepaneel*

Er is ook gebleken dat de grond in het zonnepark zompig en oneven was, dus lastig om te navigeren met een robot. Dit zonnepark wordt momenteel om de 6 maanden schoongemaakt door mensen met sproeiers met onder andere osmosewater, dat de panelen schoonmaakt zonder vlekken achter te laten. Andere parken gebruiken bijvoorbeeld een tractor met een borstel die langs de zonnepanelen gaat en deze schoonborstelt. Ook is er gevonden dat vuil zich vooral onderaan de zonnepanelen verzamelde, naast een paar vlekken vogelpoep die zich op de panelen bevonden. Deze informatie kan gebruikt worden voor de robot om te focussen op deze soorten vuil. Ook is verteld dat indelingen tussen zonnepanelen veel verschillen per park, zoals de afstand tussen rijen en kolommen van zonnepanelen.



*Figuur 4 onderkant en grond van zonnepaneel*



*Figuur 5 Foto in park*

# Theoretisch achtergrond

## Wat voor soort zonnepaneel reinigingsrobots zijn er?

Er bestaan veel verschillende soorten reinigingsrobots voor zonnepanelen. Welke robot het best geschikt is, hangt af van factoren zoals de grootte van het zonnepaneel, de hellingshoek en het gewicht dat de constructie kan dragen. Een goed voorbeeld is de *SolarROBOT Compact* (Jana, 2025). Deze robot wordt vooral gebruikt bij kleinere zonnepaneelinstallaties met een beperkte helling. Door het lichte gewicht en de compacte bouw is hij daar een stuk efficiënter dan grotere, professionele robots.

In een zonnepark waar zonnepanelen van verschillende groottes liggen, is het belangrijk dat de robot zijn schoonmaaktechniek kan aanpassen aan het type paneel dat hij reinigt (SolarGlanzz, 2025).

Soms bevinden zonnepanelen zich op moeilijk bereikbare of hoge plekken. Handmatig schoonmaken is dan niet alleen onveilig, maar ook onpraktisch. Voor zulke situaties worden speciale schoonmaakdrones ingezet. Deze drones kunnen eenvoudig over de panelen vliegen, waardoor ze het vuil sneller en veiliger verwijderen (WLSolarCleaning, sd)

Kortom, het is belangrijk dat een reinigingsrobot goed wordt afgestemd op de specifieke omstandigheden van een zonnepark, zodat de schoonmaak zo efficiënt en veilig mogelijk verloopt.

## Wat zijn de huidige reinigingstechnieken van robots?

De meest vergelijkbare robot soort zijn de raam schoonmaak robots. Die schoonmaak robots gebruiken zuigkracht om verticaal op de raam te blijven plakken. Dat wordt gedaan door een motor die een vacuüm ruimte maakt door met een ventilator snel lucht te verplaatsen. Raam schoonmaak robots kunnen ook magnetisme gebruiken om op de raam te blijven. Maar magnetisme is niet voor ons bruikbaar doordat het magnetisme zou kunnen zorgen voor storing. (Cleaning, sd)

Voor het reinigen van ramen wordt nu met een hoogdruk water spuit gedaan. (Ecovacs, 2025)

## Hoe werkt een zonnepaneel?

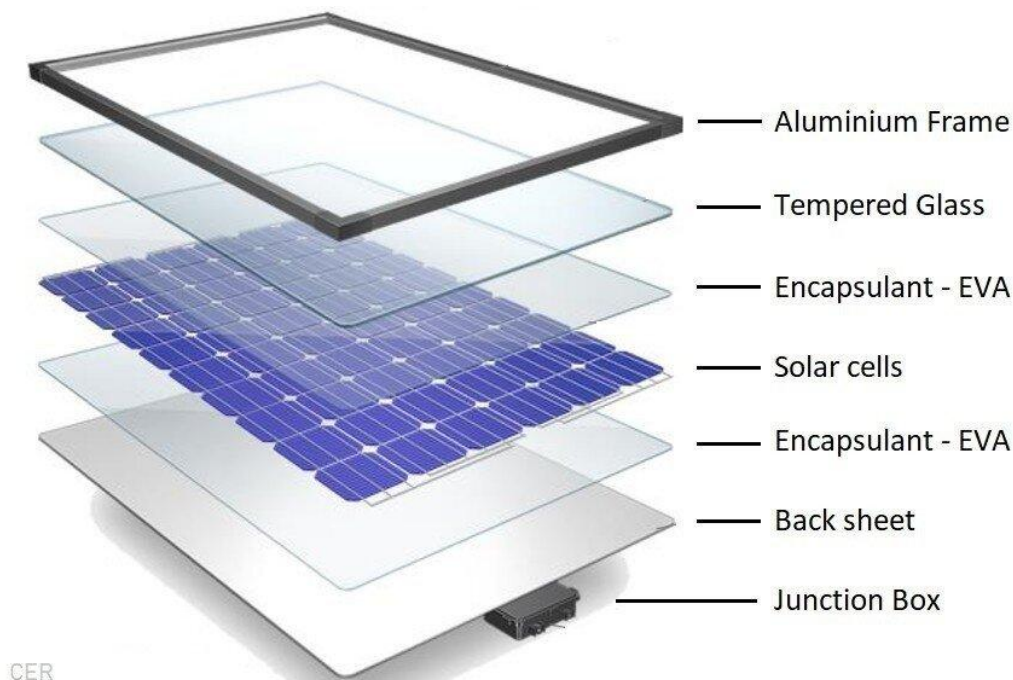
Een zonnepaneel wekt energie op door lichtenergie van de zon om te zetten in elektrische energie (zonneplan, sd), ze gebruiken fotonvoltaïsche cellen waar 2 lagen in zitten, en als een foton in de bovenste laag een elektron raakt, wordt deze verplaatst naar de onderste laag, waardoor er tussen de 2 lagen een spanning ontstaat. Dit proces werkt het beste als zoveel

mogelijk elektronen verplaatst worden en dus zoveel mogelijk fotonen binnenkomen, dus als er zoveel mogelijk licht de bovenste laag raakt en er niks in de weg zit.

## Uit welke materialen bestaat het contactoppervlak van een zonnepaneel?

Een zonnepaneel bestaat uit meerdere lagen van materiaal, het heeft de zonnecellen die ook daadwerkelijk energie produceren, en het metalen geraamte zodat de vorm krijgt. Waar dit project het meest nieuwsgierig voor is, is de bovenste beschermde laag. (Svarc, 2022)

Van de zonnepanelen die in het zonnepark van ons veldonderzoek stonden, de Astronergy panelen 585-605W, (Astronergy, 2025) staat er niet precies wat voor soort glas de ASTRO N5 gebruikt, maar wel de dikte ervan: "Front/ Back glass: 2.0+2.0 mm". Uit verder onderzoek blijkt dat bijna alle zonnepanelen hetzelfde soort glas gebruikt, gehard glas. Dat komt doordat uit onderzoek blijkt dat 98% van alle zonnepanelen Monocrystalline zonnepanelen zijn. En die Monocrystalline solar panels bestaan altijd uit gehard glas oftewel "tempered" glas. (Abeni, 2025) Het ontwerp dat gemaakt wordt is voor de meeste zonnepanelen. Dus heeft het team ervoor gekozen om contactoppervlak als gehard glas aan te nemen. Zo kan het team een ontwerp maken dat werkt voor 98% van zonnepanelen.



Figuur 6 Bouw van zonnepaneel

## Waarom is reiniging belangrijk voor zonnepanelen?

Zonnepanelen worden voortdurend blootgesteld aan allerlei stofdeeltjes in de lucht, vogeluitwerpselen en andere soorten vuil. Dat is vaak een probleem in gebieden waar het niet zo vaak regent of die in het algemeen minder schoon zijn. Zulke vuil zorgt voor schaduw en voorkomt dat zonlicht de cellen bereikt. Daardoor vermindert het de werking en heb je een lager rendement. Door de zonnepanelen te reinigen wordt er dus voor een hogere opbrengst gezorgd en dat bespaart je ook veel geld. Daarnaast verlengt het de levensduur van de installatie. (Zoofy, 2025)

## Hoe worden huidige robots voorzien van energie?

In tegenstelling tot traditionele robots die afhankelijk zijn van constante verbindingen met de elektriciteitsnet of frequente batterijvervangingen, benutten robots nu vooral energie uit zonlicht via ingebouwde zonnepanelen. Deze schone energie wordt omgezet in de elektrische energie die nodig is voor beweging, sensoren en computerbewerkingen.

(Noor, 2025)

## Hoe draagt een robot bij aan duurzamer onderhoud van zonnepanelen?

Als een robot wordt gebruikt om zonnepanelen schoon te maken, vervangt die de oude manier van werken. Op veel zonneparken worden nu nog tractoren met borstels of mensen met sproeiers ingezet, wat veel energie, water en werk kost. Een robot die op (zonne)stroom werkt is veel duurzamer, omdat er geen benzine- of dieselauto's nodig zijn om mensen of machines heen en weer te rijden. Bovendien kan de robot zichzelf opladen met de energie van de zonnepanelen, waardoor het schoonmaken niet alleen makkelijker, maar ook een stuk milieuvriendelijker wordt. Robots kunnen continu aanwezig zijn, vaker schoonmaken dan mensen, minder osmosewater gebruiken en veel preciezer werken, waardoor ze minder schade aan de panelen of de grond veroorzaken (A comprehensive review on solar panel cleaning robot technologies, 2023).

Daarnaast levert het gebruik van een autonome robot op de lange termijn ook flinke voordelen op voor energieverbruik en kosten. Doordat er minder mensen en voertuigen nodig zijn, wordt er minder energie verspild aan transport en handmatig werk. De robot kan bijna automatisch en regelmatig schoonmaken, waardoor de zonnepanelen bijna altijd in topconditie blijven. Dat betekent dat ze veel meer duurzame energie kunnen opwekken dan wanneer mensen maar af en toe komen schoonmaken. Op die manier wordt het hele zonnepark efficiënter en duurzamer, en gaat het onderhoud een stuk makkelijker (Performance Analysis of a Solar Panel Cleaning Autonomous Robot (SPCR) with comparative study, 2023).

## Wat is machinaal leren (machine learning)?

Machine learning is een vorm van computertechniek waarbij algoritmen worden gemaakt om dingen te leren zoals mensen dat doen. In plaats van dat een computer precies verteld wordt wat hij moet doen, leert hij zelf door gegevens uit zijn omgeving te bekijken. Machine learning wordt tegenwoordig bijna overal gebruikt, en heeft daardoor invloed op het dagelijks leven van miljarden mensen.

Er zijn heel veel toepassingen. Denk bijvoorbeeld aan patroonherkenning, computer vision, ruimtevaart, financiën en medische of biologische onderzoeken. In al deze gebieden wordt machine learning op verschillende momenten ingezet.

In de medische wereld, bijvoorbeeld bij de behandeling van kanker met ioniserende straling (radiotherapie), speelt machine learning een grote rol. Radiotherapie bestaat uit meerdere stappen, zoals doktersbezoeken, het maken van een behandelplan en het bepalen hoe een patiënt reageert op de straling. Machine learning helpt om deze processen sneller en nauwkeuriger te maken. Het wordt gebruikt bij het maken van behandelplannen, beeldgestuurde bestraling, het volgen van ademhalingsbewegingen, en bij het voorspellen van hoe goed de behandeling zal werken. (Murphy, sd)

In de financiële wereld wordt machine learning weer op andere manieren gebruikt, zoals voor het voorspellen van faillissementen, aandelenkoersen, olieprijsen en voor het beheren van portefeuilles. Het helpt ook bij het opsporen van witwaspraktijken en andere vormen van fraude. (Artificial intelligence and machine learning in finance: A bibliometric review, 2022)

Machine learning is tegenwoordig diep verweven met onze samenleving. Het zit in onze telefoons, auto's, ziekenhuizen en banken. Veel dingen die we dagelijks gebruiken zijn op de een of ander manier afhankelijk van machine learning, waardoor het steeds belangrijker wordt in ons leven en onze toekomst.

## Wat zijn de gevolgen van viezigheden op zonnepanelen?

Stof op zonnepanelen is een ingewikkeld probleem dat ontstaat door allerlei factoren uit de omgeving. Wind (richting en snelheid), temperatuur, luchtvervuiling, luchtvochtigheid en zelfs de ligging van een zonnepark spelen allemaal een rol. Het vuil dat op de panelen terecht komt kan van alles zijn, zoals zand, aarde, klei, vogelpoep, pollen en bladeren. Dit zorgt ervoor dat er minder zonlicht bij de zonnecellen komt, waardoor de opbrengst daalt. Daarnaast kunnen er schaduwplekken ontstaan en kan de temperatuur van het paneel omhooggaan.

Om te zorgen dat de panelen goed blijven werken, is het daarom belangrijk dat ze op de juiste manier worden schoongemaakt.

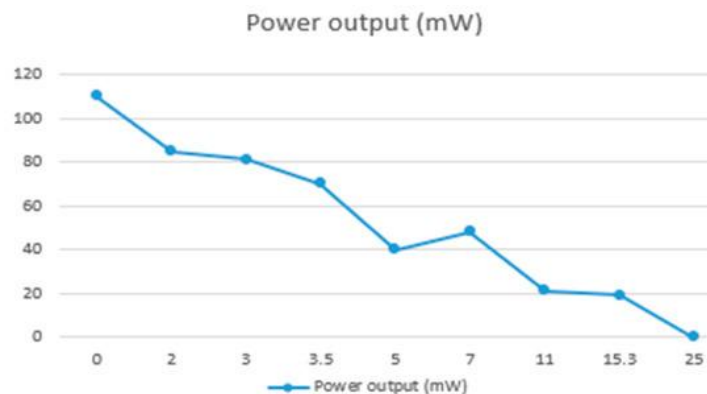
Experimenten zijn uitgevoerd om nauwkeuriger te kunnen bepalen hoe groot de invloed van zulke vervuiling is. Hieruit volgde dat een accumulatie van stof kan leiden tot een daling in het maximum opbrengst van de zonnepanelen van tientallen procenten, bijvoorbeeld 6% in een week en 20% per maand. Dit komt door de manier waarop zonlicht en stof interacteren. Een deel van de energie wordt geabsorbeerd door stof en omgezet in warmte-energie, terwijl een andere portie wordt verspreid over het oppervlak van de stof, slechts een kleine fractie bereikte de onderliggende cellen in het paneel.



*Figuur 7 Vuil op zonnepaneel*

Als bladeren of vogelpoep een deel van een zonnepaneel blokkeren, daalt de opbrengst. Wanneer panelen in serie zijn aangesloten, verlaagt één zwak (beschaduwd) paneel de prestaties van alle panelen in de reeks. Daarom zijn er elektrische componenten parallel geschakeld met substrings van zonnecellen om vermogensverlies en schade door gedeeltelijke schaduw of defecte cellen te voorkomen. Deze componenten heten bypass-diodes. Ze leiden de stroom om de beschaduwde cellen heen zodat het systeem blijft werken, maar de totale opbrengst gaat alsnog omlaag.

De regio is dus erg van belang, in droge woestijngebieden hopen stof en vuil zich snel op op zonnepanelen. Dit kan een dikke laag vormen die het zonlicht blokkeert en zo de opbrengst aanzienlijk verlaagt. Afhankelijk van de hoeveelheid stof kan de efficiëntie binnen korte tijd tot circa 20–35% dalen. Hierdoor neemt de elektriciteitsproductie sterk af.

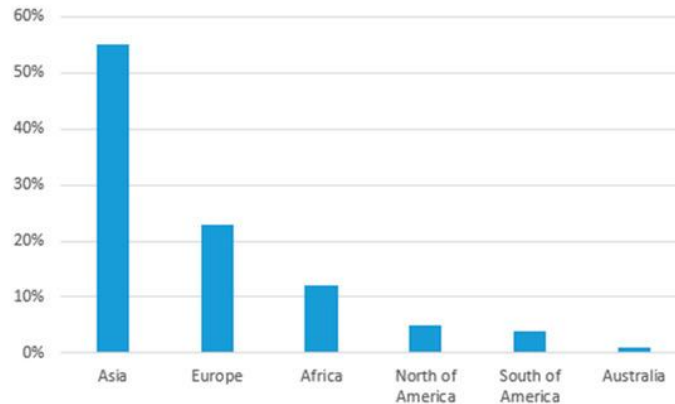


*Figuur 8 Het effect van stof ophoping op elektrische energieproductie van zonnepanelen*

De meeste onderzoeken naar stofophoping en vervuiling op zonnepanelen komen uit Azië. Vooral Saudi-Arabië loopt hierin voorop. Dat komt doordat:

- Ongeveer 60% van alle nieuwe zonnepanelen wereldwijd in Azië wordt geplaatst.
- Grote delen van Azië zijn droog, stoffig en winderig, waardoor er veel stof op zonnepanelen terechtkomt.

Afrika heeft ook veel gebieden waar veel zand en stof voorkomen.



*Figuur 9 Contributie van het effect van stof ophoping op zonnepanelen in verschillende continenten*

In kustgebieden kunnen zonnepanelen last krijgen van zout uit de zee. Dit kan op het oppervlak vast gaan zitten en zelfs corrosie veroorzaken, waardoor de panelen een heel stuk minder effectief worden. Soms kan het rendement zelfs met 40% dalen en zulke vuil is ook niet met alleen water schoon te maken.

In steden en industriële gebieden is de lucht vaak viezer door verkeer en fabrieken. Het vuil dat hierdoor op de zonnepanelen terechtkomt is vaak plakkerig en moeilijk schoon te maken, hiervoor geldt hetzelfde, alleen water of regen is niet genoeg om het schoon te maken.

In gebieden waar het vaak regent en de panelen schuin staan, zoals hier in Nederland, spoelt regen veel van het vuil ervan af.

Vuil op zonnepanelen heeft dus duidelijk een grote invloed op de opbrengst. In sommige gebieden kan dit het rendement zelfs met tientallen procenten verlagen. Omdat stof en andere viezigheid niet altijd vanzelf weggaat, is goede reiniging door middel van slimme techniek belangrijk om de panelen goed te laten werken. (Rachid, 2023)

## Hoe worden zonnepanelen tegenwoordig schoon gemaakt?

### Handmatig schoonmaken

Tegenwoordig bestaan er veel verschillende manieren om zonnepanelen schoon te maken. De eerste en meest simpele methode handmatig reinigen. Deze aanpak heeft helaas meerdere nadelen. Het levert namelijk maar weinig voordeel op als je kijkt naar de jaarlijkse opbrengst, omdat je niet dagelijks mensen eropuit kunt sturen om de panelen te reinigen. Daarnaast is het proces inefficiënt en brengt het risico's met zich mee: er wordt vaak veel water verspild en de mensen die het werk uitvoeren hebben soms te weinig kennis, wat schade aan de panelen kan veroorzaken.



Figuur 10 Schoonmaak proces van PV-zonnepanelen

### Watersproeiers (Water Spray Nozzle Systems)

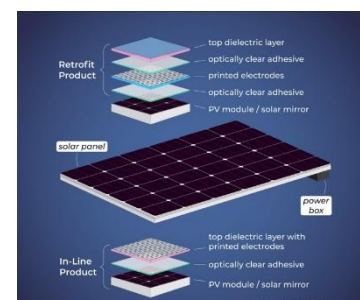
Bij dit systeem worden zonnepanelen schoongespoeld met water dat via sproeikoppen over de panelen wordt verdeeld. Het water komt uit een ondergrondse tank die automatisch wordt aangevuld. Sensoren en slimme besturing regelen hoeveel water er nodig is. Dit systeem kan automatisch reinigen waardoor de opbrengst flink wordt verbeterd.

### Reiniging met vrachtwagens

Voor grote zonneparken wordt vaak een truck met een borstel gebruikt. De truck rijdt tussen de rijen panelen door en maakt alles schoon. De druk van de borstel wordt gecontroleerd, zodat de panelen niet beschadigen. Het is een handige methode voor veel panelen tegelijk, maar je moet genoeg ruimte hebben en het blijft een beetje riskant als het terrein ongelijk is.

### Elektrodynamisch scherm (EDS)

Een innovatieve manier is het EDS-systeem. Hierbij zit er een dunne laag met kleine elektroden op de panelen die het stof elektrisch wegduwen. Zo blijft het oppervlak schoon zonder dat er water of borstels nodig zijn. Het werkt ook op gebogen of lastig bereikbare panelen, waardoor ze langer optimaal blijven werken.



Figuur 11 EDS "Stack"

## “Wash Panel” schoonmaakrobot

De Wash Panel-robot rijdt zelfstandig over de zonnepanelen en maakt ze schoon met een borstel en waterstraal. Hij kan op verschillende soorten daken en installaties worden gebruikt en kan zelfs op afstand gevolgd worden via een telefoon. Testen laten zien dat de opbrengst duidelijk stijgt, vooral bij panelen die eerst flink vuil waren.



*Figuur 12 De "Wash Panel" schoonmaak robot*

## Drones

Drones worden ook ingezet om panelen schoon te maken, vooral in grote of moeilijk bereikbare zonneparken. Ze blazen stof en vuil van de panelen met de krachtige luchtstroom van hun propellers. De drones volgen vooraf bepaalde routes zodat elk paneel wordt bereikt. Het werkt het beste in droge en stoffige gebieden en kan de energieproductie flink verbeteren.



*Figuur 13 Drone gebruikt voor paneel reiniging*

## Resola-robot

De Resola-robot kan zelfstandig van paneel naar paneel rijden zonder rails of draden. Hij gebruikt een borstel, wiper en waterreservoir om het vuil te verwijderen en beschikt over sensoren die hem helpen obstakels te vermijden. Regelmatig gebruik van deze robot zorgt voor schonere panelen en een meetbare stijging in de energieopbrengst.



*Figuur 14 Resola, PV-paneel schoonmaak robot*

## Zelfreinigende & zonvolgende panelen

Er bestaan ook panelen die zichzelf schoonmaken en tegelijkertijd de zon volgen. Een wiper veegt het vuil weg terwijl motoren het paneel draaien zodat het altijd optimaal zonlicht opvangt. Sensoren meten de zon en sturen alles automatisch aan. Zo blijven de panelen schoon en leveren ze meer energie, zonder dat iemand er zelf iets voor hoeft te doen. (Rachid, 2023)

# Concept

## Software

Op basis van de theoretische achtergrond werden concepten bedacht voor de software. Het schoonmaakproces is opgesplitst in drie fasen:

### Fase 1 Mappen van het paneel

In deze fase tekent het systeem het zonnepaneel als een kaart met behulp van verschillende sensoren. Op basis van deze data wordt een virtuele kaart opgebouwd, waarbij een coördinatensysteem ontstaat. Binnen deze virtuele kaart wordt een schoonmaak pad bepaald, dat het systeem vervolgens zal volgen.

### Fase 2 Live schoonmaak en detectie

Tijdens het schoonmaken staat de camera continu achter de robot. De live beelden worden doorgestuurd naar een centrale computer, waar machine learning wordt toegepast om viezigheid te detecteren en de coördinaten ervan te noteren.

### Fase 3 Nacontrole en herbehandeling

Na het voltooien van het standaard schoonmaak pad keert het systeem terug naar de coördinaten die tijdens fase 2 als vies zijn gemarkeerd, om deze plekken opnieuw schoon te maken.

## Mogelijke methoden

Om de drie fasen uit te voeren, hebben we verschillende methodes bedacht:

#### *Manier 1*

Machine learning wordt gebruikt om het verschil tussen schoon en vies te leren op een laptop. Het getrainde model wordt vervolgens op het systeem zelf geïmplementeerd, zodat de robot kan bepalen welke delen van het paneel schoon zijn.

#### *Manier 2*

Aan het begin wordt een deel van het paneel meerdere keren schoongemaakt om zeker te zijn dat het 100% schoon is. Daarna wordt een foto gemaakt, die gebruikt wordt om live beelden van de camera te vergelijken op kleurverschillen om viezigheid te detecteren. Dit is meer klassieke computer vision dan machine learning.

#### *Manier 3*

Net als bij manier 2 wordt een deel van het paneel meerdere keren schoongemaakt tot het volledig schoon is. Vervolgens wordt er een machine learning model getraind op deze

referentiefoto om verschillen te detecteren in de live camera beeld, zodat de coördinaten van vuil worden genoteerd.

## Schoonmaak proces

### *Initiatie*

De robot wordt geplaatst in de linkeronderhoek van het zonnepaneel. Vervolgens wordt het programma gestart.

Na het opstarten activeert de robot de ultrasone sensoren en bereidt het systeem zich voor op het mappen van het zonnepaneel.

### *Mappen van het zonnepaneel*

Tijdens de map fase volgt de robot de rand van het zonnepaneel met behulp van ultrasone sensoren.

De robot heeft twee ultrasone sensoren: een aan de voorkant en een aan de linkerzijde.

De voorste sensor detecteert of er nog een oppervlak voor de robot aanwezig is. Wanneer deze sensor geen object meer meet, betekent dit dat de robot een rand van het paneel heeft bereikt.

Tegelijkertijd controleert de linker sensor of er nog steeds een rand naast de robot aanwezig is. Wanneer dit het geval is, herkent het systeem dit als een hoekpunt.

Bij het detecteren van een hoek draait de robot 90 graden naar rechts en vervolgt hij het volgen van de rand. Dit proces wordt herhaald totdat de robot weer terugkeert naar het startpunt. Op deze manier wordt de volledige omtrek van het zonnepaneel in kaart gebracht. Hiermee wordt een navigatie gepland die de robot zal gaan volgen.

### *Navigatie volgen*

De robot volgt tijdens het schoonmaken een spiraalvormig pad naar binnen, zodat het volledige oppervlak van het zonnepaneel systematisch wordt gereinigd.

Tijdens het volgen van dit pad is de camera aan de achterzijde van de robot continu actief.

De camerabeelden worden gebruikt om vervuiling te detecteren. Wanneer vuil wordt herkend, slaat het systeem de bijbehorende coördinaten op. Deze locaties worden niet direct opnieuw gereinigd, maar gemarkeerd zodat de robot hier in een latere fase naar kan terugkeren voor extra schoonmaak.

### *Terugkeren naar vervuilde gebieden*

Na het voltooien van het reguliere schoonmaak-pad worden alle opgeslagen coördinaten van vervuilde gebieden geanalyseerd. Op basis van deze gegevens wordt een nieuw navigatie-pad berekend.

De robot volgt dit pad om gericht terug te keren naar de eerder gemarkeerde locaties en deze opnieuw schoon te maken. Op deze manier krijgen hardnekkige vervuilingen extra aandacht, zonder dat het volledige zonnepaneel opnieuw gereinigd hoeft te worden.

# Concept Onderzoek

## Welke programmeertaal is het meest effectief?

Er zijn meerdere programmeertalen die geschikt kunnen zijn voor het ontwikkelen van software voor machine learning en computer vision. Bij het kiezen van de juiste taal komen vooral C++, Java en Python naar voren. (Cass, 2024)

**C++** wordt vaak gebruikt wanneer snelheid en efficiëntie het belangrijkste zijn, bijvoorbeeld bij real-time toepassingen, robotica of software die op apparaten met beperkte rekenkracht draaien. Door de manier waarop C++ met geheugen en hardware is geïntegreerd, kunnen developers makkelijker hun programma beter maken zodat het beter kan presteren. Hierdoor werkt C++ dus beter als iets snel verwerkt moet worden en zo min mogelijk vertraging heeft.

Het nadeel van C++ is dat het minder minder goed is voor experimenten die snel uitgevoerd moeten worden of het ontwikkelen van nieuwe modellen. In tegenstelling tot Python heeft C++ niet dezelfde uitgebreide bibliotheken zoals TensorFlow of PyTorch, die veel van het werk voor machine learning en computer vision al doen. Dit betekent dat je meer zelf moet programmeren en debuggen, wat het ontwikkelen van nieuwe algoritmes veel langer en ingewikkelder maakt. (Trigolos, sd)

**Java** staat erom bekend dat het heel stabiel is en dat programma's die in Java geschreven zijn op allerlei verschillende systemen en apparaten kunnen draaien. Dus blijven programma's in Java betrouwbaar werken, ook in grote en complexe projecten. Deze eigenschappen maken Java populair voor algemene softwareontwikkeling, vooral wanneer een systeem lange tijd moet blijven draaien zonder fouten.

In machine learning en computer vision wordt Java echter veel minder gebruikt. De belangrijkste reden hiervoor is dat de meeste moderne bibliotheken en frameworks, zoals TensorFlow, PyTorch en OpenCV, vooral in Python beschikbaar zijn. Deze bibliotheken bieden kant-en-klare functies voor het bouwen, trainen en testen van modellen, waardoor het werk veel sneller en eenvoudiger gaat. In Java bestaan vergelijkbare tools wel, zoals Weka en H2O, maar deze zijn vaak minder uitgebreid en minder makkelijk te gebruiken, vooral voor beginners.

Daarnaast is de community rond Java voor machine learning kleiner dan die van Python. Dit betekent dat er minder uitleg video's, voorbeelden en ondersteuning beschikbaar zijn. Hierdoor wordt Java meestal alleen gekozen wanneer er specifieke redenen zijn, zoals bestaande systemen die al in Java zijn geschreven, of wanneer stabiliteit en compatibiliteit boven alles gaan. (Deremuk, 2024)

Voor dit project is Python dus het meest geschikt, vooral omdat je er makkelijk mee kunt experimenteren, snel modellen kunt bouwen, trainen en testen, en sneller dan in C++ of Java fouten kunt terugvinden. Ook werkt Python goed samen met robotica-systemen zoals ROS, wat handig is als je de software op een echte robot wilt laten werken. Daarom is Python voor dit project de meest logische en overzichtelijke keuze. (Davis, 2025)

## Machine learning (ML) voor visuele inspectie

Om ML voor visuele inspectie te begrijpen en te kunnen toepassen moeten we eerst een beter begrip hebben van hoe een ML model “leert”. Hierbij wordt er gedifferentieerd tussen “supervised” learning en “representational” learning.

Supervised learning, “onder begeleiding leren”, is een zeer belangrijk onderdeel van machine learning. Eigenlijk is het de meest gebruikte methode. Het komt hierop neer: je geeft een computerprogramma een heleboel voorbeelden waarbij je al weet wat het goede antwoord is.

Stel je voor: je voedt het algoritme met duizenden gelabelde foto's. Bij elke foto zeg je: "dit is een kat" of "dit is een hond". Het algoritme gaat dan op zoek naar patronen in de data, welke pixels, vormen of kleuren horen meestal bij een kat, en welke bij een hond? Het leert de relatie tussen de foto, de invoer dus en het label, de uitvoer. Als het genoeg voorbeelden heeft gezien, kun je het een compleet nieuwe foto laten zien, en het kan dan zelf voorspellen of het een kat of een hond is.

Dit wordt niet alleen gebruikt voor plaatjes. Supervised learning wordt overal toegepast: in Natural Language Processing (NLP, zoals chatbots), beeld- en videoclassificatie, in de medische industrie bij analyses om bijvoorbeeld scans te interpreteren, en nog vele andere toepassingen.

Er zijn allerlei verschillende technieken en algoritmes voor supervised learning, elk met hun eigen voor- en nadelen. Sommige zijn sneller, andere zijn nauwkeuriger. Een groot risico is overfitting: dan leert het model je trainingsvoorbeelden zo goed dat het ze uit z'n hoofd kent, maar kan het geen nieuwe, onbekende data meer goed voorspellen. Daarom is het belangrijk om het model goed te testen met meetwaarden. Onder andere; **accuracy**, **precision** en **recall**.

Supervised learning is dus de basis voor een heleboel slimme technologie die we elke dag tegenkomen. (Tiwari, 2022)

Nu volgt het concept van “representational” learning. Representationeel leren is een heel belangrijk onderdeel van moderne AI, vooral in deep learning. Het gaat erom dat een computermodel zelf nuttige kenmerken leert uit ruwe data, zoals foto's. In plaats van dat een programmeur handmatig regels schrijft dat bijvoorbeeld vuil een donkere vlek is, kijkt het model zelf naar vele voorbeelden en ontdekt zelf de onderliggende patronen. Deze patronen worden opgeslagen als een soort digitale vingerafdruk, een 'embedding'. Deze techniek is heel goed te gebruiken voor taken zoals het zoeken en terugvinden van vergelijkbare afbeeldingen. Door de geometrische structuur van de data te leren of feedback van gebruikers te gebruiken, kan het model steeds betere en efficiëntere voorstellingen maken, zelfs voor dingen die het tijdens de training nog nooit heeft gezien.

(Maria Tzelepi, 2022)

Hierop volgt de werking van “feature extraction” tegenover “End to end learning”.

Feature extraction is een belangrijke stap in machine learning. Het is het proces waarbij je de nuttige, karakteristieke informatie uit ruwe data haalt, zoals een afbeelding. Stel je hebt een foto van een zonnepaneel, daar zit ontzettend veel informatie in, maar niet alles is relevant om bijvoorbeeld vuil te spotten. Feature extraction zet die ruwe data om in een reeks specifieke, meetbare kenmerken, zoals textuurpatronen in een afbeelding. Voor deze stap is vaak veel wiskundige analyse nodig. Het doel is om een compacte, machine leesbare “vingerafdruk” van de data te maken die het belangrijkste opvangt, zodat het model er daarna beter en sneller mee kan werken. Het is dus niet zomaar data verwerken, maar echt de verborgen, karakteristieke informatie analyseren die nodig is voor de specifieke taak. (M. Menagadevi, 2024)

“End to end learning” is juist wat we zien bij deep learning. In plaats van dat je als programmeur eerst zelf handmatig de belangrijke kenmerken oftewel features, uit ruwe data moet halen, zoals randen of hoeken in een afbeelding en die pas daarna aan een model geeft, doet een deep learning model dit alles in één keer. Het model krijgt de ruwe data, bijvoorbeeld de ruwe pixels van een foto en leert zelf, via zijn verschillende lagen, welke features het belangrijkste zijn voor de taak. De eerste lagen leren simpele dingen zoals randen, en diepere lagen combineren die tot complexe patronen en objecten. Omdat het model precies de features leert die nuttig zijn voor het specifieke doel, zoals het herkennen van vuil, is het veel efficiënter en nauwkeuriger dan de oude methode waar je riskeerde dat verkeerde of onnodige features het model in de war brachten.

(Abtin Mahyar, 2025)

## Keuze voor de moderne aanpak

Zoals hierboven beschreven, berust traditionele beeldverwerking met supervised learning dus op een twee-staps proces. Eerst worden via technieken voor feature extraction specifieke, meetbare kenmerken handmatig uit de ruwe data gehaald.

Vervolgens wordt een supervised learning-model getraind met deze vooraf gedefinieerde features om voorspellingen te doen. Deze aanpak is beperkt voor taken met veel variabiliteit, zoals het detecteren van vuil op zonnepanelen. De complexiteit en onvoorspelbaarheid van vuilpatronen, in vorm, textuur en kleur, maken het praktisch onmogelijk om van tevoren de perfecte set handmatige features te definiëren die alle variaties betrouwbaar kan onderscheiden. Hierdoor loopt het model het risico irrelevante of misleidende kenmerken te gebruiken, wat slecht is voor de nauwkeurigheid.

Dit is precies de beperking die wordt opgelost door representational learning en end-to-end learning. In plaats van afhankelijk te zijn van voorgeprogrammeerde regels, leert een model via representational learning zelf de onderliggende, betekenisvolle patronen uit de data, die het opslaat als efficiënte 'embeddings'. End-to-end learning binnen deep learning gaat nog een stap verder: Het elimineert de handmatige feature extractie gedeelte volledig. Het model verwerkt de ruwe pixels en leert geïntegreerd in één training welke features gebaseerd op een hiërarchie van simpele randen tot complexe texturen, het belangrijkste zijn voor de specifieke taak. Door deze geautomatiseerde aanpak kan het model zich aanpassen aan de enorme variatie in echte vuilpatronen, wat leidt tot nauwkeurigere detectie dan met de klassieke methode mogelijk is.

## Covolutionele Neurale Netwerken

Convolutionele Neurale Netwerken, CNN's of ConvNets, zijn een speciaal type neuraal netwerk dat is ontworpen om driedimensionale data, zoals afbeeldingen, te verwerken voor taken als classificatie en objectherkenning. Ze bestaan uit drie hoofdtypen lagen: de convolutionele laag, de poolinglaag en de "fully connected"(FC) laag. In tegenstelling tot oudere methoden, waarbij feature-extractie handmatig was en veel tijd kostte, zijn CNN's een schaalbare aanpak. Ze zijn superieur in prestaties bij het verwerken van beeld-, spraak- of audiogegevens. Hun kracht ligt in het hiërarchisch leren van features, daarom zijn ze zo geschikt voor afbeeldingen.

De kern van een CNN is de convolutionele laag. Hierbij schuift een kleine filter of kernel over de invoerafbeelding. Op elke positie berekent hij een dot product tussen de filterwaarden en de onderliggende pixels, wat resulteert in één getal in een feature map wat ook wel activatiemap wordt genoemd. Dezelfde filter, met dezelfde gewichten wordt over de hele afbeelding gebruikt, wat parameter sharing heet. De eerste lagen van een CNN halen de eenvoudige, laag-niveau features uit zoals kleuren en randen. Hoe

verder de data door de lagen van het netwerk gaat, hoe grotere en complexere vormen het herkent. Uiteindelijk kan het netwerk zo het juiste object herkennen. Dit wordt hiërarchisch feature learning genoemd.

Zoals verwacht is het trainen van een diep CNN van scratch iets wat enorme hoeveelheden gelabelde data en aanzienlijke rekenkracht vereist, vaak met speciale hardware zoals GPUs. Dit is vaak onpraktisch. Daarom zijn pretrained CNN backbones zo gebruikelijk. Dit zijn CNN's, zoals de bekende LeNet-5, die al zijn voorgetraind op gigantische, algemene datasets zoals ImageNet. Tijdens dit voortrainen hebben ze al een rijke bibliotheek van algemene visuele features geleerd, van randen tot complexe texturen. Deze getrainde CNN-backbone kan vervolgens als een krachtig startpunt worden genomen voor een nieuwe, specifiekere taak, zoals vuil detecteren op zonnepanelen. Alleen de laatste lagen hoeven dan te worden aangepast en getraind met een veel kleinere, eigen dataset. Dit maakt "state-of-the-art" computervisie efficiënt en toegankelijk voor de meeste mensen.

(What are convolutional neural networks? , 2025)

## Siamese Neuraal Netwerken

Een Siamese Neurale Netwerk bestaat uit twee of meer precies dezelfde subnetwerken die samen gewichten delen. Het netwerk is speciaal ontworpen voor vergelijking. In plaats van dat het één afbeelding direct in een categorie indeelt, zoals een gewone classifier doet, verwerkt het twee afbeeldingen tegelijk. Beide afbeeldingen gaan door identieke netwerken, die elk een uitvoervector, ook wel een embedding genoemd, maken.

Vervolgens worden deze twee vectoren met elkaar vergeleken om te bepalen hoe sterk ze op elkaar lijken.

Dit heet metric learning of metriekleren. Het netwerk leert niet om namen van categorieën te voorspellen, maar het leert een slimme manier om de onderlinge afstand tussen gegevens te meten. Het doel is om de afbeeldingen zo om te zetten in getallen (embeddings) dat soortgelijke dingen ook op de getallenlijn dicht bij elkaar komen te liggen. Dit gebied waar die getallen in liggen wordt de latente ruimte genoemd. Tijdens de training wordt het netwerk getraind om de afstand tussen bij elkaar horende afbeeldingen klein te maken en de afstand tussen niet-bij elkaar horende afbeeldingen groot.

Dit principe is perfect voor het vinden van vuil op zonnepanelen. Je kunt het netwerk trainen met voorbeelden van schone panelen. Het leert dan hoe een schoon paneel er in getallen uit ziet, en het slaat dat patroon op als een referentie in de latente ruimte. Later, als je een nieuw paneel controleert, maakt het netwerk een embedding van die nieuwe foto. Vervolgens meet het hoe ver deze nieuwe getallenreeks af ligt van de bekende, schone referentie. Vuil wordt gedetecteerd als een afwijking van die schone referentie. Een

grote afstand in de latente ruimte betekent dat het nieuwe paneel afwijkt van wat schoon is, en dus waarschijnlijk vuil bevat. Dit werkt zelfs als je maar een paar voorbeelden van schone panelen hebt om mee te trainen. (Siamese Neural Network , 2021)

## Image Embeddings en Afstandsgebaseerde Detectie

Een embedding vector is een vector met een vaste lengte die een numerieke representatie vormt van de inhoud van een afbeelding. Het is een vorm van dimensionality reduction waarbij hoge-dimensionale pixeldata wordt geprojecteerd naar een lager-dimensionale ruimte die de meest kenmerkende visuele eigenschappen behoudt. Deze vectoren worden gegenereerd door modellen zoals Convolutionele Neurale Netwerken (CNN's), die kenmerken hiërarchisch eruit halen en coderen als punten in een embedding space. (Nwoke, 2023)

In deze ruimte kan de wiskundige afstand tussen twee vectoren geïnterpreteerd worden als hun visuele gelijkheid: vergelijkbare afbeeldingen liggen dicht bij elkaar, terwijl ongelijke objecten ver uit elkaar liggen. (Embeddings: Embedding space and static embeddings, sd)

De gelijkheid tussen de twee embedding vectoren wordt meestal gemeten met Euclidean distance en cosine similarity. De Euclidean distance meet de directe 'rechte lijn' afstand tussen twee punten in de vectorruimte. Vectoren met een kleine Euclidean distance bevinden zich in hetzelfde gebied van de ruimte en hebben een vergelijkbare soort kenmerken. Cosine similarity meet daarentegen de hoek tussen twee vectoren en is daarom ongevoelig voor hun absolute grootte. Het geeft aan of de vectoren in dezelfde algemene richting vanuit de oorsprong wijzen, wat betekent dat ze dezelfde verhouding of patroon van kenmerken hebben, zelfs als de ene veel sterker is uitgedrukt dan de andere. De keuze tussen deze twee metingen hangt af van de taak. (Luca, 2024)

In plaats van een model direct te trainen om klasselabels zoals schoon of vuil te voorspellen, kan een aanpak gebaseerd op gelijkheid worden gebruikt met afstandsrempels. Hierbij wordt eerst een referentie-embedding bepaald, bijvoorbeeld van een schoon zonnepaneel. Voor nieuwe afbeeldingen wordt de afstand tot deze referentie berekend. Blijft deze afstand onder een vooraf ingestelde drempelwaarde, dan wordt de afbeelding als vergelijkbaar met schoon beschouwd. Overschrijdt de afstand de drempel, dan wordt de afbeelding geclassificeerd als afwijkend of vuil. De keuze van de drempel bepaalt de balans tussen precisie en recall: een hogere drempel vermindert fout-positieven, terwijl een lagere drempel zorgt voor hogere gevoeligheid en minder gemiste vervuiling.

(How to use classification threshold to balance precision and recall, 2025)

## Data Preprocessing en Normalisatie

In het domein van machine learning, en specifiek bij neurale netwerken, is het een belangrijk principe om alle inputdata te voorzien van een uniforme schaal. Dit proces heet normalisatie. Het doel is om te voorkomen dat kenmerken met grotere numerieke waarden onevenredig veel invloed krijgen op het leerproces van het model.

Een simpele en veel gebruikte manier om dit te doen is door de waarden om te schalen naar een vast, klein bereik, zoals tussen 0 en 1. Deze vorm van normalisatie, min-max normalisatie genoemd, zorgt ervoor dat alle waarden proportioneel binnen dit vaste kader vallen. Voor afbeeldingsdata betekent dit in de praktijk dat de ruwe pixelwaarden, die bijvoorbeeld kunnen liggen tussen 0 en 255, worden gedeeld door de maximale waarde (255) om ze om te zetten naar het bereik [0, 1].

De reden om deze normalisatie toe te passen heeft te maken met de manier waarop een neuraal netwerk leert. Tijdens het trainen wordt gebruikgemaakt van wiskundige optimalisatie, en dit proces is gevoelig voor de schaal van de invoer. Wanneer de invoerwaarden op een consistente en genormaliseerde schaal staan, kan het netwerk efficiënter en stabielere leren. Zonder deze consistentie zouden heldere pixels met een hoge waarde te veel invloed kunnen krijgen ten opzichte van donkere pixels, wat het leerproces minder betrouwbaar en minder efficiënt maakt. (Jaiswal, 2024)

## Tile-Based Image Analysis voor Geavanceerde Inspectie

Voor de analyse van hoge-resolutiebeelden, zoals opnames van zonnepanelen, is een score voor het hele beeld onvoldoende, omdat het geen informatie geeft over waar een eventueel probleem zich bevindt. Om deze spatiale lokalisatie te bereiken zonder dure pixel-per-pixel segmentatie, wordt tile-based of patch-based analyse toegepast. Hierbij wordt het grote beeld opgedeeld in een raster van kleinere vierkantjes oftewel tiles. (Rajshekhar, 2022)

Deze aanpak maakt gebruik van een sliding window dat over de afbeelding schuift. Om te voorkomen dat objecten of afwijkingen precies op de randen van tiles vallen en worden gemist, worden deze windows vaak overlappend gemaakt, wat de dekking en detectienauwkeurigheid verbetert.

Deze methode sluit aan bij het concept van weakly supervised learning. Hierbij krijgt een afbeelding alleen een algemeen label, bijvoorbeeld vies, maar hoeft niet voor elke pixel aangegeven te worden waar het vuil precies zit, wat volledige segmentatie zou vereisen. Het model leert zelf om binnen de afbeelding de specifieke tiles te identificeren die het sterkst bijdragen aan dat label. (Dogra, 2020)

Voor elke tile kan een model zoals een Convolutionele Neurale Netwerk (CNN) of een Siamese Netwerk een embedding genereren, een vaste lengte vector die de visuele kenmerken samenvat. Door de embedding van een tile te vergelijken met die van een referentiebeeld van een schoon paneel, bijvoorbeeld met een afstandsmaat, ontstaat er een anomaliescore per tile.

Deze plaatsgebonden scores vormen de basis voor een heatmap. Een heatmap is een handige vorm van data visualisatie die numerieke waarden omzet in een overzichtelijk kleurenspectrum, bijvoorbeeld van blauw, laag risico/gelijk aan de referentie, naar rood wat hoog risico/afwijkend voorstelt. Dit maakt de spatiale verdeling (spatial averaging) van het probleem meteen duidelijk.

(What is Heatmap Data Visualization and How to Use It?, 2024)

Door de gekleurde tiles terug te leggen op hun oorspronkelijke positie, ontstaat er een visueel overzicht dat in één oogopslag de locatie van potentiële vuilplekken laat zien. Deze hele aanpak van het opdelen in tiles tot het genereren van een heatmap is precies hoe geavanceerde beeldanalyse uitgevoerd kan worden ondanks de beperkte, zwakke labels.

Dit is verder ook nog van groot belang voor debugging. Een goede heatmap laat niet zomaar willekeurige gekleurde vlekken zien, maar duidelijke, samenhangende patronen die logisch zijn. Als de heatmap er rommelig uitziet of onlogische patronen laat zien, bijvoorbeeld sterk afwijkende scores verspreid over een verder schoon paneel, is dat een duidelijk signaal dat er iets mis kan zijn in het model, de data of de gekozen drempelwaarden. De heatmap werkt dus als een visueel controle-instrument dat helpt de betrouwbaarheid van het hele inspectiesysteem te beoordelen. (Yi, 2025)

## Computer Vision met OpenCV

OpenCV (Open Source Computer Vision Library) is een open-source library voor computer vision en beeldverwerking, met een uitgebreide verzameling algoritmen voor het inlezen, verwerken en analyseren van beelden en video. (Alvi, 2023)

Het proces begint met “image acquisition” (afbeeldingen verkrijgen). OpenCV kan eenvoudig afbeeldingen van een bestand inlezen, maar ook direct frames verkrijgen van een aangesloten camera, wat essentieel is voor real-time inspectiesystemen. Als het eenmaal verkregen is, begint de voorbewerking (preprocessing). Dit zijn de belangrijkste stappen om de ruwe beelddata voor te bereiden voor latere, complexere analyse. Veelvoorkomende bewerkingen zijn het omzetten van kleurafbeeldingen naar grijswaarden (om complexiteit te verminderen) en het toepassen van filters, zoals Gaussian blur, om ruis te onderdrukken en de beeldkwaliteit te verbeteren. (Alvi, 2023)

Om specifieke objecten zoals zonnepanelen te isoleren, worden technieken voor maskering en randdetectie gebruikt. Maskering kan gebaseerd zijn op kleur of helderheid, waarbij pixels die binnen een bepaalde drempelwaarde vallen worden geselecteerd om een binaire maskerafbeelding te creëren.

Voor het vinden van de exacte contouren van een object is randedetectie vaak effectiever. De “Canny edge” detector is een veelgebruikte methode hiervoor binnen OpenCV. Deze detector vindt randen door eerst ruis te verminderen, vervolgens de gradiënt dus de sterkte van helderheidsverandering te berekenen, en tenslotte met een slim algoritme alleen de sterkste en meest samenhangende randen over te houden. Het resultaat is een duidelijk beeld van de randen en contouren van objecten in het plaatje, wat de eerste stap is om een individueel paneel te identificeren en te isoleren voor verdere analyse. (Team, 2021)

## Pad Planning/ Mapping

Pad planning is een fundamenteel probleem in de robotica. Het gaat om het vinden van een pad zonder botsingen voor een robot vanaf zijn startpositie naar een doelpositie. Dit heet ook wel het plannen van een collision-free path. Het ideale, complete algoritme vindt altijd een pad als het bestaat en stopt ook op tijd als dat niet zo is. Omdat dit probleem erg ingewikkeld is voor complexe robots, zijn moderne algoritmen vooral praktisch: ze moeten een goede balans vinden tussen betrouwbaarheid en efficiëntie. Enkele veelgebruikte methoden zijn Artificial Potential Fields, waarbij de robot wordt aangetrokken door het doel en afgestoten door obstakels, en “Rapidly exploring Random Trees” (RRT), een slimme methode die door willekeurige steekproeven een boomstructuur groeit van start naar doel.

(Frank Dellaert, 2023)

## Ultrasonische Sensoren

Ultrasonische sensoren zijn instrumenten die de afstand tot een object meten door gebruik te maken van ultrasonische geluidsgolven. Dit zijn geluidsgolven met een frequentie boven de 20 kHz, die het menselijk oor niet kan horen. De sensor werkt volgens het time-of-flight principe: een zender stuurt een ultrasonische puls uit. Als deze puls een object raakt, kaatst deze terug naar de sensor waar een ontvanger hem opvangt. De sensor meet de tijd tussen het uitzenden en ontvangen. Omdat de geluidssnelheid in lucht bekend is (ongeveer 340 m/s), kan de afstand tot het object worden berekend met de formule:  $\text{afstand} = (\text{geluidssnelheid} * \text{gemeten tijd}) / 2$ . Een groot voordeel van deze techniek is dat deze onafhankelijk werkt van lichtomstandigheden, rook, stof of de kleur van het object. Hierdoor kan de sensor bijvoorbeeld transparante objecten detecteren waar optische sensoren zouden falen. Ze worden daarom vaak ingezet voor taken zoals aanwezigheidsdetectie, afstandsmeting en het voorkomen van botsingen bij robots. Een voorwaarde is wel dat het object het geluid reflecteert.

(Understanding Ultrasonic Sensor, 2023)

## Raspberry pi

Een Raspberry Pi is een compacte en betaalbare single board computer. In tegenstelling tot een microcontroller, zoals een Arduino, is een Raspberry Pi een hele computer die een besturingssysteem kan draaien. Dit betekent dat hij in staat is om meerdere programma's tegelijk uit te voeren, bestanden op te slaan en complexe berekeningen uit te voeren.

Raspberry pi's worden vaak gebruikt in toepassingen waar een normale computer te groot, te duur of te energie intensief is. Door het kleine formaat en het lage stroomverbruik zijn ze geschikt voor gebruik in mobiele systemen, zoals robots, slimme apparaten en meetopstellingen. Tegelijkertijd hebben ze voldoende rekenkracht om beeldverwerking, netwerkcommunicatie en zelfs eenvoudige machine learning taken uit te voeren.

Een belangrijk kenmerk van de raspberry pi is dat hij een algemene invoer en uitvoerpinnen (GPIO) heeft. Hiermee kunnen sensoren, motoren en andere elektronische componenten direct worden aangesloten.

Binnen robotica wordt raspberry pi erg vaak ingezet als centrale besturingseenheid. Hij verwerkt sensorgegevens, neemt beslissingen op basis van programma's en stuurt motoren aan.

Hoewel de raspberry pi niet zo krachtig is als een laptop of desktopcomputer, kan hij toch effectief worden gebruikt voor machine learning toepassingen. Dit gebeurt meestal door

het model niet op de raspberry pi zelf de trainen. Het trainen van een machine learning model heeft veel rekenkracht en tijd nodig, wat beter geschikt is voor een laptop of server.

(about, n.d.)

## Unity Engine

Unity is een veelgebruikte software engine die oorspronkelijk is ontwikkeld voor het maken van videogames, maar tegenwoordig ook wordt toegepast voor simulatie en technische visualisatie. Met Unity kunnen driedimensionale omgevingen worden opgebouwd waarin objecten zich realistisch gedragen door middel van een ingebouwde physics engine.

Programmeren in Unity gebeurt met de programmeertaal C#. (Physics, n.d.)

Unity wordt vaak gebruikt in situaties waarin systemen eerst virtueel getest moeten worden voordat ze in de echte wereld worden toegepast. Door objecten te voorzien van massa, botsingen en beweging kan het gedrag van machines en robots worden nagebootst zonder risico op schade.

Binnen robotica wordt Unity gebruikt om robots en hun omgeving te simuleren. Sensoren zoals camera's en afstandssensoren kunnen virtueel worden gesimuleerd, waardoor algoritmes voor navigatie en beeldverwerking getest kunnen worden. Dit maakt het mogelijk om software te ontwikkelen en te testen voordat deze op echte hardware wordt ingezet. (Unity, n.d.)

Hoewel Unity zelf geen besturingseenheid is, vormt het een belangrijk hulpmiddel bij het ontwerpen en testen van robotsystemen. De resultaten uit simulaties kunnen later worden toegepast op fysieke robots, zoals robots die worden aangestuurd door een Raspberry Pi.

## Hoe kan deze schade op zonnepanelen voorkomen worden?

Zonnepanelen kunnen beschermd worden door een krasbestendig laagje van gehard glas. Maar dat beschermt het niet van alle schade. Zoals de naam al zegt beschermt het, het zonnepaneel alleen voor krassen, en niet voor gewicht of schade aan de hardware. Om te weten hoeveel gewicht er op een zonnepaneel gezet kan worden, kan gekeken worden naar de “mechanical test load” die onderzocht is bij het veldonderzoek. Deze is voor de onderzochte panelen 5400 Pa. Er mag niet aangenomen worden dat alle panelen zo sterk zijn dus er is gekeken naar hoe sterk zonnepanelen van andere merken zijn. De standaard waarmee deze test gedaan wordt is de IEC 61215 waarbij de eerder gevonden 5400 Pa drukkracht de standaard is om te slagen. Het is in de praktijk realistisch om te zeggen dat alle panelen die gebruikt worden op zonneparken, geslaagd zijn voor dit certificaat, omdat het anders voor zonneparken zeer lastig is om een verzekering te krijgen. Er wordt dus aangenomen in dit PWS dat alle panelen waar de robot op komt te staan voldoen aan deze normen. Ook wordt aangenomen dat panelen van boven 1.567m<sup>2</sup> (60 zonnecellen) gebruikt worden op zonneparken waar de robot op komt omdat in de industrie panelen van een kleiner formaat niet vaak voorkomen in de industrie (solar size). Met deze druk van 5400 Pa en een oppervlakte van 1.567m<sup>2</sup> kan berekend worden hoeveel Newton de robot maximaal zou mogen wegen als hij op een standaard paneel rijdt dat precies horizontaal (voor maximale druk) gemonteerd is:

$$F = P * A$$

Met F = kracht in Newton, P = druk in Pascal, A = oppervlakte in m<sup>2</sup>.

En met de gegevens wordt het volgende gekregen:

$$5400 * 1.567 = 8461.8N$$

Dit wordt omgerekend naar een gewicht voor de robot in KG door 8461.8N te delen door te delen door 9.81 (standaard gravitatie op aarde) om op een gewicht van 862.9kg uit te komen.

Bij de grotere formaat panelen:

$$5400 * 2.31(\text{solar size}) = 12474N$$

$$\frac{12474}{9.81} = 1271.6g$$

Als doel is gesteld om de robot 25kg te laten wegen, dus gewicht is voor het beschadigen geen probleem. Ook wordt bij de IEC 61215 test een rubberen zuignap gebruikt dus als de robot wordt gemaakt zodat alleen rubberen rupsbanden het paneeloppervlakte raken dan zijn de materialen ook hetzelfde en dus veilig voor de zonnepanelen. (iectestmachine)

## Welk materiaal is het meest kost efficiënt en bruikbaar

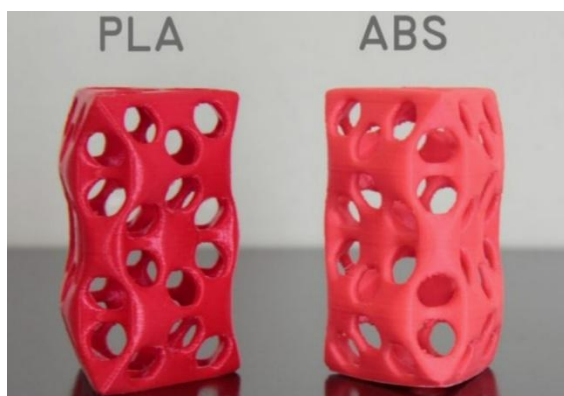
Eerst moet er gevonden worden wat het beste materiaal is voor het maken van een prototype, daar zijn een paar opties voor: 3d printen, hout, karton, rubber, metaal en foam (blogger, 2025) .

Klanten willen waarschijnlijk een goedkoop en sterk materiaal voor het prototype. Ook moet het makkelijk bewerkbaar zijn daarom is ervoor gekozen om 3d print filament gebruiken voor het prototype. Het is makkelijk bewerkbaar, er kan met een 3d bestand precies worden geprint wat er nodig is mits er een goed model is.

Maar welke materialen voor het 3d printen zijn het best? Er is een materiaal nodig dat sterk is en dus niet makkelijk breekt, goedkoop zodat er meerdere keren een prototype gemaakt kan worden en waterdicht is zodat de Raspberry pi niet kapotgaat door waterschade.

Uit onderzoek naar materiaal blijkt uit alle 3d print filamenten, specifiek PLA plastic of PETG-filament het beste is. (Polygenis, 2023). De bron trekt de conclusie dat PETG het beste is voor een prototype bouwen van een robot door zijn hoge smeltpunt van ongeveer 220-260°C. PLA is de goedkopere optie dat makkelijker is om mee te printen en ook sterk is. Het heeft een lager smeltpunt van 200-220°C. Het smeltpunt is niet een probleem voor ons project doordat ons robot nooit op zo een hoge temperatuur gaat komen. (Wickstrom, 2025) De breukvastheid van PLA is 50MPa - 60MPa terwijl die van PETG minder is met 40-50 MPa, maar doordat PETG zelf meer flexibel is kan het technisch gezien meer kracht houden. Wel kan PETG veel beter tegen olie en andere chemicaliën.

Doordat voor dit project niet rekening gehouden hoeft te worden met de temperatuur van de robot, maar alleen de sterkte en prijs is PLA de betere optie voor ons prototype.



Figuur 15 Verschil uiterlijk tussen PLA en ABS



Figuur 16 rol van PLA-filament

# Uitwerking Concept Software

Om de hardware te ondersteunen hebben we daarbij een software nodig. Dit is namelijk de robots “gedachtes”.

De software is gesplits in twee delen. De AI “oog” deel en de C++ “gedachtes” deel.

## C++ “gedachtes” gedeelte

### Bewegingsmodel van de robot

De robot wordt in de simulatie beschreven aan de hand van zijn positie en oriëntatie. Deze worden samengevat in een state vector.

Op ieder gegeven moment is de toestand van de robot te beschrijven als:

$$\begin{cases} x(t) \\ y(t) \\ \theta(t) \end{cases}$$

Waarbij:

1.  $x(t)$  = horizontale positie (cm)
2.  $y(t)$  = verticale positie (cm)
3.  $\theta(\theta)(t)$  = richting (radialen)
4.  $t$  = tijd

De beweging van de robot wordt gemaakt met het unicycle model. Dit model beschrijft een robot die vooruit kan bewegen en kan draaien, maar niet zijwaarts kan glijden (zoals een auto).

De robot heeft twee besturingsinputs:

1.  $v$  = voorwaartse snelheid(cm/s)
2.  $w$  = hoekdraaisnelheid (rad/s)

De bewegingsvergelijkingen zijn:

$$\frac{dx}{dt} = v * \cos \theta$$

$$\frac{dy}{dt} = v * \sin \theta$$

$$\frac{d\theta}{dt} = w$$

## Navigatie en richtingscorrectie

Om naar een doelpunt te bewegen, berekent de robot eerst in welke richting hij moet kijken.

De gewenste stuurhoek is:

$$\theta_{target} = \tan^{-1}\left(\frac{y_t - y}{x_t - x}\right)$$

Het verschil tussen de gewenste richting en de huidige oriëntatie wordt de “heading error” genoemd.

$$e_{\theta} = \theta_{target} - \theta$$

Dit vertelt de robot:

Positief → Naar links draaien

Negatief → Naar rechts draaien

Nul → Niks te veranderen, al in de juiste richting kijkend

## Proportioneel controller (P-Controller)

Om deze fout soepel te corrigeren wordt een proportionele regelaar gebruikt.

De hoeksnelheid is:

$$w = K_p * e_{\theta}$$

Waarbij:

1.  $w$  = draaisnelheid
2.  $K_p$  = proportioneel constante
3.  $e_{\theta}$  = koersfout

Hiermee wordt bedoeld:

1. Als de koersfout groot is → sneller draaien
2. Als de koersfout klein is → Draai langzaam en voorzichtig

Deze regelaar was oorspronkelijk niet aanwezig in de robot code, maar werd toegevoegd nadat tijdens de simulatie bleek dat zonder deze correctie het draaigedrag instabiel werd.

## Mapping en schoonmaakpad (spiraal)

Tijdens de simulatie gaat het systeem ervan uit dat de afmetingen van het zonnepaneel vooraf onbekend zijn.

Het bepaalt de grens door sonar data in vier richtingen te detecteren: omhoog, rechts, omlaag en links.

Op basis hiervan vindt de robot het volgende:

$$x_{min}, x_{max}, y_{min}, y_{max}$$

Deze waarden vormen een rechthoekig gebied:

$$A = (x_{max} - x_{min}) * (y_{max} - y_{min})$$

Waarbij A = totale paneeloppervlakte

Voor het schoonmaken wordt een spiraalvormig pad gebruikt waarbij het werkgebied na elke iteratie kleiner wordt.

De krimpende afstand wordt als volgt gedefinieerd:

$$d = w * (1 - o)$$

Waarbij:

1. w = robot breedte
2. o = overlap verhouding (0.5 = 50%)

De grenzen bij lus k zijn:

$$x_{min}^{(k)} = x_{min} + k * d$$

$$x_{max}^{(k)} = x_{max} - k * d$$

$$y_{min}^{(k)} = y_{min} + k * d$$

$$y_{max}^{(k)} = y_{max} - k * d$$

De spiraal stopt wanneer:

$$x_{min}^{(k)} \geq x_{max}^{(k)} \text{ or } y_{min}^{(k)} \geq y_{max}^{(k)}$$

Oftewel het is in het centrum van het paneel.

## Vuildetectie

Elk camerabeeld wordt geanalyseerd door het neurale netwerk.

Het netwerk geeft een waarschijnlijkheid als uitvoer:

$$P(\text{vuil} \mid \text{afbeelding})$$

Als  $P(\text{vuil}) > r$ , locatie als 'vuil' gemarkeerd

$r$  = drempelwaarde

$P$  = kans

Wanneer deze kans groter is dan een ingestelde drempelwaarde, wordt de locatie als vervuild opgeslagen.

## Schoonmaak route-optimalisatie

*“Nearest neighbor algorithm”*

Na het scannen van het zonnepaneel kan de robot meerdere vuile plekken hebben gedetecteerd. Elke plek wordt beschreven door een coördinaat  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ , ...,  $(x_n, y_n)$ . Waarbij ‘n’ het totale aantal gevonden vuile plekken is.

Stel dat de robot zich momenteel op positie  $(x_i, y_i)$  bevindt. Voor elke gevonden vuile plek wordt de Euclidische afstand tot de huidige positie van de robot berekend:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

De volgende locatie die de robot bezoekt wordt bepaald door de plek te kiezen waarvoor deze afstand minimaal is:

$$\text{Next spot} = \arg \min_j (d_{ij})$$

Min → geeft de kleinste waarde

Arg min → geeft aan welk punt de kleinste waarde oplevert

Nadat een volgorde is gekozen, dus 1->2->3 enzovoort, is de totale afgelegde afstand de som van alle individuele afstanden tussen opeenvolgende punten.

$$L = \sum_{k=1}^{n-1} d_{k,k+1}$$

De robot verplaatst zich vervolgens naar deze locatie, waarna het proces wordt herhaald met de nieuwe positie als startpunt. Op deze manier ontstaat een efficiënte, lokaal optimale schoonmaakroute langs alle gedetecteerde vuile plekken.

### Conclusie C++ “gedachtes” gedeelte

Om de robot gebruikbaar te maken is de software van belangrijk component. Met hulp van eenvoudige wiskundige berekeningen kan aan alle eisen worden voldoen. Dit gedeelte vormt het “brein” van de robot, hier wordt bepaald hoe hij kijkt, beslissingen neemt en uiteindelijk functioneert.

## AI/ML Gedeelte

Binnen ons concept is machine learning het belangrijkste onderdeel van het systeem. De robot kan wel over het paneel bewegen en schoonmaken, maar zonder AI weet hij niet waar het echt nodig is. Het idee is dat de robot niet zomaar alles standaard opnieuw schoonmaakt, maar dat hij zelf kan zien waar vuil zit en daar extra aandacht aan geeft. Hierdoor wordt het hele proces slimmer en efficiënter.

Machine learning zorgt er dus voor dat het systeem kan herkennen wat schoon is en wat afwijkt van schoon. In plaats van vaste regels te gebruiken zoals “donkere vlek = vuil”, leert het model zelf hoe een schoon zonnepaneel eruitziet en kan het afwijkingen daarvan detecteren.

## Hoe het AI systeem werkt

We hebben gekozen voor een combinatie van een Convolutionele Neurale Netwerk (CNN) en een Siamese Neurale Netwerk structuur. Het komt hierop neer: het model zet een afbeelding om in een numerieke representatie, een embedding. In plaats van miljoenen pixelwaarden te vergelijken, vergelijken we alleen deze representaties.

Een Siamese netwerk is geschikt omdat het leert hoe sterk twee afbeeldingen op elkaar lijken. Het model leert dus niet direct “dit is vuil”, maar leert hoe een schoon paneel eruitziet en hoe ver iets daarvan afwijkt. Dat is handig omdat vuil er elke keer anders uit kan zien. Stof, vogelpoep, reflectie of schaduw zien er allemaal anders uit. Met deze methode kan het systeem beter omgaan met variatie.

## Training van het model

Het model is getraind op een laptop die de kracht gebruikt van een GPU, omdat trainen veel rekenkracht kost. We hebben afbeeldingen gebruikt van volledig schone zonnepanelen en van zonnepanelen met verschillende soorten vervuiling. Alle afbeeldingen zijn eerst verkleind naar hetzelfde formaat en genormaliseerd zodat het model consistente invoer krijgt.

Tijdens de training kreeg het netwerk steeds twee afbeeldingen tegelijk te zien. Soms waren dat twee schone panelen, soms een schoon en een vuil paneel. Het model werd zo getraind dat afbeeldingen die op elkaar lijken ook embeddings krijgen die dicht bij elkaar liggen. Afbeeldingen die verschillen, krijgen embeddings die verder uit elkaar liggen.

Het model leert dus niet simpele regels, maar leert een soort afstand in een wiskundige ruimte. Dat zorgt ervoor dat het ook nieuwe soorten vuil kan herkennen, zolang ze maar genoeg afwijken van “schoon”.

## Detectie met embeddings

Wanneer de robot een nieuwe afbeelding maakt, wordt deze eerst geschaald en genormaliseerd. Daarna wordt de afbeelding door het model gehaald en omgezet in een embedding. Deze embedding wordt vergeleken met een referentie-embedding van een schoon zonnepaneel.

De afstand tussen die twee embeddings wordt berekend. Hoe groter de afstand, hoe groter de afwijking van schoon. Deze afstand wordt omgezet in een score tussen 0 en 100. Zo krijgen we een duidelijke maat voor hoe vervuild een bepaald gebied is.

Dit is veel efficiënter dan pixel-per-pixel vergelijken. Een afbeelding kan bijna 200.000 waarden bevatten, terwijl een embedding maar bijvoorbeeld 256 waarden heeft. Dat scheelt enorm in rekestijd en geheugen.

## Tile based analyse

Vuil zit meestal niet over het hele paneel, maar op specifieke plekken. Daarom delen we de afbeelding op in kleinere stukjes, tiles genoemd. Voor elke tile wordt apart een embedding gemaakt en vergeleken met de referentie.

Zo krijgen we per deel van het paneel een score. Deze scores worden omgezet in een heatmap waarin je direct kunt zien waar afwijkingen zitten. Blauwe delen zijn vergelijkbaar met schoon, rode delen wijken sterk af.

Dit is belangrijk omdat de robot zo precies weet waar hij later nog een keer moet schoonmaken. Zonder deze stap zou het systeem alleen weten dat er ergens vuil zit, maar niet waar.

## Gebruik op Raspberry Pi

Na de training wordt het model opgeslagen en overgezet naar de Raspberry Pi. Op de Raspberry Pi wordt het model niet opnieuw getraind, maar alleen gebruikt om nieuwe afbeeldingen te analyseren. Dat heet inference.

Bij het opstarten laadt de Raspberry Pi het model en de referentie-embedding in. Daarna kan het systeem zelfstandig werken. Omdat we embeddings gebruiken in plaats van volledige pixelvergelijking, is de rekenkracht van de Raspberry Pi voldoende.

## Beslissingslogica

De berekende afstand wordt omgezet in een vervuilingsscore. Op basis van die score wordt bepaald of een gebied schoon, licht vervuild of zwaar vervuild is. Deze informatie wordt opgeslagen samen met de coördinaten.

In de derde fase van het systeem kan de robot dan terugkeren naar plekken met een hoge score en deze opnieuw behandelen. Hierdoor hoeft niet het hele paneel opnieuw schoongemaakt te worden, maar alleen de probleemgebieden.

## Conclusie AI/ML Gedeelte

Het AI-gedeelte maakt het verschil tussen een simpele schoonmaakrobot en een slim systeem. Door gebruik te maken van embeddings en een Siamese netwerk kan vervuiling betrouwbaar worden gedetecteerd zonder afhankelijk te zijn van simpele kleurregels of vaste drempels.

De combinatie van CNN-feature extractie, tile-based analyse en implementatie op een Raspberry Pi laat zien dat het concept niet alleen theoretisch klopt, maar ook praktisch uitvoerbaar is. Machine learning wordt hiermee echt geïntegreerd in een autonoom systeem dat efficiënter onderhoud mogelijk maakt.

# Uitwerking Concept Hardware

Er is een uiteindelijke ontwerp gemaakt voor dit project. Een robot dat kan rijden, draaien en schoonmaken door middel van een borstel. Hieronder staat de benodigde elektronica, de stappen van ontwerpen en de fysieke eindproduct uitgelegd.

## PCB

Het ontwerpen van de PCB begon vanuit een probleem, dat de bedrading van het systeem te rommelig werd. Door losse snoeren tussen de Pi, motor drivers en servo's was erg moeilijk om componenten te testen en isoleren. Daarom werd besloten om een custom PCB te ontwerpen waarin alle verbindingen geïntegreerd konden worden.

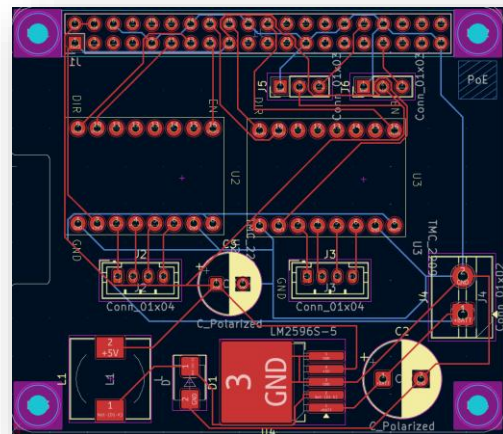
Het eerste prototype werd erg compact ontworpen. Alle componenten (inclusief een ontworpen buck converter), werden op een board geplaatst. Technisch werkte het ontwerp, maar in de praktijk zal er elektrische storing ontstaan.

De oorzaak van de storing is door twee redenen:

Ten eerste werd een eigen buck converter op het board geïntegreerd. Een schakelende buck converter werkt met hoge frequenties en kan ruis (EMI, elektromagnetische interferentie) veroorzaken.

Ten tweede waren de kopertraces (lijnen van koper) te lang. Dit komt omdat er veel omwegen werd genomen omdat de componenten allemaal erg dichtbij elkaar zitten. Langte traces hebben weinig weerstand. Wanneer een raspberry pi 2A stroom trekt, ontstaat er spanningsverlies over de trace. Daardoor ontvangt de Pi geen stabiele 5V meer, wat instabiliteit veroorzaakt.

Met deze gegevens werd aan versie twee begonnen. In deze versie zijn koperbanen veel korter en schoner gerouteerd. De layout was logischer opgebouwd, met scheiding tussen power en signaal. De buck converter was nog steeds geïntegreerd, wat dus nog voor problemen kan zorgen.



Figuur 17 PCB versie 1

Hierna werd het huidige ontwerp naar de expert Alec Li gestuurd voor feedback. Hij wees een belangrijke punt, “trace width”.

Zijn advies was om de power traces, vooral tussen de buck converter en de Pi, minimaal rond de 1mm breed te maken voor een stroom van ongeveer 2A. Dit geldt ook voor de power traces tussen de driver en andere hoog stroming onderdelen.

Hierbij kan een formule worden gebruikt om uit te rekenen hoe dik de traces moet zijn.

$$I = K * \Delta T^{0.44} * (W * H)^{0.725}$$

Hier staat:

I = maximum stroming in A

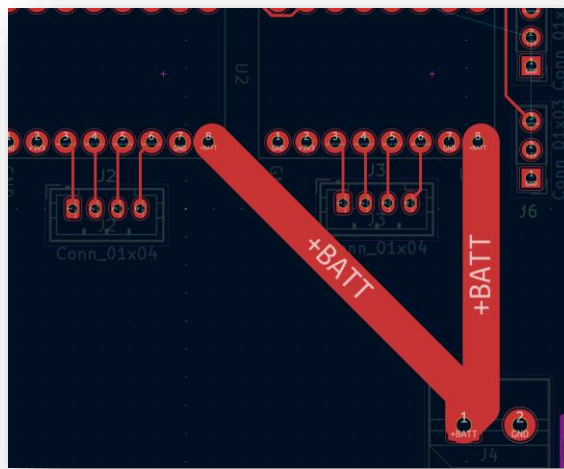
$\Delta T$  = temperatuur

W = dikte in mils

H = Dikte (hoogte) in mils

K = 0.024

Met deze formule kan worden uitgerekend hoe dik de kopertrace moet zijn als we de stroom van 2A heeft. Dit is namelijk 2mm, wat dus veel te dik is om rond de boven te routen.

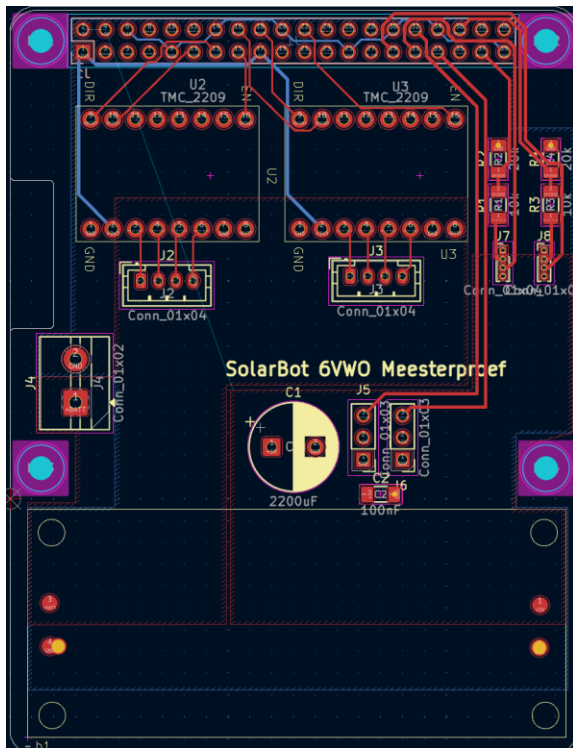


*Figuur 18 Dikke Traces*

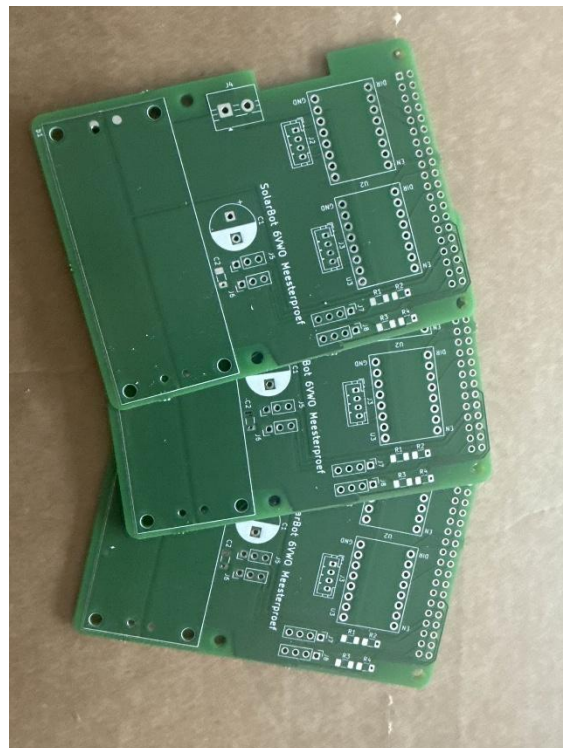
In versie 3 werd een belangrijke keuze gemaakt. De geïntegreerde buck converter is nu verwijderd. In plaats daarvan werd een externe, vooraf geteste buck converter module gebruikt. Hierdoor wordt schakelruis beter geïsoleerd en wordt het ontwerp betrouwbaarder.

Daarnaast werd het powermanagement verbeterd door niet alleen traces te gebruiken, maar ook grotere koperoppervlakken (power planes). Power planes zorgen voor even distributie van stroom en minder spanningsverlies. Ook werd een volledige BOM (bill of materials) gemaakt, zodat het ontwerp reproduceerbaar wordt.

Deze versie van PCB is vervolgens naar een fabrikant in Duitsland gestuurd (AISLER) en na ontvangst van het PCB werd de componenten er op gesoldeerd.



Figuur 20 Eind PCB



Figuur 19 Eind PCB

## Robot

Er wordt een robot ontworpen als eindproduct, daarvoor is ook een model nodig waar te zien is hoe alles in elkaar zit. Dit model moet ook uiteindelijk uitgeprint worden op de 3d printer zodat de robot ook in elkaar gezet kan worden.

De uiteindelijke robot ontwerp moet aan de eisen van ons opdrachtgever voldoen. De eisen die opgesteld zijn door de opdrachtgever zijn:

### Functionele eisen (opdrachtgever)

6. De robot moet kunnen werken zonder rails of een extern geleidingssysteem.
7. De robot moet stabiel kunnen bewegen op hellende of oneffen oppervlakken.
8. De robot moet zelfstandig kunnen navigeren en voorkomen dat hij van de panelen afvalt.
9. De robot moet verschillende soorten vervuiling kunnen verwijderen, zoals stof, pollen, aangekoekte vervuiling in kieren en vogelpoep.
10. De robot moet na het opstarten zo min mogelijk menselijke controle of begeleiding nodig hebben.

Verder moeten de robot ontwerp ook aan de volgende eisen voldoen die wij zelf hebben opgesteld als prestatie eisen:

### Prestatie-eisen

5. De robot mag max 35kg wegen (om schade aan panelen en frames te voorkomen). (Asewe, 2023)
6. De robot moet een voldoende lange gebruiksduur per acculading hebben om een aanzienlijk deel van het zonnepark te reinigen, 30 m<sup>2</sup> per lading als robot op accu rijdt.
7. De robot moet het vermogen van vervuilde zonnepanelen kunnen herstellen met een meetbare efficiëntiewinst. Na reiniging moet de robot minimaal 80% van het rendementsverlies terugwinnen.
8. De robot moet bestand zijn tegen warme omgevingen van boven 40°C, de robot moet ook compleet waterdicht zijn.

Ook zijn er extra eisen waar het ontwerp aan voldoen, dat zijn deze:

Eisen rondom onderhoud en duurzaamheid

3. De robot moet zodanig ontworpen zijn dat minimaal 80% van de onderdelen zonder speciaal gereedschap binnen 30 minuten vervangen kan worden door één persoon met basis technische kennis.
4. De robot mag in volledig operationele toestand (inclusief watertank) niet meer dan 40 kg wegen, en het waterverbruik per schoonmaakcyclus mag maximaal 20 liter per 10 m<sup>2</sup> zonnepanelen bedragen.

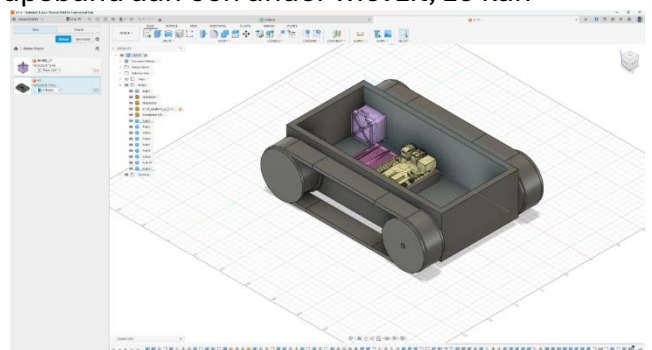
Optionele of toekomstige uitbreidingen

5. Mogelijkheid voor afstandsmonitoring of statusrapportage via een netwerkverbinding.
6. Mogelijkheid voor automatisch opladen via een dockingstation.
7. Ondersteuning voor samenwerking tussen meerdere robots.
8. De robot moet volledig waterloos kunnen reinigen

Daarnaast moet het ontwerp zijn eigen ontwerp eisen ook aan voldoen zodat de robot ook daadwerkelijk kan werken dat zijn de volgende Ontwerp eisen:

1. De motoren moeten vastzitten en kunnen draaien zonder dat iets het blokkeert.
2. De Raspberry pi moet bij de breadboard kunnen zonder onderbreking.
3. Het ontwerp moet makkelijk 3d te printen zijn.

Fusion 360 wordt gebruikt als CAD-applicatie. Hierin is een 3d-model geschetst op basis van alle eisen en verzamelde technieken. Om een draai van 90 graden te kunnen maken met maar 2 stepper motoren is gekozen voor een rupsband opstelling, en als startvorm voor de robot is een bak gemaakt waar de Pi en drivers in kunnen en de motors een wiel aan kunnen drijven die via een rupsband aan een ander wiel zit, zo kan er met 2 motoren en een simpel ontwerp een robot gemaakt worden die een draai van 90 graden kan maken. Echter na het printen van dit ontwerp was het zeer uitdagend om een passend materiaal te vinden voor de rupsband. Uiteindelijk bleek een tot vorm geknipte fietsbinnenband de juiste dikte te zijn, maar hier kwam nog een probleem uit voort.

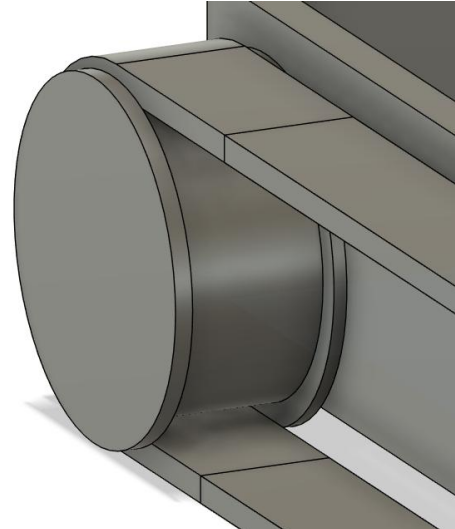


Figuur 21 Prototype ontwerp in Fusion 360

## Iteratief proces

Na dit model te hebben uitgeprint zijn er problemen naar boven gekomen. Het prototype is in elkaar gezet en getest, maar er bleek een probleem te zijn met het rupsband aandrijving systeem.

Het ontwerp zou werken met een wiel met dunne zijwanden zodat de rupsband op zijn plek bleef, maar wel dikker was dan de zijwanden zodat de rupsband de grond raakt. Dit ontwerp bleek niet te werken omdat de band alsnog over de zijwand heen zou klimmen. Hierdoor zou de band van het wiel af vallen of tegen de robot aan zitten en stroef lopen.



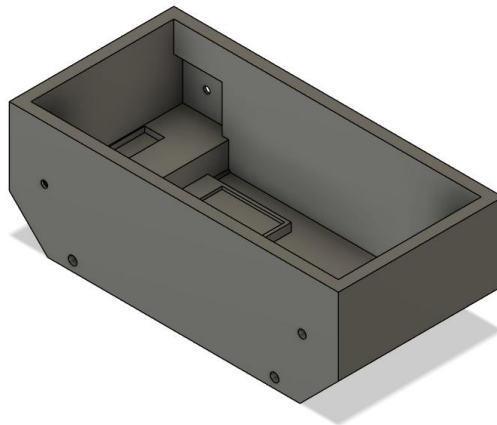
*Figuur 22 Rupsbanden*

Om dit probleem op te lossen is het rupsband systeem opnieuw ontworpen. Hierbij is inspiratie genomen van echte tanks die aandrijfwielen hebben en aan de onderkant wielen die alleen functioneren om de rupsband omlaag te houden zodat het tandwiel van de aandrijfwielen niet beschadigd wordt.



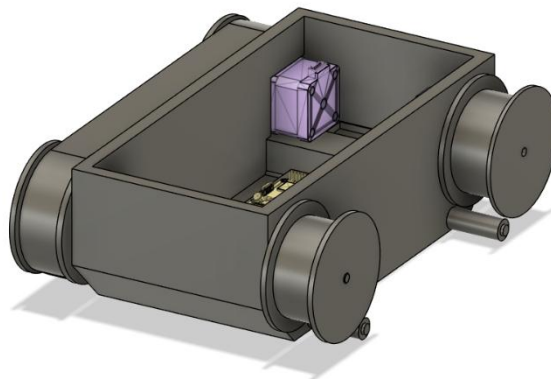
*Figuur 23 Inspiratie tank voorbeeld*

Hier is te zien dat het voorste tandwiel de enige is die de rupsbanden van de tank aandrijft, de rest functioneert alleen om de rupsbanden tegen de grond aan te houden en aan te spannen. Onze robot is hierna opnieuw ontworpen met een gelijksoortig systeem.



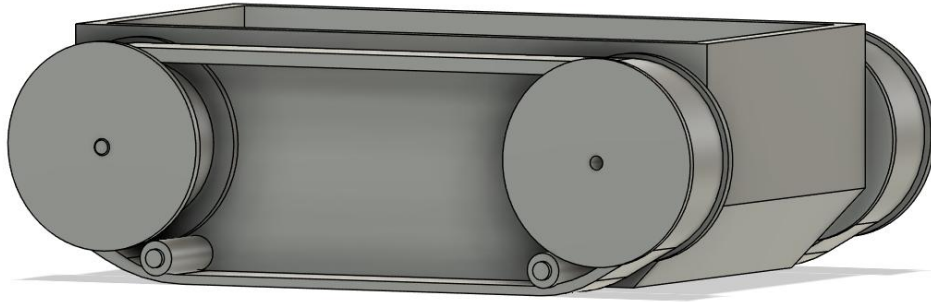
*Figuur 24 nieuw basis voor ontwerp*

Voor de nieuwe versie van de bak waar de elektronica in komt is de bodem verlaagd en zijn er gaten in de onderkant gekomen om aanspannende wielen toe te voegen naast de bestaande wielen, ook is er een gat in de zijmuur gemaakt om de motoren beter op hun plek te houden, en een bakje in het midden waar de Raspberry pi geplaatst wordt.



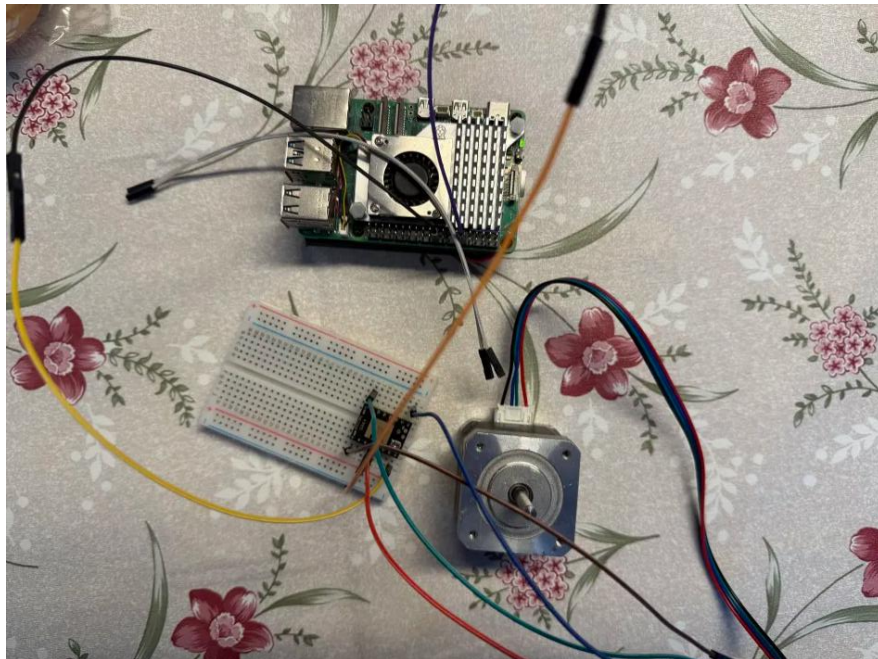
*Figuur 25 Nieuw tank systeem model*

Bij het nieuwe systeem zijn de zijwanden van de wielen aanzienlijk verbreed. De kleinere wieltjes die functioneren om de band aan te spannen en op de grond te houden zijn zo ver mogelijk uit elkaar zonder de andere wielen te raken om de oppervlakte van band op de grond te maximaliseren.



*Figuur 26 Nieuw ontwerp met aangespannen band*

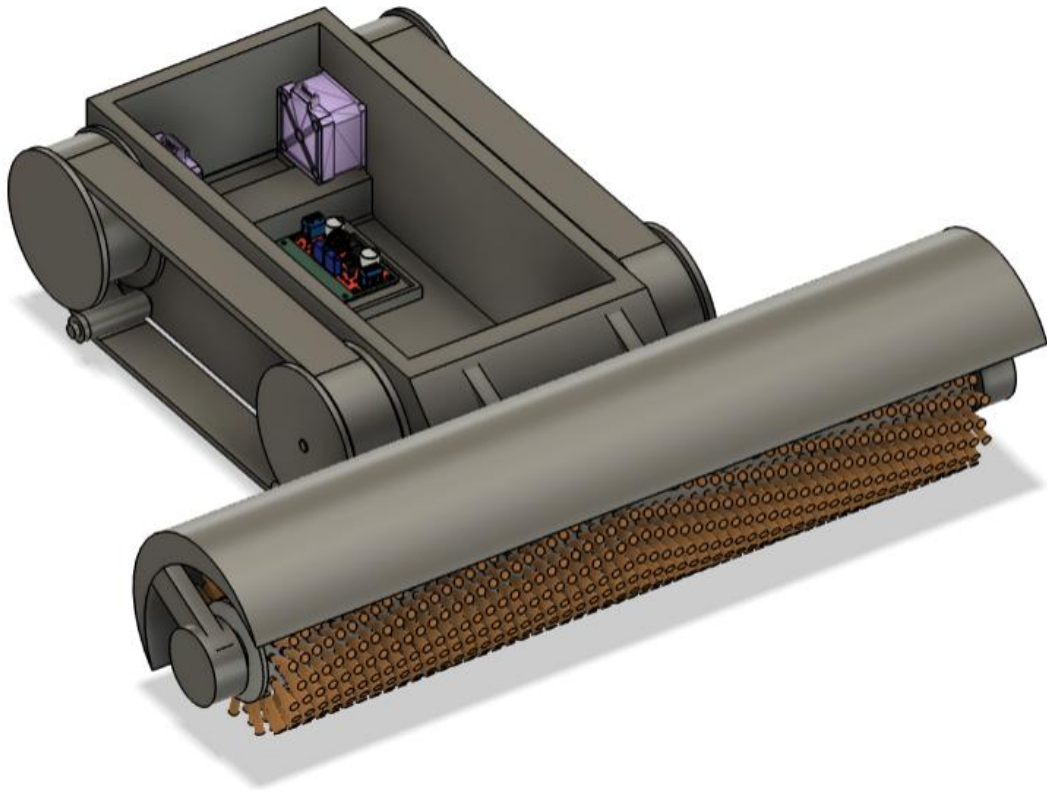
Om het massamiddelpunt zo laag mogelijk te houden voor stabiliteit is het systeem ingesteld zodat de aangespannen band niet ver onder de robot uit steekt. Met dit nieuwe ontwerp is het probleem van een band die over de zijwand van een wiel heen klimt verholpen en kan de robot weer getest worden. In het nieuwe ontwerp is ook ruimte gemaakt voor de motor drivers die ontworpen zijn voor dit project.



*Figuur 27 Een aangesloten stappenmotor aan een Raspberry pi 5 via de driver op een breadboard*

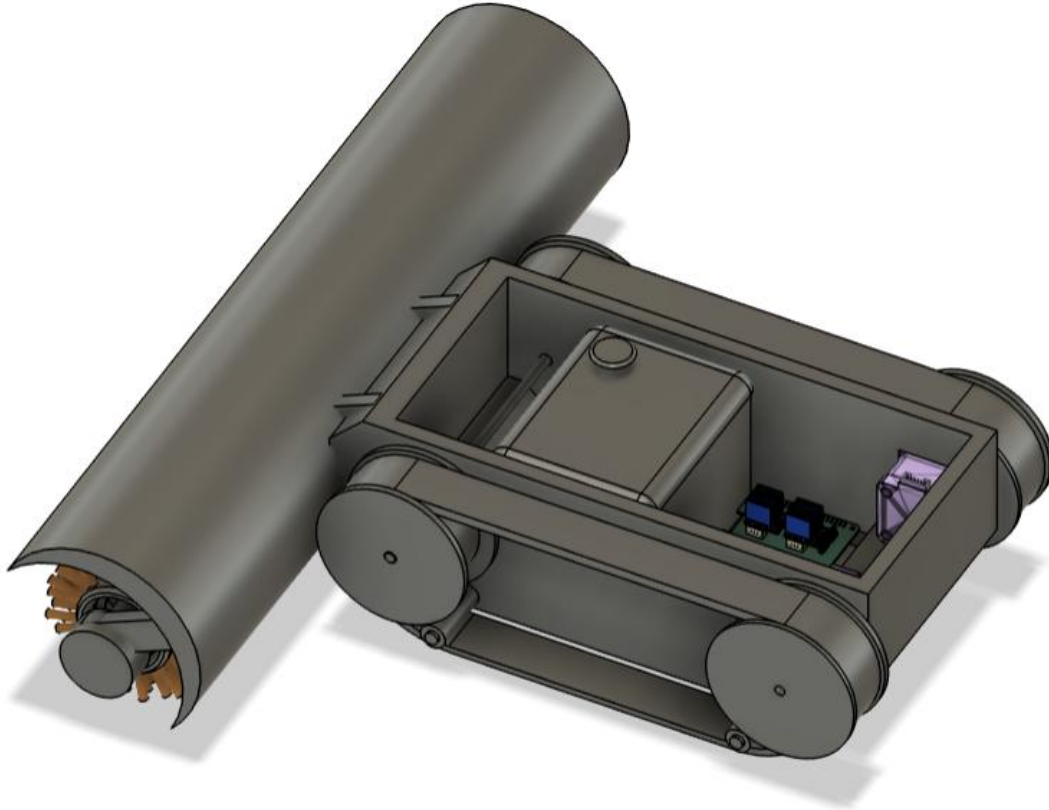
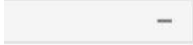
## Borstel en osmosewater systeem

Na een robot gemaakt te hebben die over zonnepanelen heen kan rijden moet er aan het ontwerp een systeem komen met de gekozen technieken om de zonnepanelen schoon te maken. Hiervoor is er gekozen om een borstel te maken en een watertank te maken waar osmosewater uit gespreeid wordt op het oppervlakte van de zonnepanelen. Eerst maken we dit ontwerp in Fusion360.



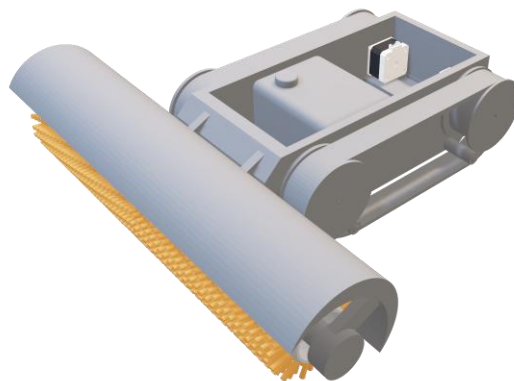
*Figuur 28 Met borstel*

Nu er een ontwerp is voor de borstel kunnen we ruimte vrijzetten voor elektronica binnen de robot, voor de sproeier en de draaiende borstel wordt gebruik gemaakt van gemodificeerde servo motoren die na de modificatie bestuurbaar zijn met de pi 5 via code en niet een limiet hebben van 720 graden maar gewoon door kunnen draaien.



*Figuur 29 Bovenkant*

Naast de elektronica past nog een watertank voor het osmosewater in het ontwerp met een inhoud van 691.2mL. Hiermee zou de robot meerdere panelen per opvulling schoon kunnen maken en als hij een station heeft om meer water te krijgen zou hij (sinds hij met de pi 5 via wifi van afstand bestuurbaar is) zelf weer met water gevuld kunnen worden zonder menselijke interactie.



*Figuur 30 Solarbot prototype (viewable 3d model)*

## Batterij

Voor het bepalen van de juiste batterij moet eerst worden vastgesteld hoe lang de robot autonoom moet kunnen rijden om te voldoen aan de prestatie-eis van 30 m<sup>2</sup> panelen oppervlak:

Om de benodigde rijtijd te bepalen, is eerst de gemiddelde snelheid van de robot experimenteel gemeten. De robot reed hierbij over een rechte afstand van 1 meter op een vlakke ondergrond, waarbij de tijd werd gemeten met een stopwatch. Uit meerdere metingen volgde een gemiddelde snelheid van:

6 cm/s

Deze waarde wordt in de verdere berekeningen gebruikt.

De robot heeft een effectieve schoonmaakbreedte van ongeveer 20 cm. In een ideale situatie zou de oppervlaktecapaciteit dan gelijk zijn aan:

$$20 * 6 = 120 \text{ cm}^2/\text{s}$$

Dit betekent dat de robot theoretisch 120 cm<sup>2</sup> per seconde kan afleggen.

In de praktijk is de situatie echter niet ideaal. De robot moet draaien aan het einde van banen, ook is er soms lichte overlap tussen rijstroken en er is tijdverlies door kleine correcties in richting. Daarom wordt een efficiëntiefactor van 75% aangenomen, wat definitief hoger ligt in de werkelijkheid, maar voor de zekerheid, zorgt dit voor een ruime veiligheidsmarge.

De effectieve oppervlaktecapaciteit wordt dan:

$$0.75 * 120 = 90 \text{ cm}^2/\text{s}$$

De totale te reinigen oppervlakte is:

$$30 \text{ m}^2 = 300000 \text{ cm}^2$$

De benodigde tijd wordt vervolgens berekend door de totale oppervlakte te delen door de effectieve oppervlaktecapaciteit:

$$\frac{300000 \text{ cm}^2}{90 \text{ cm}^2} = 3333 \text{ seconden} = 55 \text{ minuten}$$

De robot moet dus minimaal 55 minuten autonoom kunnen functioneren om aan de prestatie-eis te voldoen.

Vervolgens wordt bepaald of de gekozen batterij de robot gedurende deze tijd van voldoende energie kan voorzien. Hiervoor wordt het totale vermogen van het systeem berekend met behulp van de formule:

$$P = I * V$$

Met:

P = Power (Watt)

I = Stroom (Ampere)

V = Spanning (Volt)

De raspberry pi 5 verbruikt gemiddeld ongeveer 10W. Met een extra camera die circa 1W verbruikt, komt het totale gemiddelde uit op ongeveer 11W. (Raspberry Pi 4 vs 5 Benchmarks, n.d.)

De TMC2209 driver levert ongeveer 2A aan een NEMA 17 stepper motor die 2.55V verbruikt. Met de formule hierboven krijgen we als resultaat: 5.1W en omdat er twee motoren worden gebruikt, is het totale vermogen 10.2W. (NEMA17 stappenmotor, n.d.)

Het totale vermogen dat wordt verbruikt is dus 11W + 10.2W= 21.2W

De batterij die wordt gebruikt heeft een capaciteit van 2250mAh bij 11,1V, dit kan worden omgerekend naar wattuur met de volgende formule:

$$Wh = \frac{mAh * V}{1000}$$

Dit komt neer op ongeveer 25Wh. Omdat voor de veiligheid alleen 80 procent van de batterijcapaciteit wordt gebruikt, blijft er ongeveer 20Wh bruikbare energie over.

Hiermee kan worden berekend hoe lang de batterij de robot kan voeden.

$$Tijd(uren) = \frac{Energy (Wh)}{Power (W)}$$

Met de ingevulde gegevens:

$$\frac{20Wh}{21,2W} = 0.9434uur$$

Dit komt overeen met ongeveer 56,6 minuten, wat ruim genoeg is aangezien er met veiligheidsmarge, 55 minuten nodig was volgens de eisen.

# Eindproduct

## Simulaties

Om zonder de echte hardware de software te testen, werd drie simulaties gemaakt met het concept van de software voor de robot.

### Simulatie Versie 1

In de eerste versie van de simulatie lag de focus vooral op het testen van de spiraalnavigatie en de mapping van de robot. Hiervoor werd een eenvoudig C++-programma geschreven waarin de formules voor coördinaten en spiraalnavigatie binnen vaste grenzen werden getest.

R = Robot

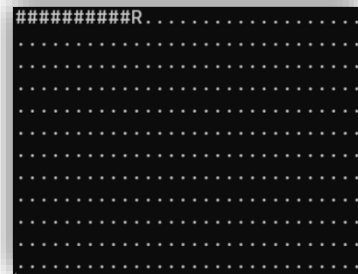
# = Mapping

\* = Navigatie

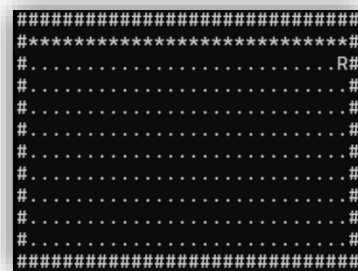
### Simulatie Versie 2

In de tweede versie is het bewegingsmodel en het mapping-concept uitgewerkt in Unity tot een werkende simulatie. Hierbij zijn de eerder beschreven formules gebruikt om het bewegingsgedrag van de robot realistischer te maken, samen met een gesimuleerde sensor. In figuur 15 is de kubus boven de simulatie-robot te zien, met daaronder het zonnepaneel. Voor en links van de robot is een object geplaatst dat continu naar beneden controleert of er nog grond aanwezig is. Wanneer er geen grond wordt gedetecteerd, kleurt de sensor rood. Als er wel grond wordt gedetecteerd, kleurt de sensor groen.

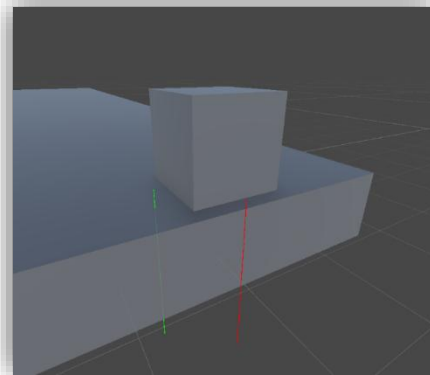
Wanneer aan de voorkant nog grond wordt gedetecteerd terwijl aan de linkerkant geen grond meer is, betekent dit dat de robot zich op een rand bevindt. In dat geval volgt de robot deze rand totdat er ook aan de voorkant geen grond meer wordt gedetecteerd. Vervolgens draait de robot 90 graden naar rechts, noteert hij de coördinaten van zijn



Figuur 31 Simulatie V1 mappen



Figuur 32 Simulatie V1 Navigatie



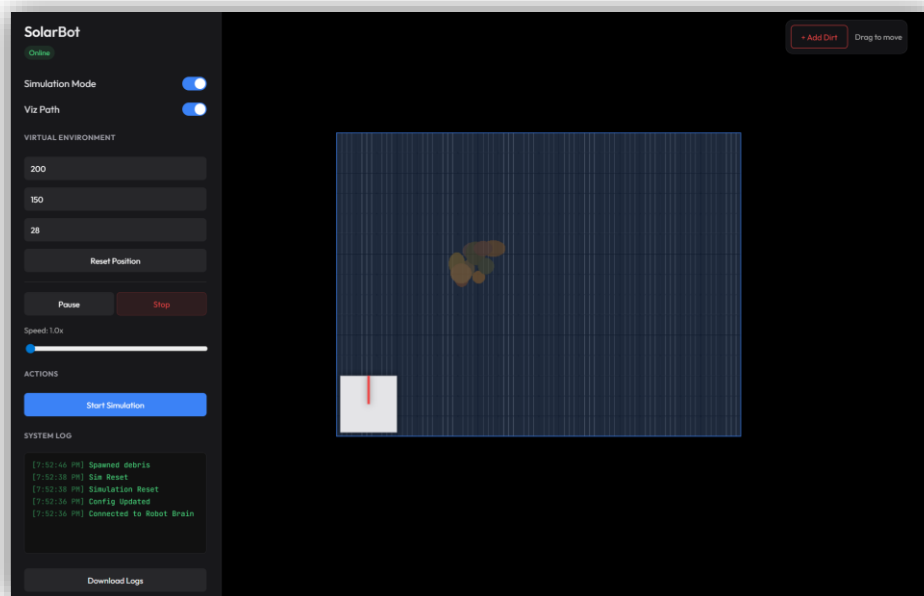
Figuur 33 Simulatie V2 Sensoren

huidige positie en herhaalt hij dit proces. Dit gaat door totdat alle vier de hoeken zijn gevonden. Deze coördinaten worden opgeslagen in een JSON-bestand, zodat de navigatiecode hier later een schoonmaak pad op kan baseren.

## Simulatie eind versie

In de laatste simulatie wordt alles samengebracht en wordt het machine learning-model daadwerkelijk gebruikt. Hiervoor is een Python-script geschreven waarin alle eerder beschreven functies zijn gecombineerd, samen met de code die het ML-model toepast. In deze simulatie wordt een TensorFlow Lite-bestand ingelezen en gebruikt om vervuiling te detecteren op basis van camerabeelden.

In deze uitgebreide simulatie is het mogelijk om zelf te kiezen hoe groot het zonnepaneel is en waar het vuil zich bevindt. Daarnaast kunnen zowel het pad dat de robot daadwerkelijk aflegt als het geplande navigatie pad worden weergegeven. Ook het uiteindelijke “nearest neighbour”-systeem, dat bepaalt in welke volgorde de vuile plekken worden schoongemaakt, is zichtbaar.



Figuur 34 Simulatie V3

## C++ bewegingssoftware

Nadat de simulatie het concept had bevestigd, werd er van de simulatie een echte werkende c++ code gemaakt voor het hardware. Hier werd simpel python code vervangdt door C++ raspberry pi motor code en OpenCV camera vision code.

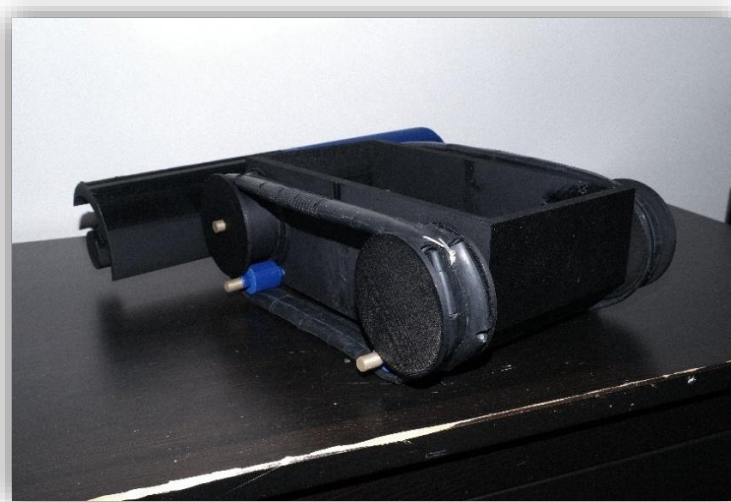
## Robot

Met het ontwerp en de software afgerond, kan de uiteindelijke robot in elkaar worden gezet. Daar is ook hardware voor nodig, de fysieke onderdelen van de robot. Hiervoor is al veel over vertelt, er is hier een lijstje van alle onderdelen waar de robot uit bestaat:

1. Zelfontworpen PCB
2. 2 Drivers
3. Raspberry Pi 5
4. 2 Stappen motoren
5. Borstel
6. 3d print behuizing

Het ontworpen model wordt eerst met een 3D printer geprint. Vervolgens worden de elektronische componenten en de batterij in de behuizing geplaatst. De elektronica bestaat uit, een zelfontworpen PCB, een breadboard met drivers voor de motoren, twee stappen motoren, een raspberry Pi 5 en een batterij als energiebron.

De motoren worden dan aangesloten met een snoer aan de PCB, en worden op hun plaats gezet om de wielen vast te maken. De PCB reguleert stroom en commando's naar de driver dat vervolgens de rupsbanden laat voortbewegen. De raspberry Pi kan beschouwd worden als de hersenen van de robot. De Pi gebruikt machine learning om de uiteindelijke robot aan te sturen. Als laatst worden de rupsbanden om de wielen gezet. De borstel komt dan in de borstel houder dat op de behuizing zit. Met alles bij elkaar kan de robot eindelijk getest worden.



*Figuur 35 Eindproduct*



*Figuur 36 Eindproduct zijkant*



*Figuur 37 Eindproduct vooraanzicht*

## Eindconclusie

In dit project hebben wij onderzocht hoe zonnepanelen efficiënter schoongemaakt kunnen worden met behulp van een autonome robot die gebruikmaakt van machine learning. Het oorspronkelijke probleem was dat zonnepanelen in grote zonneparken vervuilen door stof, zand en vogeluitwerpselen, waardoor het rendement afneemt. Handmatig schoonmaken kost veel tijd, geld en arbeid. Daarom hebben wij een systeem ontworpen dat zelfstandig een zonnepaneel kan scannen, analyseren en gericht schoonmaken.

Het concept is stap voor stap opgebouwd. Eerst hebben we het bewegingsmodel van de robot uitgewerkt op basis van het unicycle model, waarbij de positie wordt beschreven met een state vector en de beweging wordt bepaald door een voorwaartse snelheid en een hoeksnelheid. Met behulp van een proportionele regelaar wordt de koersfout gecorrigeerd zodat de robot stabiel naar een doelpunt kan bewegen. Dit was nodig omdat tijdens de eerste simulaties bleek dat het draaigedrag anders instabiel werd.

Voor het scannen van het paneel hebben we een mapping-systeem ontwikkeld. In de simulatie bepaalt de robot eerst de grenzen van het zonnepaneel met behulp van sensordata, waarna het oppervlak als rechthoek wordt berekend. Vervolgens rijdt de robot een spiraalvormig pad waarbij het werkgebied steeds kleiner wordt totdat het midden is bereikt. Hierdoor wordt het hele paneel systematisch gescand zonder stukken over te slaan.

Het belangrijkste onderdeel van het systeem is het AI/ML-gedeelte. In plaats van simpele kleurregels hebben wij gekozen voor een neurale netwerk dat een waarschijnlijkheid berekent voor vervuiling op basis van een afbeelding. Tijdens de eindversie van de simulatie wordt een TensorFlow model gebruikt om daadwerkelijk vervuiling te detecteren. Het model draait op een Raspberry Pi, waar het alleen wordt gebruikt voor inference en niet opnieuw wordt getraind. Door gebruik te maken van embeddings en afstandsvergelijking kan het systeem afwijkingen van een schoon referentiebeeld detecteren zonder afhankelijk te zijn van vaste drempelregels.

Wanneer vervuilde plekken zijn gevonden, slaat de robot de coördinaten op. Daarna wordt een nearest neighbor algoritme gebruikt om een efficiënte schoonmaakroute te bepalen langs alle gedetecteerde plekken. Dit zorgt ervoor dat de robot niet willekeurig rijdt, maar een logisch en relatief kort pad volgt.

Naast software hebben we ook gekeken naar de hardware. Tijdens het ontwerpen van de PCB bleek dat een te compact ontwerp en lange kopertraces spanningsverlies en instabiliteit konden veroorzaken. Hierdoor hebben we geleerd dat theoretisch werkende

schema's in de praktijk nog steeds problemen kunnen geven door elektromagnetische storing en stroomverliezen. Dit was een belangrijke technische les binnen het project.

Als we alles samennemen, kunnen we concluderen dat het concept technisch haalbaar is. De simulaties laten zien dat navigatie, mapping, AI-detectie en route-optimalisatie samen kunnen werken in één systeem. Natuurlijk is dit nog een prototype en zijn er beperkingen, zoals rekenkracht, weersomstandigheden en schaalbaarheid naar echte zonneparken. Maar als proof of concept laat het project zien dat een intelligente schoonmaakrobot voor zonnepanelen realistisch is.

Wat dit project vooral heeft laten zien, is dat een goed werkend systeem niet alleen draait om één onderdeel zoals AI. Het is juist de combinatie van wiskunde, programmeren, machine learning, elektronica en testen die ervoor zorgt dat het geheel klopt. Door meerdere simulatieversies te bouwen en telkens te verbeteren, hebben we het concept steeds realistischer gemaakt.

Uiteindelijk bewijst SolarML dat machine learning niet alleen iets theoretisch is, maar daadwerkelijk geïntegreerd kan worden in een autonoom robotsysteem om een praktisch probleem op te lossen. Daarmee sluit het project goed aan bij de energietransitie en laat het zien hoe technologie kan bijdragen aan efficiënter onderhoud van duurzame energiebronnen.

Uiteindelijk met de onderzoek die gedaan is, en de fouten die gemaakt zijn is er een finaal ontwerp gekomen. Dit ontwerp moet werken op de meerderheid van de zonneparken. Ons product is bestemd voor zonneparken die niet makkelijk bereikbaar zijn met voertuigen of andere manieren van transport.

# Logboek

Datum	Kevin Kang	Uren	Jivraj Singh	Uren	Xinxin Wang	Uren	Jono Wink	Uren
14-Sep	onderzoek AI/ML	1	onderzoek AI/ML	5	onderzoek zonnepaneel	1	onderzoek zonnepaneel	1
15-Sep	onderzoek AI/ML	2	onderzoek AI/ML	4	onderzoek bestaande robots	2	onderzoek bestaande robots	2
22-Sep	python toepassing onderzoeken	2	python toepassing onderzoeken	2	onderzoek bestaande robots	3	onderzoek bestaande robots	2
28-Sep	c++ toepassing onderzoeken	2	python basiskennis leren	2	onderzoek zonnepaneel	2	onderzoek zonnepaneel	2
3-Oct	project voorstel inleveren		project voorstel inleveren		project voorstel inleveren		project voorstel inleveren	
5-Oct	python project starten	2	Python OOP projecten bouwen	1.5			onderzoek CAD	2
7-Oct			toetsweek		toetsweek			
8-Oct								
14-Oct								
15-Oct								
21-Oct								
22-Oct								
28-Oct	veldonderzoek	4	veldonderzoek	4	veldonderzoek	4	veldonderzoek	4
29-Oct	veldonderzoek resultaat ordenen	1.5	veldonderzoek resultaat ordenen	1.5	veldonderzoek resultaat ordenen	1.5	veldonderzoek resultaat ordenen	1.5
4-Nov	start project bestanden	0.5	start project bestanden	0.5	start project bestanden	0.5	start project bestanden	0.5
5-Nov	onderzoek zonnepaneel	2	start onderzoek AI/ML	5	onderzoek zonnepaneel	2	3d ontwerpen leren	3
11-Nov	start onderzoek software	3	onderzoek AI/ML	6	start onderzoek hardware	3	3d ontwerpen leren	2
12-Nov	concept bedenken	2	Data verzamelen voor dataset om ML-model te trainen.	2	motoren onderzoek	1.5	motoren onderzoek	1.5

18-Nov	Data verzamelen voor dataset om ML-model te trainen.	2	Data verzamelen voor dataset om ML-model te trainen.	2	elektronisch componenten (drivers) onderzoek	2	elektronisch componenten (drivers) onderzoek	2
19-Nov	Data verzamelen voor dataset om ML-model te trainen.	2	Data verzamelen voor dataset om ML-model te trainen.	2	Data verzamelen voor dataset om ML-model te trainen.	2	Data verzamelen voor dataset om ML-model te trainen.	2
21-Nov	Concept onderzoeken	3	Numpy leren en gebruiken	2	Concept onderzoeken	2	Concept onderzoeken	2
22-Nov	Concept onderzoeken	2	Pandas leren en gebruiken	1.5	Concept onderzoeken	3	fusion 360 leren	4
25-Nov	Bewegingsmodel wiskunde schrijven	2	tensorflow gebruiken	5	fusion 360 leren	4	fusion 360 leren	3
26-Nov	bewegingsmodel wiskunde schrijven	2	tensorflow gebruiken	4	fusion 360 leren	2	oefen ontwerp fusion 360	2
28-Nov	reflectie fase 1 inleveren		reflectie fase 1 inleveren		reflectie fase 1 inleveren		reflectie fase 1 inleveren	
2-Dec	elektronica overleggen met teamgenoot	2	elektronica overleggen met teamgenoot	2	elektronica overleggen met teamgenoot	2	elektronica overleggen met teamgenoot	2
3-Dec	robot software c++ schrijven	2	Siamese neural network voorkennis leren	3	elektronica overlegging finaliseren en listen	1	elektronica overlegging finaliseren en listen	1
4-Dec	robot software c++ schrijven	1	onderzoek AI/ML	4	circuit voor robot maken en designen	3	prototype 1 3d model starten	4
5-Dec	pcb design	3	AI/ML concept	2	circuit naar pcb communiceren	2	prototype 1 3d model maken	3
6-Dec	pcb design	2	gemaakte werk uitleggen voor eindverslag en pws	3	python leren voor robot prototypen	2	prototype 1 3d model maken	3
9-Dec			toetsweek		toetsweek			

10-Dec								
16-Dec								
17-Dec								
19-Dec	Bewegingsmodel code schrijven	4	Converter programmeren	2	fusion 360 leren	3	prototype 1 3d model finaliseren	2
21-Dec	Bewegingsmodel code schrijven	3	AI/ML model trainen	4	fusion 360 leren	2	python leren voor robot prototypen	3
22-Dec	Bewegingsmodel code schrijven	5	AI/ML model trainen	4	python leren voor robot prototypen	5	python leren voor robot prototypen	2
23-Dec	robot software c++ schrijven	4	AI/ML model trainen	4	python leren voor robot prototypen	4	python leren voor robot prototypen	6
24-Dec	robot software c++ schrijven	6	AI/ML model trainen	4	Simpel code schrijven voor circuit	4	code helpen voor circuit	3
26-Dec	pcb design	7	AI/ML model testen	2	pcb helpen	4	prototype circuit finaliseren	2
27-Dec	pcb design	3	AI/ML model testen	1	pcb helpen	3	prototype circuit finaliseren	3
28-Dec	pcb design	6	AI/ML model testen	3	pcb helpen	3	prototype 1 circuit testen	4
29-Dec	pcb design	5	robot software c++ helpen	2	prototype 1 helpen	2	prototype 1 circuit testen	3
30-Dec	robot software c++ schrijven	6	robot software c++ debuggen	1	prototype 1 helpen	2	prototype 1 circuit testen	4
31-Dec	Simulatie versie 1 maken	3	gemaakte werk uitleggen voor eindverslag en pws	4	prototype 1 helpen	1	prototype 1 circuit testen	1
6-Jan	Simulatie versie 1 maken	1	Encoder programmeren	1	prototype 1 fysiek maken	3	prototype 1 fysiek maken	3
7-Jan	Simulatie versie 2 maken	3	Encoder programmeren	2	prototype 1 fysiek maken	2	prototype 1 fysiek maken	2
13-Jan	Simulatie versie 2 maken	2	AI/ML model met encoder testen	3	prototype 1 repareren	2	prototype 1 repareren	2

14-Jan	Simulatie versie 2 maken	3	AI/ML model met encoder testen	3	gemaakte werk uitleggen voor eindverslag en pws	3	gemaakte werk uitleggen voor eindverslag en pws	3
16-Jan	Simulatie versie 3 maken	0.5	gemaakte werk uitleggen voor eindverslag en pws	3	rupeesbanden op prototypen zetten	1	rupeesbanden op prototypen zetten	1
18-Jan	Simulatie versie 3 maken	6	gemaakte werk uitleggen voor eindverslag en pws	2	gemaakte werk uitleggen voor eindverslag en pws	2	gemaakte werk uitleggen voor eindverslag en pws	2
20-Jan	Simulatie versie 3 maken	3			code testen	2	code testen	2
21-Jan	Eindverslag document starten	0.5	Eindverslag document starten	0.5	eindverslag document starten	0.5	eindverslag document starten	0.5
24-Jan			Eindverslag AI/ML uitleg	4	prototype 2 inelkaar zetten	2	prototype 2 inelkaar zetten	2
27-Jan	Eindverslag Concept schrijven	2	Eindverslag orientatie schrijven	2	prototype 2 testen	2	prototype 2 testen	2
28-Jan	Eindverslag Concept uitwerking software schrijven	1.5	Eindverslag onderzoek schrijven en sorteren	3	Eindverslag onderzoek schrijven	2	Eindverslag onderzoek schrijven	2
30-Jan	PWS inleveren		PWS inleveren		PWS inleveren		PWS inleveren	
3-Feb	prototype ontwerp helpen	1	prototype ontwerp helpen	1	prototype finaliseren	4	prototype finaliseren	4
4-Feb	prototype ontwerp helpen	1	prototype ontwerp helpen	1	prototype finaliseren	3	prototype finaliseren	3
10-Feb	pcb testen	2	robot code sanity check	2	prototype finaliseren	4	prototype finaliseren	4
11-Feb	pcb testen	3	robot code sanity check	2	prototype finaliseren	5	prototype finaliseren	5
12-Feb	PWS-avond		PWS-avond		PWS-avond		PWS-avond	

17-Feb	eindverslag finaliseren	3	eindverslag finaliseren	3	eindverslag finaliseren	3	eindverslag finaliseren	3
18-Feb	eindverslag finaliseren	4	eindverslag finaliseren	3	eindverslag finaliseren	3	eindverslag finaliseren	3
Totaal tijd		133.5		130.5		122		126

## Literatuurlijst

- (n.d.). Retrieved from WLSolarCleaning: [https://www.wlsolarcleaning.com/wls-9yq-ld1-solar-panel-cleaning-drone.html?gad\\_source=1&gad\\_campaignid=23031524878&gbraid=0AAAABBWR7GQJw1b8yl80jC88d\\_Wr\\_8TZe&gclid=Cj0KCQjw9obIBhCAARIsAGHm1mRYKBw85pn58TkEG3onsdkmNxMaTQtEoxhggfad6P0MYAQSBZRZHDEaAn9hEALw\\_wc](https://www.wlsolarcleaning.com/wls-9yq-ld1-solar-panel-cleaning-drone.html?gad_source=1&gad_campaignid=23031524878&gbraid=0AAAABBWR7GQJw1b8yl80jC88d_Wr_8TZe&gclid=Cj0KCQjw9obIBhCAARIsAGHm1mRYKBw85pn58TkEG3onsdkmNxMaTQtEoxhggfad6P0MYAQSBZRZHDEaAn9hEALw_wc)
- (n.d.). Retrieved from <https://met.com/en/mind-the-fyouture/mindthefyouture/solar-panel-size/>
- (n.d.). Retrieved from <https://ht87925057.en.made-in-china.com/product/ljOxVWFckbht/China-IEC-61215-Strength-Testing-Machine-Solar-PV-Module-Mechanical-Load-Tester.html>
- (2025). Retrieved from SolarGlanz: <https://www.solarglanzz.com/nl/toepassingen#:~:text=%20De%20GEKKO%20robot%20doet%20zijn,maar%20tegelijkertijd%20snel%20%C3%A9n%20mobiel%20'&text=Dankzij%20de%20innovatieve%20techniek%20van,Zonnepanelen%20die%20volledig%20schoon%20zijn>
- (2025, October 2). Retrieved from Zoofy: <https://zoofy.nl/blog/voordelen-van-zonnepanelen-reinigen/#:~:text=Waarom%20zonnepanelen%20schoonmaken%20zo%20belangrijk,de%20levensduur%20van%20je%20installatie.>
- A comprehensive review on solar panel cleaning robot technologies.* (2023, January). Retrieved from ResearchGate: [https://www.researchgate.net/publication/370252807\\_A\\_comprehensive\\_review\\_on\\_solar\\_panel\\_cleaning\\_robot\\_technologies](https://www.researchgate.net/publication/370252807_A_comprehensive_review_on_solar_panel_cleaning_robot_technologies)
- Abeni, M. (2025, november 10). *Sunsave*. Retrieved from Sunsave: <https://www.sunsave.energy/solar-panels-advice/solar-energy/what-are-solar-panels-made-of>
- about.* (n.d.). Retrieved from raspberrypi.org: <https://www.raspberrypi.org/about/>
- Abtin Mahyar, H. M. (2025). *Feature extraction and deep learning*. Retrieved from ScienceDirect: <https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780443328183000198>

Alvi, F. (2023, December 13). *Computer Vision and Image Processing: Understanding the Distinction and Interconnection* . Retrieved from OpenCV:  
<https://opencv.org/blog/computer-vision-and-image-processing/>

*Artificial intelligence and machine learning in finance: A bibliometric review*. (2022, October). Retrieved from ScienceDirect:  
<https://www.sciencedirect.com/science/article/abs/pii/S0275531922000344>

Asewe, J. (2023). *medium*. Retrieved from Can I Walk on My Solar Panels?:  
<https://medium.com/@joeasewe/can-i-walk-on-my-solar-panels-a47e9483f2b6>

*astron5*. (n.d.). Retrieved from [https://www.astronergy.com/wp-content/uploads/2023/08/585605ASTRO-N5\\_CHSM72NDGF-BH\\_2278x1134x30\\_EN\\_20250819.pdf](https://www.astronergy.com/wp-content/uploads/2023/08/585605ASTRO-N5_CHSM72NDGF-BH_2278x1134x30_EN_20250819.pdf)

*astronergy*. (n.d.). Retrieved from <https://www.astronergy.com/product-series/astro-n5-595w-bifacial-series-182>

Astronergy. (2025). Retrieved from Astro N5: [https://www.astronergy.com/wp-content/uploads/2023/08/585605ASTRO-N5\\_CHSM72NDGF-BH\\_2278x1134x30\\_EN\\_20250819.pdf](https://www.astronergy.com/wp-content/uploads/2023/08/585605ASTRO-N5_CHSM72NDGF-BH_2278x1134x30_EN_20250819.pdf)

blogger, I. (2025, July 14). *What Materials Should I Use for My Product Prototype?* Retrieved from sparkinnovations: <https://www.sparkinnovations.com/what-materials-should-i-use-for-my-product-prototype/#:~:text=Product%20design%20firms%20typically%20use,Functional%20prototypes%20and%20engineering%20validation>

Cleaning, P. W. (n.d.). *How does a window cleaning robot work?* Retrieved from Pivotal Window Cleaning: <https://pivotalwindowcleaning.com/post/how-does-a-window-cleaning-robot-work>

*Concord New Energy*. (n.d.). Retrieved from Innovating for a Greener Tomorrow with Renewable Energy Solutions: <https://concordnewenergy.com/about-us/>

Davis, E. (2025, June 26). *AI in Robotics: How ROS and Python Are Powering Autonomous Machines*. Retrieved from Codingcops: <https://codingcops.com/ai-in-robotics/>

Deremuk, I. (2024, September 4). *Python vs. Java: Which One to Choose for Your AI Project?* Retrieved from Litslink: <https://litslink.com/blog/python-vs-java-which-one-to-choose-for-ai>

- Dogra, M. (2020, December 13). *Weakly Supervised Learning for Object Localization*. Retrieved from Medium: <https://medium.com/analytics-vidhya/weakly-supervised-learning-for-object-localization-4b73d4f4a6>
- Ecovacs. (2025, 08 05). *How Do Robot Window Cleaners Work to Keep Your Windows Spotless?* Retrieved from Ecovacs: <https://www.ecovacs.com/uk/blog/how-do-robot-window-cleaners-work>
- Embeddings: Embedding space and static embeddings*. (n.d.). Retrieved from developers.google.com: <https://developers.google.com/machine-learning/crash-course/embeddings/embedding-space>
- Frank Dellaert, S. H. (2023). *5.5 Path Planning*. Retrieved from roboticsbook: [https://www.roboticsbook.org/S55\\_diffdrive\\_planning.html](https://www.roboticsbook.org/S55_diffdrive_planning.html)
- How to use classification threshold to balance precision and recall*. (2025, January 9). Retrieved from EvidentlyAI: <https://www.evidentlyai.com/classification-metrics/classification-threshold>
- Jaiswal, S. (2024, January 4). *What is Normalization in Machine Learning? A Comprehensive Guide to Data Rescaling*. Retrieved from Datacamp: <https://www.datacamp.com/tutorial/normalization-in-machine-learning>
- Jana. (2025, March 25). Retrieved from Hycleaner: <https://hycleaner.de/nl/vind-het-perfecte-zonnepaneelreinigungsrobot/#:~:text=De%20solarROBOT%20compact:%20de%20deale,worden%2C%20zelfs%20onder%20moeilijke%20omstandigheden>
- Luca, G. D. (2024, August 30). *Euclidean Distance vs Cosine Similarity* . Retrieved from Baeldung: <https://www.baeldung.com/cs/euclidean-distance-vs-cosine-similarity>
- M. Menagadevi, S. D. (2024). *Machine and deep learning approaches for alzheimer disease detection using magnetic resonance images: An updated review*. Retrieved from ScienceDirect: <https://www.sciencedirect.com/science/article/abs/pii/S0263224123016640>
- Mancheva, M. (2015, March 11). *renewablesnow.com*. Retrieved from China WindPower officially becomes Concord New Energy: <https://renewablesnow.com/news/china-windpower-officially-becomes-concord-new-energy-467530/>
- Maria Tzelepi, P. N. (2022). *Representation learning and retrieval*. Retrieved from ScienceDirect: <https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780323857871000154>

- Murphy, I. E. (n.d.). *What Is Machine Learning?* Retrieved from Springer Link:  
[https://link.springer.com/chapter/10.1007/978-3-319-18305-3\\_1](https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1)
- NEMA17 stappenmotor. (n.d.). Retrieved from 123-3d.nl: [https://www.123-3d.nl/123-3D-NEMA17-stappenmotor-1-8-graden-per-stap-40-mm-lang-4-08-kg-cm-SL42S240A105-0524-i3421.html?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=PPC-SEA-NL-Google-Shopping-B-All-Segments-Parts-Other-CPA-02-00&gad\\_source=1&gad\\_campaign](https://www.123-3d.nl/123-3D-NEMA17-stappenmotor-1-8-graden-per-stap-40-mm-lang-4-08-kg-cm-SL42S240A105-0524-i3421.html?utm_source=google&utm_medium=cpc&utm_campaign=PPC-SEA-NL-Google-Shopping-B-All-Segments-Parts-Other-CPA-02-00&gad_source=1&gad_campaign)
- Noor, U. (2025, July 30). Retrieved from 8MSOLAR: <https://8msolar.com/solar-powered-robotics/#:~:text=Unlike%20traditional%20robots%20that%20rely,%2C%20sensors%2C%20and%20computing%20operations.>
- Nwoke, T. (2023, May 30). *Image Embeddings explained*. Retrieved from Picsellia:  
<https://www.picsellia.com/post/image-embeddings-explained>
- Performance Analysis of a Solar Panel Cleaning Autonomous Robot (SPCR) with comparative study*. (2023, December). Retrieved from ResearchGate:  
[https://www.researchgate.net/publication/378541389\\_Performance\\_Analysis\\_of\\_a\\_Solar\\_Panel\\_Cleaning\\_Autonomous\\_Robot\\_SPCR\\_with\\_comparative\\_study](https://www.researchgate.net/publication/378541389_Performance_Analysis_of_a_Solar_Panel_Cleaning_Autonomous_Robot_SPCR_with_comparative_study)
- Physics*. (n.d.). Retrieved from unity3d.com:  
<https://docs.unity3d.com/Manual/PhysicsSection.html>
- Polygenis, T. (2023, May 03). *How Strong Is 3D Printed Plastic: A Comprehensive Guide*. Retrieved from wevolver: <https://www.wevolver.com/article/pla-strength>
- Rachid, N. N. (2023, December 8). *A Review on Solar Panel Cleaning Systems and Techniques*. Retrieved from MDPI: <https://www.mdpi.com/1996-1073/16/24/7960#B32-energies-16-07960>
- Rajshekhar. (2022, September 8). *Object Detection Series 3.0 : Sliding Window Approach* . Retrieved from Medium: [https://medium.com/@rajshekhar\\_k/object-detection-series-3-0-basic-of-object-detection-196d91550671](https://medium.com/@rajshekhar_k/object-detection-series-3-0-basic-of-object-detection-196d91550671)
- Raspberry Pi 4 vs 5 Benchmarks*. (n.d.). Retrieved from origin-ic.com: <https://www.origin-ic.com/blog/raspberry-pi-4-vs-5/47744>
- Siamese Neural Network* . (2021). Retrieved from ScienceDirect:  
<https://www.sciencedirect.com/topics/computer-science/siamese-neural-network#6.%20Conclusion>

- Svarc, J. (2022, Februari 2). *Solar Panel Construction*. Retrieved from CER:  
<https://www.cleanenergyreviews.info/blog/solar-panel-components-construction>
- Team, L. (2021, June 10). *Edge Detection Using OpenCV*. Retrieved from LearnOpenCV:  
<https://learnopencv.com/edge-detection-using-opencv/>
- Tiwari. (2022). *Supervised learning: From theory to applications*. Retrieved from ScienceDirect: <https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780128240540000265>
- Trigolos, A. (n.d.). *C++ VS Python For Machine Learning*. Retrieved from elinext:  
<https://www.elinext.com/solutions/ai/machine-learning/trends/cpp-vs-python-for-machine-learning>
- Understanding Ultrasonic Sensor*. (2023, November 20). Retrieved from Medium:  
<https://medium.com/@robotamateur123/understanding-ultrasonic-sensor-e3791f883061>
- Unity. (n.d.). *Unity Robotics Hub*. Retrieved from Github: <https://github.com/Unity-Technologies/Unity-Robotics-Hub>
- What are convolutional neural networks?* . (2025, November 17). Retrieved from IBM:  
<https://www.ibm.com/think/topics/convolutional-neural-networks>
- What is Heatmap Data Visualization and How to Use It?* (2024, June 12). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/data-visualization/what-is-heatmap-data-visualization-and-how-to-use-it/>
- Wickstrom, S. (2025, march 21). *PETG vs PLA vs ABS: 3D Printing Strength Comparison*. Retrieved from Ultimaker: <https://ultimaker.com/learn/petg-vs-pla-vs-abs-3d-printing-strength-comparison/>
- Yi, M. (2025). *Heatmap: A complete Guide*. Retrieved from Atlassian:  
<https://www.atlassian.com/data/charts/heatmap-complete-guide>
- zonneplan*. (n.d.). Retrieved from  
<https://www.zonneplan.nl/kenniscentrum/zonnepanelen/werking/zonnecellen>