

GEÏNTEGREERDE PROEF

Schooljaar 2021-2022

Henri Lahousse - 6ELA

Onward



PROVINCIAAL TECHNISCH INSTITUUT

Campus Techniek & Design

Graaf Karel de Goedelaan 7

8500 Kortrijk

T 056 22 13 41 E gip.td@pti.be



GEÏNTEGREERDE PROEF

Schooljaar 2021-2022

Henri Lahousse - 6ELA

Onward



PROVINCIAAL TECHNISCH INSTITUUT

Campus Techniek & Design

Graaf Karel de Goedelaan 7

8500 Kortrijk

T 056 22 13 41 E gip.td@pti.be

Woord vooraf

Met mijn geïntegreerde proef (gip) sluit ik mijn secundaire studies af in elektriciteit elektronica. De school speelde hier een grote rol in. Ze stimuleerde ons om een eigen creatief project te zoeken en dit uit te werken. Hiervoor kregen we de nodige tijd en hulp. Deze hulp kwam vooral van onze leerkrachten en de kennis die we via hen hebben opgedaan.

Mijn ouders speelden een belangrijke rol voor mijn gip. Ze verbeterden mijn gipboekje, boden financiële steun en gaven tips waar nodig. Daarvoor ben ik ze zeer dankbaar.

Natuurlijk bedank ik meneer Durnez, onze klastitularis, die met het nodige geduld alles goed liep verlopen. Ik bedank ook meneer Dewaele die de lessen uitbreidde om onze gips te verbeteren. Hij gaf met veel plezier extra uitleg en advies. Meneer Neiryck stak ook super veel tijd en moeite in ons gipboekje waar ik hem voor dank.

Tot slot bedank ik RS Components dat camera's, Raspberry Pi's, batterijen en processoren sponsorde.

Inhoudsopgave

Inleiding	1
1 Hardware	5
1.1 Inputs	6
1.1.1 Camera	6
1.1.2 Voice-assistent	7
1.1.3 Ultrasonische sensoren	8
1.1.4 Drone	8
1.1.5 Druksensor	9
1.1.6 Reedcontact	9
1.2 CPU	10
1.3 Outputs	11
1.3.1 DC-motor	11
1.3.2 Lineaire motoren	11
1.3.3 Motor driver	13
1.4 Voeding	14
1.4.1 Spanningsregelaar	14
2 Software	15
2.1 Gezichtsherkenning	15
2.1.1 Problemen	16
2.2 Motorcontrol	16
2.2.1 AI-model	17
2.2.2 Problemen	20
2.2.3 Oplossing	20
2.3 Voice-assistent	21
2.3.1 Rhino	21
2.3.2 Porcupine	21
2.3.3 Problemen	22
2.3.4 Oplossing	22
2.4 Agent	23
2.4.1 Artificiële intelligentie	23

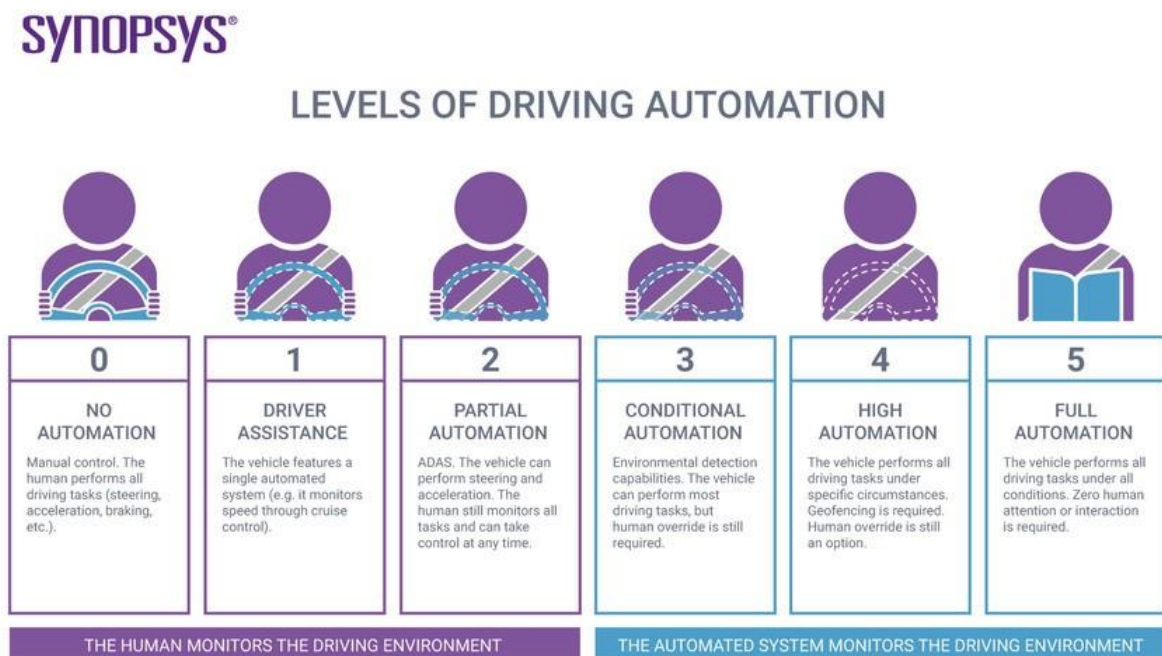
2.5	Drone	25
2.5.1	Superimpose	25
2.5.2	Objecttracking.....	26
2.6	Cloud	27
2.7	Kubernetes.....	27
2.7.1	Wat is Kubernetes?	28
3	Design	30
4	Prijsberekening	36
	Algemeen besluit	37
	Bijlagen	39
	Bijlage 1: curriculum vitae	40
	Bijlage 2: lijst van afbeeldingen	41
	Bijlage 3: schema's	43
	Bijlage 4: datasheets	45
	Bijlage 5: code	47
	Bronvermelding	48

Inleiding

In 2020 reden er ongeveer tien miljoen zelfrijdende voertuigen op de openbare weg. Binnen tien jaar zou wereldwijd één op de drie voertuigen zelfrijdend zijn. Deze markt is momenteel 50 miljard euro waard. Tegen 2030 zal dit met minimaal 4 000 procent stijgen.

Aangezien er zoveel geld gepompt wordt in deze industrie gaat alles razendsnel vooruit. Zelfrijdende voertuigen zorgen niet alleen dat u van punt a tot punt b raakt zonder zelf iets te moeten doen. Ze zorgen er ook voor dat software en hardware enorme evoluties meemaken die dan op ieder domein toegepast kunnen worden, zoals in de geneeskunde, industrie ...

De infografiek hieronder legt de verschillende levels in autonoom rijden uit. Mijn doel is om een volledig autonoom voertuig, een level 5, dus, te ontwerpen.



FIGUUR 1: LEVELS SELFDRIVING

Als gip wil ik een zelfrijdend éénpersoonsvoertuig genaamd Onward ontwerpen. De naam verwijst naar het vooruitgaan met het voertuig en de vooruitgang in software. De basis van mijn eindwerk is een elektrisch voertuig. Het einddoel is om Onward zelfstandig op een eenvoudige weg te laten rijden terwijl hij de juiste keuzes maakt op basis van artificiële intelligentie (AI). De AI krijgt gefilterde camerabeelden als input en wordt ook zo getraind.

Onward haalt zijn energie uit twee 12V-loodzuurbatterijen met een capaciteit van 24 Ah. Samen vormen ze de 24 V die nodig is voor de motoren. Een 24V DC-motor drijft de achterwielen aan. Een lineaire DC-motor regelt de stand van de voorwielen. Voor het op- en neerduwen van het dak, dat ook als deur functioneert, wordt een tweede lineaire DC-motor gebruikt. De sensoren voor dit project zijn vier ultrasonic sensoren en een camera.

Nu komen we aan het belangrijkste, interessantste en tofste deel: de software. Opencv filtert de camerabeelden op kleur, zodat de AI zich gemakkelijker kan trainen. Eerst worden de kleuren op het beeld versterkt, dan worden de storingen uit het beeld gefilterd en tot slot detecteert het programma sterke veranderingen in kleur en tekent zo op een zwart vlak witte lijnen. Wit is in het vlak van beeldverwerking de beste kleur omdat het de meeste pixels heeft, wat het programma beter verwerkt dan zwart.

Om de data efficiënt te gebruiken doen we aan maxpooling. Dit houdt in dat we ieder beeld spiegelen, de scherpte ervan verminderen, de kleuren aanpassen ... om zo van één trainingsbeeld vier beelden te maken. Hierdoor winnen we tijd.

Voor de AI zelf zal ik behaviour learning gebruiken, wat inhoudt dat de mens het voor toont. De AI moet leren om correlaties te maken op basis van wat de mens doet en wat hij zelf voorspelde te moeten doen.

Tot slot heb je het design. Ik laat me inspireren door een conceptauto van Apple. De afmetingen bedragen twee meter twintig bij een meter vijftig.

Ik heb ook een paar extra's in gedachten.

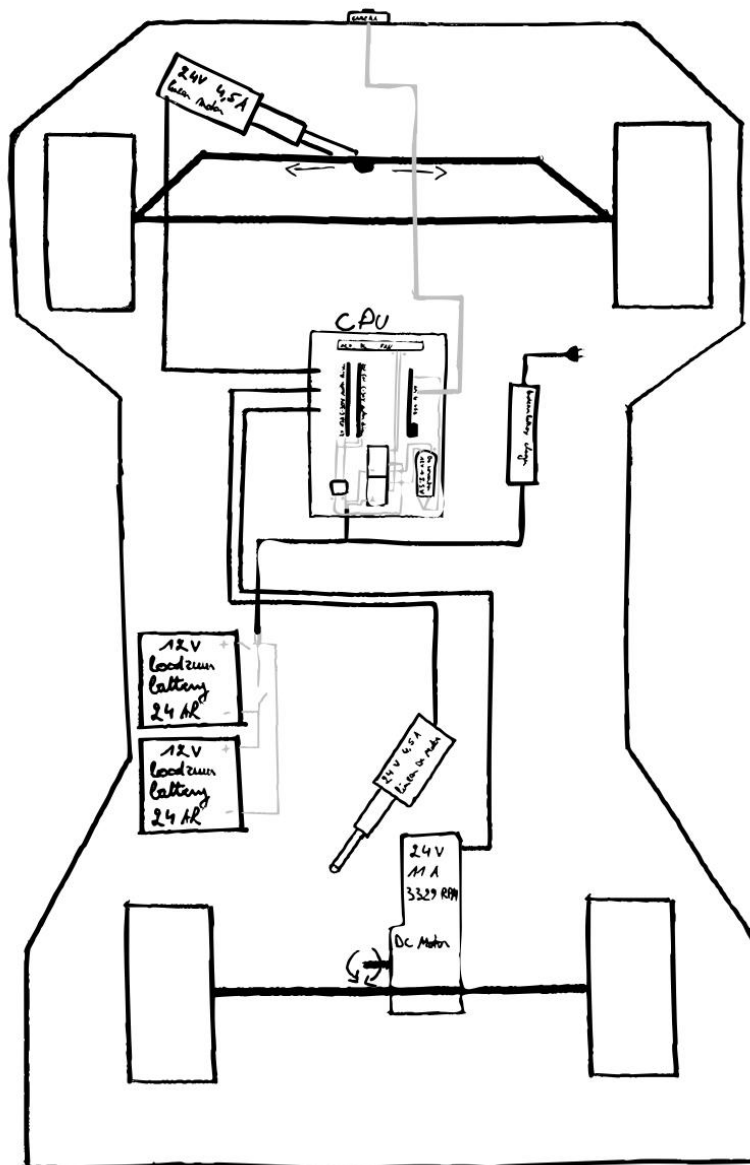
- Een tweede camera maakt het mogelijk om Onward met face id te openen. Dit is mogelijk en vrij eenvoudig met Deepface. Dit is een bibliotheek ontwikkeld door een

MIT-student. Zijn software maakt het ook mogelijk om onder andere emoties te detecteren. Hier zou ik dan gewoon de lichten van kleur laten veranderen.

- Een drone zal gelinkt zijn met Onward om hem te evalueren.

1 Hardware

In de schema bijlages vindt u het blokschema en gedetailleerd schema. Onderstaande schets geeft weer waar de belangrijkste componenten zich in het voertuig bevinden.



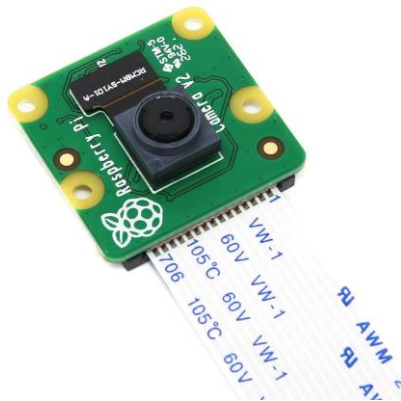
FIGUUR 2: SCHETS

1.1 Inputs

Ik behandel in dit hoofdstuk alles qua hardware dat data ontvangt.

1.1.1 Camera

Ik gebruik de V2 Raspberry Pi camera module. Dit is een camera op maat van een Raspberry Pi (RPI). Deze camera is aan de voorkant het voertuig gemonteerd. Bij verdere uitbreiding plan ik om in het totaal vijf camera's te gebruiken. De Tello drone, die verder vernoemd wordt heeft ook een camera.



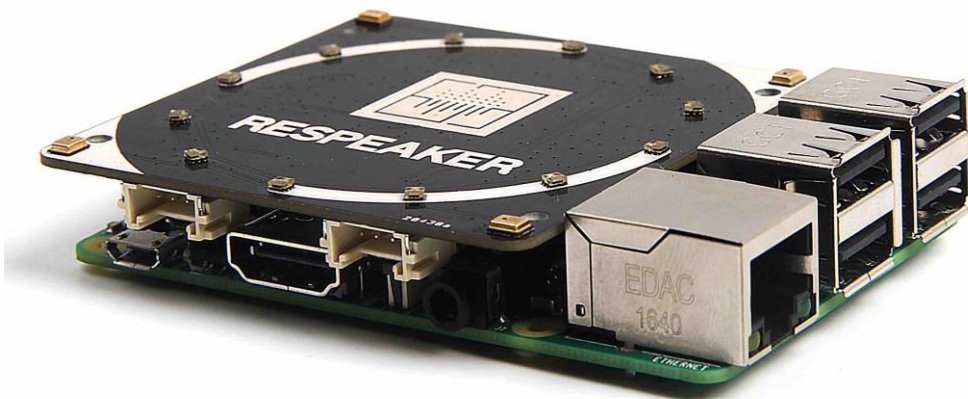
FIGUUR 3: RPI-CAMERA

Een camera neemt licht en focust het via de lens op een sensor die uit silicone bestaat. Dit oppervlak bestaat uit superkleine lichtgevoelige photosites, een pixel genoemd. Er zijn miljoenen van deze pixels in een rooster aanwezig op de sensor.

Die worden dan in een digitaal signaal omgezet en naar de verwerkingseenheid doorgestuurd.

1.1.2 Voice-assistent

Om Onward gebruiksvriendelijker te maken implementeer ik een voice-assistent. Ik koos de Respeaker 4-mics array Raspberry Pi module om dit waar te maken. Dit is een quad-microfoon uitbreidingsbord gemaakt voor AI en stem applicaties..



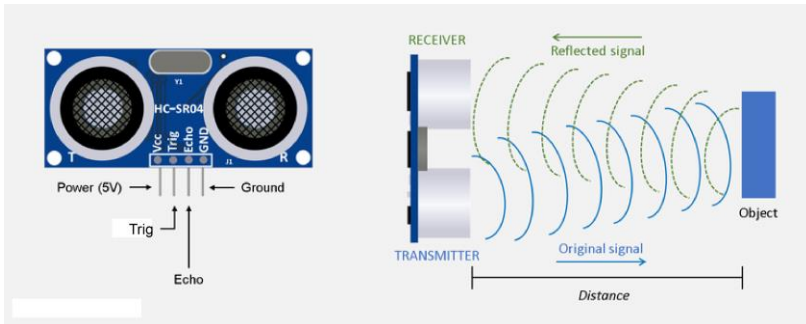
FIGUUR 4: RESPEAKER

De module is uitgerust met vier microfoons, in elke hoek één. Door deze oriëntatie is de module geschikt om de bron van het geluid te detecteren. Dit is één van de redenen waarom ik dit model koos. Wanneer iemand langs het voertuig wandelt, kan de module dit perfect detecteren en bepalen waar de persoon staat. De Respeaker-module kan geluid accuraat opvangen binnen een radius van drie meter.

U ziet dat er twaalf blokjes in een cirkel op het bordje staan. Dit zijn twaalf APA102 programmeerbare leds. Dit is vergelijkbaar met de led-ringen van een Alexa Echo en een Google Home.

1.1.3 Ultrasonische sensoren

Onward is uitgerust met vier ultrasonische sensoren: drie langs de voorkant en één langs achteren. Ultrasonische sensoren bestaan uit twee delen. De trigger stuurt een signaal uit dat de receiver dan ontvangt. Dit signaal is uniek voor iedere module om verwarringen tussen de sensoren te vermijden. Door de tijd te berekenen tussen het versturen en het ontvangen van het signaal kan de afstand worden ingeschat.



FIGUUR 5: ULTRASONISCHE SENSOR

1.1.4 Drone

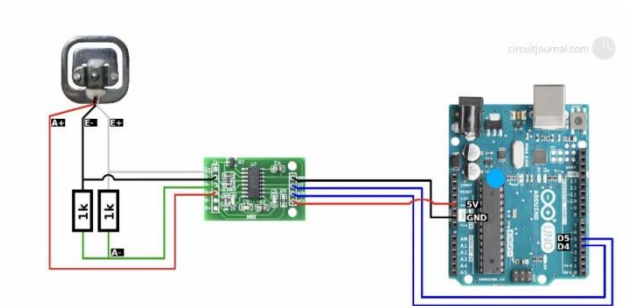
De Tello drone is een veel gebruikte beginnersdrone aangezien het super gebruiksvriendelijk is. Het is gemakkelijk om te besturen en is goedkoop. Deze drones zijn open source en kan je gemakkelijk over wifi programmeren in scratch of python. De drone bevat een 5mp camera dat resulteert in een 720p hd beeld, altitude sensoren, 13 min vliegtijd en een Intel processor.



FIGUUR 6: TELLO DRONE

1.1.5 Druksensor

Om te weten wanneer er iemand neerzit in het voertuig en het dak terug naar beneden mag plaatste ik een druksensor in de rugleuning. Ik koos voor de SEN-10245 druksensor omdat het tot een kracht van 50kg kan meten. Zelf al drukt er een grotere kracht dan 50kg op de rugleuning kan dit geen kwaad. Ik wil enkel weten wanneer er iemand is gaan neerzitten.



FIGUUR 7: DRUKSENSOR

De druksensor bevat een weerstand. Deze weerstand vergroot naarmate er een grotere kracht drukt op de sensor. Dit weerstandsverschil is zo klein dat je een amplifier, een versterker, nodig hebt om dit signaal leesbaar te maken voor de verwerkingseenheid. De metalen grijze component is de druksensor. Het groene plaatje is de versterker.

1.1.6 Reedcontact

Deze sensor is de gemakkelijkste en goedkoopste manier om te detecteren wanneer er iets op of dicht is. Ik bevestigde een reedcontact aan het dak om zo te zien wanneer deze gesloten is en wanneer niet. Een reedcontact is een schakelaar die aangestuurd wordt door een magneet. Het contact opent of sluit dan.



FIGUUR 8: REEDCONTACT

1.2 CPU

Ik koos de Raspberry Pi (RPI) model 4 met 4gb ram als centrale verwerkingseenheid. Een RPI is een IOT-device dat alles kan wat een computer kan. Ze zijn extreem gegeerd aangezien ze relatief goedkoop zijn maar vooral omdat er een gigantische community is ontstaan rond deze toestellen. Dit plus dat je ze met Python kan programmeren. Python is de populairste en simpelste taal en heeft ook een enorme community.



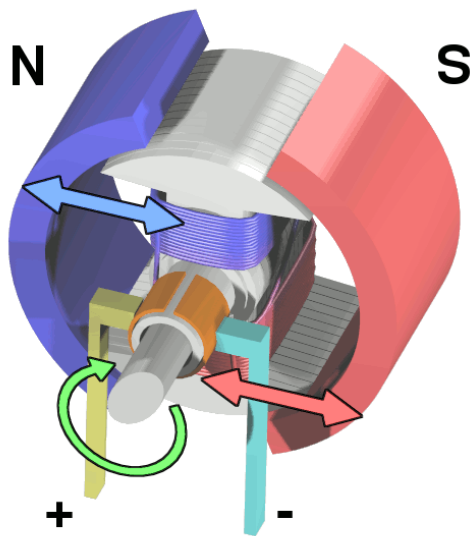
FIGUUR 9: RASPBERRY PI 4

De RPI ontvangt alle data, stuurt alles in dit project aan en verwerkt alles lokaal. Eerst was ik van plan om de cloud te gebruiken voor berekeningen. Met de cloud zou ik zeer zware programmas kunnen draaien. Maar aangezien ik dan overal een stabiele internetconnectie moest hebben ben ik van dit idee afgestapt. Deze RPI kan redelijk wat aan maar zoals je verder zal lezen heeft dit me wel gelimiteerd. Ik vond hier een oplossing voor en zal dit later nog toevoegen. Ik ben wel nog verbonden met de cloud maar alleen voor data mooi en live te visualiseren voor de gebruiker. Als de verbinding even weg valt kan dit helemaal geen kwaad. In mijn gip gebruik ik twee RPI's en zal er later waarschijnlijk nog toevoegen. De master RPI stuurt alles aan. Het enige doel van de tweede RPI is om de data van de voice-assistent om te zetten naar bruikbare data en deze naar de master door te sturen. Dit gebeurt op een aparte RPI omdat de voice-assistent module alle pinnen in beslag neemt.

1.3 Outputs

1.3.1 DC-motor

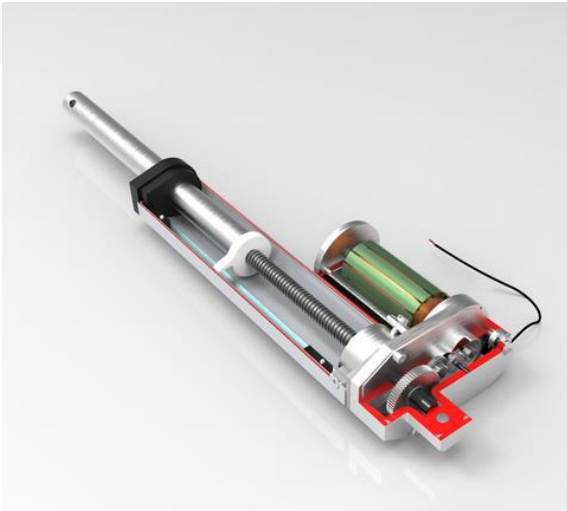
Één 24V/11A DC-motor drijft de achterwielen aan met een ketting. De snelheid en de richting van de motor wordt via een motor driver via de master RPI bepaald. In de bijlage vindt u een tekening dat de werking van een DC-motor uitlegt.



FIGUUR 10: DC-MOTOR

1.3.2 Lineaire motoren

Ik gebruik twee 24V DC 4,5A 2400N lineaire motoren in mijn gip. Deze worden op dezelfde manier maar met een andere motor driver aangestuurd als bovenstaande motor. Oorspronkelijk waren dit er vier, verder in dit boekje leest u waarom dit is veranderd. Een lineaire motor is gewoon een dc-motor dat een tandwiel aandrijft die dan een as naar binnen en buiten doet draaien. In onderstaande bijlage ziet u hoe dit in elkaar zit.



FIGUUR 11: LINEAIRE MOTOR

Er is één lineaire motor gemonteerd onder het voertuig om de voorwielen van kant te doen draaien. De tweede motor dient om het dak te openen en sluiten.



FIGUUR 12: LINEAIRE MOTOREN SETUP

1.3.3 Motor driver

Om motoren van richting te veranderen moet je de polen omwisselen. Gelukkig bestaan er h-bruggen die dit voor ons doen. Nu kunnen we vooruit en achteruit maar dit is direct volle bak. Een motor driver regelt zowel de richting en de snelheid van de motoren. In onderstaande bijlage ziet u een foto van beide motor drivers dat ik gebruik.



FIGUUR 13: MOTOR DRIVER 20 A 30 V



FIGUUR 14: MOTOR DRIVER 10 A 30 V

Maar hoe verandert de motor driver de snelheid? Dit doet hij met behulp van pulse width modulation (PWM) of in het nederlands pulse breedte modulatie (PBM). PWM schakelt de verbruiker super snel aan en uit. Hoe minder je aan en uit schakelt hoe trager de motor draait.

1.4 Voeding

Het zou een beetje belachelijk zijn om een zelfrijdende auto te laten werken op het net dus koos ik om loodzuur batterijen te gebruiken. Dit zijn goedkope en veilige batterijen die tegen een stootje kunnen. Ze hebben wel als nadeel dat ze groot, zwaar en niet zo krachtig zijn. Mijn voeding bestaat uit twee 12V-loodzuur-batterijen die dan samen 24 V kunnen leveren voorzien met een aftakking voor de 12V-verbruikers. De batterijen worden opgeladen door een loodzuurbatterijoplader. Deze is ingebouwd in het voertuig. Je moet het gewoon nog in oplaadmodus zetten en in het stopcontact steken.



FIGUUR 15: LOODZUUR-BATTERIJ

1.4.1 Spanningsregelaar

Mijn voeding levert 24 V en 12 V maar mijn RPI werkt op 5,5 V. Zoals de naam vermeldt, regelt deze module de spanning. Met onderstaande spanningsregelaar regelde ik 12 V tot 5,5 V. De RPI heeft maar 3 A nodig, deze module kan tot 8 A leveren.



FIGUUR 16: SPANNINGSREGELAAR

2 Software

De software is essentieel voor mijn gip. Een voertuig zelfstandig laten rijden is een enorme taak. Dit is niet haalbaar in één schooljaar als ik helemaal alleen, alles van a tot z moet ontwikkelen. Daarom focus ik me op het rijden tussen twee lijnen, zoals op een autostrade.

2.1 Gezichtsherkenning

Sefik Ilkin Serengil is een Turkse software-ingenieur die als eindwerk Deepface ontwikkelde. Zijn bibliotheek analyseert gezichten op basis van leeftijd, emoties, geslacht en etniciteit. Ik botste toevallig op deze student zijn paper en voegde dit toe als extraatje.



FIGUUR 17: DEEPFACE

Mijn code vergelijkt foto's van mijn database met de input van de camera. Zodra hij een gezicht herkent, ontgrendelt het voertuig. Met Sefik zijn bibliotheek analyseer ik gezichten op emotie en verander de leds op basis van deze output. Als er een gezicht in beeld komt dat hij niet herkent, slaat hij dat toch op. Hij genereert een willekeurige naam voor deze persoon en voegt die toe aan de database. Bijkomende data worden ook opgeslagen en hiermee gelinkt.

2.1.1 Problemen

Door het plots uitkomen van vele nieuwe softwareversies ondervond ik even problemen met dit deel van de software. De bibliotheek van de student werd enkel ondersteund door bepaald softwareversies. Voor andere delen van mijn software had ik een andere of de telkens de nieuwste update nodig.

2.2 Motorcontrol

Ik maakte het mezelf moeilijker door het voertuig met twee lineaire motoren voort te bewegen. De pedalen van een fiets werden omgekeerd gemonteerd op de achterkant en werden via een ketting verbonden met de as. De lineaire motoren duwen en trekken dan telkens in tegengestelde richting aan de pedalen. Ik ontwikkelde mijn eigen AI-model dat de positie van het tandwiel kan bepalen op basis van de gefilterde camera-output. Hierdoor weet ik exact wanneer ik de motoren van richting moet veranderen.



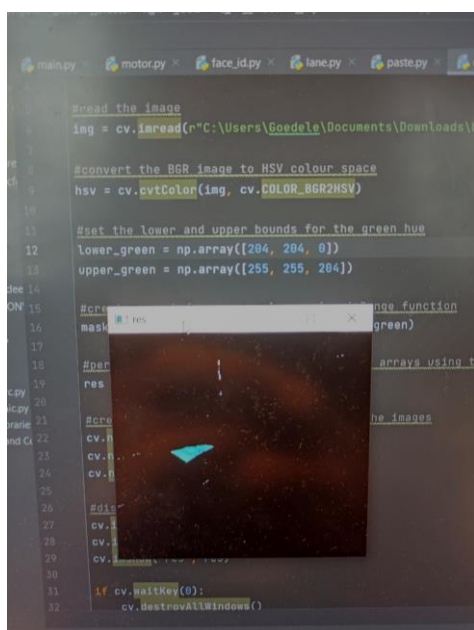
FIGUUR 18: LINEAIRE MOTOREN ACHTERWIELAANDRIJVING

2.2.1 AI-model

De RPI neemt met een V2 RPI camera iedere milliseconde een foto van de pedalen. Het uiteinde van de pedaal is felblauw gekleurd.

2.2.1.1 Colorfiltering

De colorfiltering software filtert de afbeeldingen op kleur. Het vergelijkt iedere pixel met de ingestelde upper en lower bond grenzen. Iedere pixel dat binnen dit spectrum valt wordt wit gekleurd. Alle andere pixels worden zwart. Uiteindelijk bekom je een afbeelding waar enkel het felblauw gekleurde deel wit gekleurd is en de rest zwart. Ik koos zwart als achtergrond omdat dit de minst opwindende kleur is voor een computer. Wit is dan weer het tegenovergestelde, de computer is hier het meest in geïnteresseerd. Onderstaande afbeelding toont hoe accuraat mijn colorfilteringsoftware werkt. Het lichtblauwe stukje papier is het enige dat zijn kleur behoudt de rest is weggefilterd.



FIGUUR 19: KLEURFILTERING

2.2.1.2 Image classification

Aan de hand van supervised machine learning genereerde ik een image classification model.

Ik genereerde een dataset van een paar duizend afbeeldingen gefilterd door de colorfilteringssoftware en gesorteerd per hoek van de pedalen. De dataset wordt dan opgesplitst in een train en test dataset. Dit noemt men de hold-outmethode. Indien je je model zou testen met dezelfde train data zou je geen accurate predictie krijgen voor ongeziene data. Je wilt dat je model begrijpt waarom je afbeeldingen in bepaalde groepen opdeelt en niet dat hij het gewoon memoriseert.

Let op welke dataset je gebruikt! Vooraleer je je data gebruikt pas exploratory data analysis (EDA) toe! Dit houdt in dat je check of je data compleet is, nuttig, goed gesorteerd, ... Niemand doet dit graag maar het is noodzakelijk. Gelukkig bestaat er software die ons hierbij helpt.

Ik neem nu het voorbeeld van een model dat honden van wolven onderscheidt. De kans dat de wolf sneeuw als achtergrond heeft is redelijk groot. Bij een hond zou dit zelden voorkomen. Het zou dus evengoed kunnen zijn dat het model denkt, ahhhh een witte achtergrond, dit zal een wolf zijn. Je afbeeldingen worden dan op basis van de achtergrond gesorteerd en niet het dier. Hij zou dan denken dat een hond met een witte achtergrond een wolf is.



FIGUUR 20: EDA/WOLF

Het is enorm moeilijk om een perfecte dataset te verzamelen maar zorg ervoor dat de groepen enkel te onderscheiden zijn door het object dat je bedoeld. Het is onmogelijk om te weten hoe je model werkt maar grote fouten zoals deze vermijd je best.

Vooraleer we beginnen aan het trainen genereren we meer data (maxpooling) door foto's te draaien, lichtinval aan te passen, ... Bij deze toepassing kunnen we moeilijk de foto's gaan draaien aangezien we trainen om de hoeken te herkennen. Dit hangt af van project tot project en mag je zeker niet vergeten. Met deze extra data zal het model veel accurater zijn doordat het al meer heeft kunnen oefenen. Dit is ook extreem handig wanneer je weinig data ter beschikbaar hebt.

Eenmaal dat het trainen voltooid is en de accuraatheid, precisie en andere parameters voldoen aan de ingestelde waarden wordt dit model opgeslagen. Het evalueren van een model gebeurt met het confusion matrix.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

FIGUUR 21: CONFUSION MATRIX

Het tweede deel van de code, de productiecode, gebruikt het getrainde model om nieuwe data te classificeren.

2.2.2 Problemen

Uiteindelijk was dit deel software gewoon een uitdaging. Ik wist dat het zeker niet de gemakkelijkste oplossing was maar ik wou kijken of ik het kon. De software werkt perfect en ik heb het al gebruikt voor andere projecten.

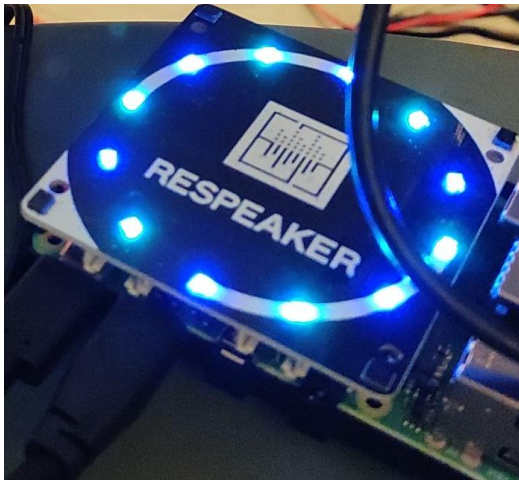
Het mechanische deel, wat zeker niet mijn sterke kant is, stak teken. De twee lineaire motoren werkten meestal perfect samen om het voertuig voort te bewegen maar heel soms liep dit mis. Door de enorme kracht van de motoren trokken deze mijn laswerk kapot en trokken de as van de achterwielen scheef.

2.2.3 Oplossing

Ik heb een tijd gewerkt aan het oplossen van deze problemen maar moest uiteindelijk vlak voor de gipverdediging nog overschakelen naar een dc-motor.

2.3 Voice-assistent

Tot mijn verbazing vond ik amper projecten waar er met de Respeaker module een eigen voice-assistent werd gemaakt. Uiteindelijk botste ik op software genaamd Picovoice. Picovoice maakt het mogelijk om de computerwoorden te laten verstaan. Dit wordt Natural Language Understanding of NLU genoemd. Met deze software is het mogelijk om een model te trainen voor specifiek applicaties. Deze software kan je dan nog eens in twee delen opsplitsen.



FIGUUR 22: RESPEAKER

2.3.1 Rhino

Rhino is een onderdeel van de Picovoice software dat wakewords modeleert. Een wakeword is het woord waarmee je de voice-assistent aanspraak zoals 'Alexa' bij Alexa Echo of 'Hey Google' bij een Google home. Bij Onward is het wakeword 'Hey Onward'.

2.3.2 Porcupine

Porcupine is het onderdeel van Picovoice dat bepaalde situaties en de commando's in die functies modeleert. Bijvoorbeeld: 'open the roof'. Het is redelijk gecompliceerd om situaties af te kunnen leiden gelukkig doet Porcupine dit voor ons.

Picovoice heeft een website waar je gemakkelijk een model kan ontwikkelen voor specifieke applicaties. (De link naar deze website is terug te vinden in de bronvermeldingen.) Dit werkt zowel voor je computer als de respeaker module. Ieder wakeword en situatie wordt apart gemodelleerd. Deze bestanden kan je dan downloaden en in een stuk productiecode smijten. Het getrainde model doet voorspellingen op nieuwe data. Indien een wakeword of situatie herkend wordt stuurt hij deze door naar de cloud. De master RPI haalt deze data uit de cloud en voert indien nodig commando's uit. Het dak laten opgaan tot de drone laten opstijgen tot stemmen herkennen. Simpelweg iedere mogelijke actie kan bestuurd worden met de voice-assistent.

2.3.3 Problemen

Ik kwam erachter waarom er amper mensen deze module gebruiken voor projecten. Alles werkte perfect en plots reageerde het op geen enkel commando. Ik had niks veranderd zowel softwarematig als hardwarematig. Na het lopen van verschillende built-in basisprogramma's die altijd zouden moeten werken ging ik ervanuit dat de module kapot was. Ik kocht een nieuwe module waar ik uiteindelijk de exact zelfde problemen tegenkwam. Via Audacity software checkte ik beide Raspberry Pi's op inkomende geluidsgolven. Beide modules detecteerden geluid dat er niet was en vingen effectief aanwezig geluid niet op. Na een tijd te googelen kwam ik op een forum waar meerdere mensen dit probleem tegenkwamen. Het bleek dat de software die het mogelijk maakt voor de module om geluid op te vangen in onderhoud was. Het bedrijf dat deze module onderbouwt blijkt verre van gebruiksvriendelijk te zijn aangezien dit vaak voorkomt. Ze raadden aan van een volgende versie te installeren aangezien de oude niet meer ondersteund worden door andere bibliotheken. De versie die ze aanraadden zou onlangs uitgekomen zijn maar er is niets nieuws uitgekomen sinds 2021-05-01.

2.3.4 Oplossing

Verschillende groepen proberen zelf software te ontwikkelen om deze module terug in gang te krijgen. Ik zal blijven proberen deze module terug te laten werken maar ik vrees ervoor.

2.4 Agent

Dit brengt ons tot het kloppende hart of eerder het brein. Dit deel van de code is verantwoordelijk voor het zelfstandig rijden en beslissingen nemen. Dit alles werkt met artificiële intelligentie.

2.4.1 Artificiële intelligentie

Iedereen hoorde wel al eens van artificiële intelligentie (AI) en heeft er zijn eigen mening over. Sommigen vinden dat dit een tijdelijke nutteloze rage is, anderen geloven dat dit de oplossing is voor alles, nog anderen denken dat AI de wereld zal overnemen en de mens zal uitmoorden. Ik persoonlijk ben een grote fan van AI en zie er wel degelijk het nut van in.

2.4.1.1 Wat is AI?

AI zou in theorie moeten kunnen voelen, discussiëren en evolueren, zoals de naam vermeldt. Dan moet AI dus ook problemen kunnen oplossen waar het niet specifiek voor ontwikkeld is, zelf initiatief nemen, ... Dit wordt general of strong AI genoemd. Spijtig genoeg bestaat dit nog niet.

De AI waar iedereen voortdurend over praat is narrow of weak AI. Deze programma's zijn elk specifiek gemaakt om een bepaald probleem op te lossen of taak uit te voeren. Dit is echter niet helemaal accuraat. Je kan AI-modellen voor meerdere applicaties gebruiken, maar dan moet er wel een correlatie zijn tussen de twee problemen. Dit noemt men transfer learning.

Hoewel weak AI de mens niet zal overheersen, heeft deze wel zeer veel nuttige toepassingen in onze maatschappij. Van 'ongeneeslijke' ziektes aanpakken, over kunst creëren tot studenten online bijscholen.

2.4.1.2 Machine learning

Nu we weten wat AI is, kunnen we ons hierin verdiepen. AI bestaat uit machine learning (ML). Machine learning zijn algoritmes die met de tijd verbeteren doordat ze meer data verwerken. Hier bestaan drie types van:

- supervised learning (meest gebruikt, voorspelt categorieën op basis van de input)
- unsupervised learning (analyseert relaties tussen input/niet-gelabelde data)
- reinforcement learning (leert door te doen, reward and punishment methode)

2.4.1.2.1 Behaviour cloning

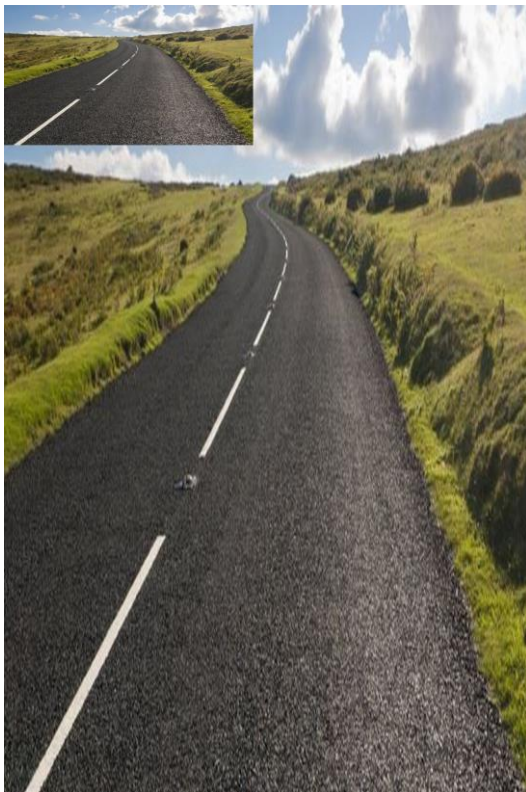
Behaviour cloning is een onderdeel van reinforcement learning. Het idee achter behaviour cloning is dat je het programma voortoont hoe het moet, waarna het programma het 'nadoet'. Het zoekt correlaties tussen de input en wat het voorbeeld doet. Terwijl hij kijkt naar het voorbeeld maakt hij zelf een voorspelling en vergelijkt deze dan met de uitkomst van het voorbeeld. Uiteindelijk zou het programma zelf logische beslissingen moeten kunnen nemen bij data die het nog nooit heeft gezien. Dit kan hij doen door gewoon te redeneren op basis van zijn ervaring. Hierdoor is het dus ook logisch dat hoe meer en beter je het model traint, hoe accurater het eindresultaat wordt.

Om Onward autonoom te laten rijden pas ik deze techniek toe. Het neural net van de code gebruik ik van een MIT-student die een kleiner gelijkaardig project tot stand bracht.

2.5 Drone

2.5.1 Superimpose

Zelfrijdende auto's gebruiken de superimpose-techniek om de bochten van een weg beter te bepalen. Aangezien de camera's op deze voertuigen dicht bij de grond gemonteerd zijn is het moeilijk om een nuttig beeld op te vangen. Met de superimpose-techniek neem je als het ware een foto boven op de originele afbeelding maar vanuit het vogelperspectief. In onderstaande afbeelding ziet u hier een voorbeeld van. Linksboven ziet u de originele afbeelding en er onder het resultaat.



FIGUUR 23: SUPERIMPOSE

2.5.1.1 Problemen

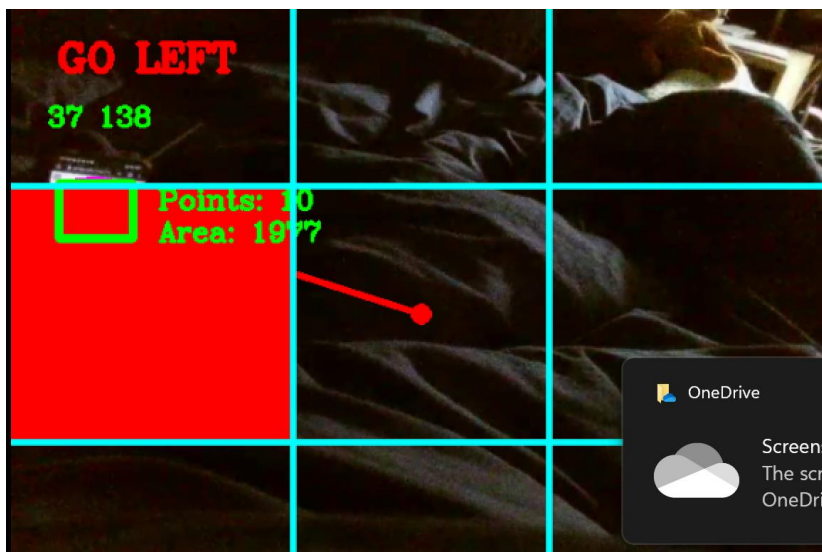
Mijn software voor deze toepassing draait perfect en vlot op een RPI. Dit leek me dan ook perfect om aan mijn gip toe te voegen. Ik hield er geen rekening mee dat dit maar één van de vele processen is dat de RPI moet berekenen. De beelden moeten dan nog gefilterd en verwerkt worden. Ik gebruik dus dit deel van mijn software niet in de afgewerkte versie.

2.5.1.2 Oplossing

Om dit proces te vervangen linkte ik een drone met het voertuig. Deze vliegt boven Onward en stuurt zijn camerabeelden door. De Tello drone wordt aangestuurd over wifi. De drone volgt mijn voertuig aan de hand van objecttracking software die bepaalde kleuren uitfiltert. De drone start in het voertuig. Wanneer het dak opent vlieg hij eruit en gaat naar zijn plaats achter het voertuig. Dan begint hij aan objecttracking.

2.5.2 Objecttracking

De drone filtert de camerabeelden op kleur met de colorfiltering software. Het beeld van de drone wordt dan in negen delen opgedeeld. Als het gewenste kleur gedetecteerd wordt berekent de drone de afstand van het kleur tot het middelste vak. Indien nodig worden de velocities aangepast. De drone blijft dus altijd even ver en perfect achter het object dat hij volgt.



FIGUUR 24: DRONE OBJECTTRACKING

De originele beelden worden dan nog eens gefilterd om de lijnen van de weg te detecteren. De drone coacht als het ware het voertuig tijdens zijn train periode.

2.6 Cloud

ThingSpeak is een soort van Facebook maar voor data. Velen gebruiken dit als cloud. Je kan elkaars data zien en gebruiken. Dit wordt bijzonder veel gebruikt bij IOT-toestellen. Ik gebruik dit om de temperatuur van de RPI's, batterijpercentage van de drone en commando's van de voice-assistent op te slaan. Deze gegevens zijn gemakkelijk in een mooie live grafiek te gieten.



FIGUUR 25: THINGSPEAK

2.7 Kubernetes

Dit is zonder twijfel het ingewikkeldste deel maar eenmaal dat je het doorhebt, zoals bij alles, is het super simpel.



FIGUUR 26: KUBERNETES

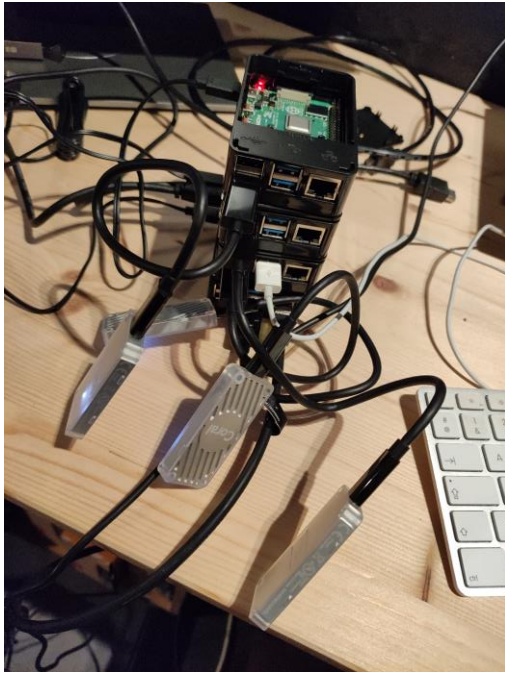
2.7.1 Wat is Kubernetes?

Kubernetes wordt afgekort met K8S. Deze 8 tussen de K en S komt van de acht letters tussen de twee letters. K8S is een open-source systeem dat automatisch containers plaats, schaaft en onderhoud. Dit was ontwikkeld door Google en wordt nu onderhouden door the Cloud Native Computing Foundation.

Deze alinea is voor mensen die het concept van containers nog niet kennen. Indien je dit wel kent sla dit deel gerust over. Containers zijn een soort van virtuele machine. Het doel van deze containers is om taken compleet van elkaar gescheiden en zonder elkaar te storen te laten lopen op een computer. Met volgend simpel voorbeeld schets ik kort hoe dit werkt. Stel je voor dat je GTA, Minecraft en Call of Duty wilt spelen. Je computer kan al deze last niet aan. Je bent toevallig een kenner van Kubernetes en je smijt GTA in een container, Minecraft in een tweede container en Call of Duty in een derde. Doordat de games nu compleet gescheiden van elkaar draaien kan je computer dit plots wel aan. Je ziet dat je games nogal haperen, dus je maakt voor iedere game een extra container. Zoals je kan raden lopen je games nog beter.

Kubernetes (K8S) was origineel gebouwd voor de cloud. Ik gebruik K3S, de lokale versie. Dit is een lichtere en snellere versie dat lokaal op een RPI geïnstalleerd kan worden. Je hebt wel nog internettoegang nodig om meerdere hardware toestellen met elkaar te verbinden.

Er is een master/pilot (Kubernetes gebruikt vliegtuigtermen) de rest zijn nodes (de andere RPI's). De master geeft de instructies, verdeelt de taken en laat de nodes het werk doen. Wanneer de nodes klaar zijn verzamelt de master de resultaten. In onderstaande bijlage ziet u een foto van mijn setup. De grijze aanhangsels zijn externe processoren gemaakt om machine learning modellen te moduleren.



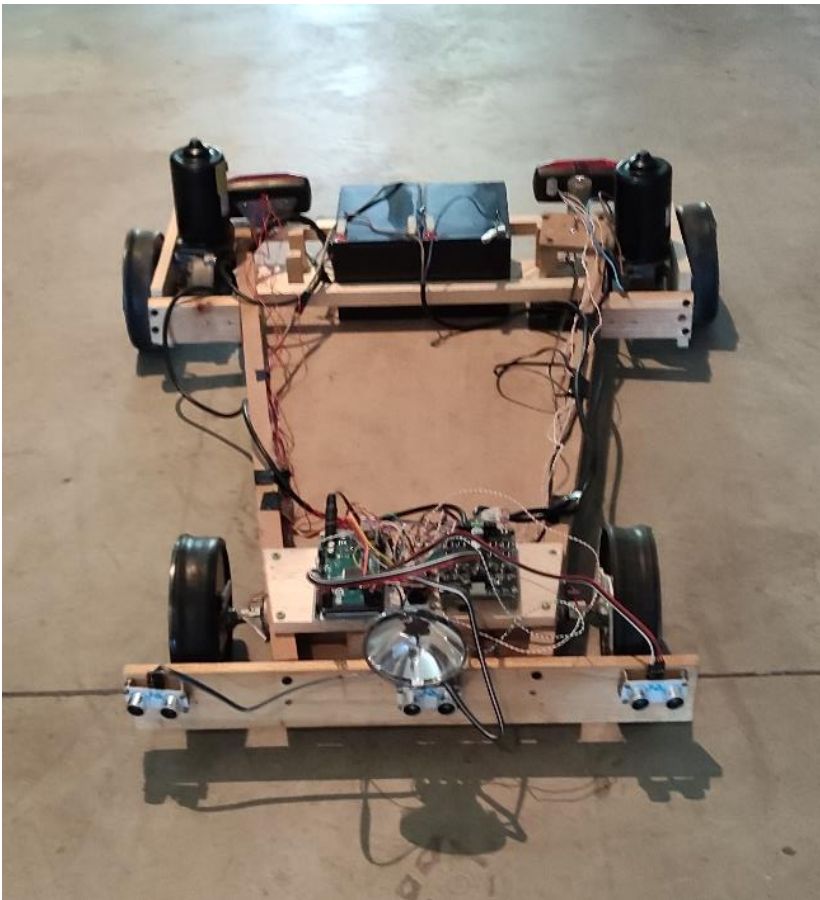
FIGUUR 27: K3S ONWARD

Om een actie uit te voeren maak je een manifest aan. Dit is een soort van tekstbestand waar je in verifieert wat je wilt vervoeren. Sommige processen zijn niet rechtstreeks te vervoeren met Kubernetes. Voor deze gevallen kan je Docker gebruiken om het proces te vermommen. Dit is vergelijkbaar met het vervoer van illegale goederen, je vermomt ze en poef niemand ziet het verschil. Het grote voordeel van Kubernetes is dat het systeem geautomatiseerd is en stel dat er een container beschadigd is wordt deze automatisch gerepareerd.

De software dat ik schreef verbond mijn vier RPI's met elkaar en verdeelde automatisch alle opdrachten. Het beste deel aan mijn software is het automatisch verzamelen van de verwerkte data. Dit is normaal een enorm irritant proces. Doordat alle opdrachten op aparte containers draaien die niets met elkaar of het toestel te maken hebben is het moeilijk om de data terug te krijgen.

3 Design

Mijn oorspronkelijk idee was om een deliveryvoertuig te maken. Alles was ontworpen om er een pizzadoos in rond te brengen. Het volledige voertuig meet 1,20 bij 0,50 meter en weegt rond de 20kg. In onderstaande afbeelding vindt u het eindresultaat.



FIGUUR 28: DELIVERYVOERTUIG

Ik was snel klaar dus besloot om een grotere versie te bouwen. Ik ontdekte deze oude stalen gocart bij mijn grootouders.



FIGUUR 29: GOCART

Ik strippte alles van de cart zodat alleen het chassis overbleef. Aangezien ik zelf in het voertuig wil passen sneedt ik het chassis in twee. Dit is de eerste keer dat ik met staal werk en de eerste keer dat ik las. Ik heb dit allemaal zelf thuis geleerd.



FIGUUR 30: GESNEDEN CHASSIS

Ik verlengde het chassis door een stalen buis met een net iets kleinere diameter in het chassis te kloppen en vast te lassen.

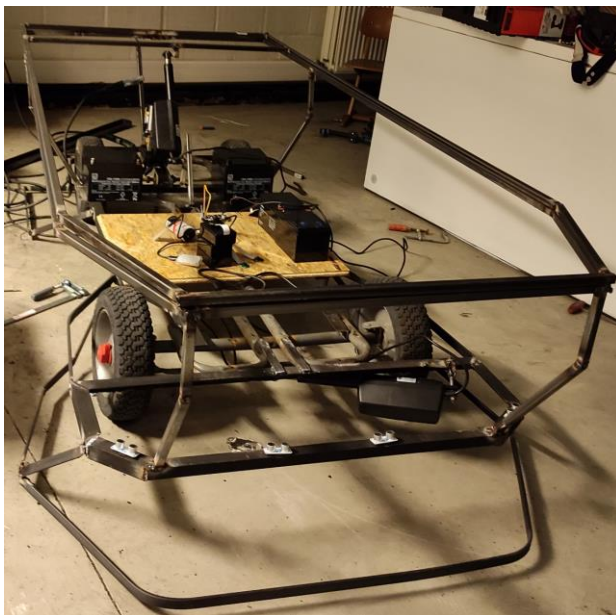


FIGUUR 31: VERLENGD CHASSIS

Ik laste een stalenframe op het chasis om het zitvlak te verbreden en te verstevigen.



FIGUUR 32: STALEN FRAME



FIGUUR 33: STALEN FRAME 2

Onderstaande afbeelding toont een voorlopig eindresultaat. Onder de zetel bevindt zich het besturingssysteem, de oplader, batterijen, motor voor de achterwielaandrijving en de voice-assistent.



FIGUUR 34: VOORLOPIG EINDRESULTAAT



FIGUUR 35: NYLON OVERTREK

Het is uiteindelijk de bedoeling om heel het voertuig in nylon in te pakken. Dit is een extreem licht en rekbaar materiaal ideaal voor de buitenkant. Ik haalde dit idee van een conceptauto van BMW.

Scan onderstaande QR-code om een video van deze conceptauto te zien.



4 Prijsberekening

In onderstaande bijlage vindt u spreadsheet met de prijsberekening. Het is mogelijk dat er iets hier en daar niet exact klopt. Sommige onderdelen zijn meer dan een jaar geleden aangekocht en zijn in prijs veranderd.

naam component	prijs per onderdeel	#	totaal onderdeel
Raspberry Pi 4 4gb	110	4	440
cytron motor driver 20A	46,2	1	46,2
cytron motor driver 10A	32,5	2	65
Raspberry Pi camera	28,5	4	114
camera	240	1	240
Google ml processor	90	4	360
loodzuur-batterijen	118	2	236
staal buis 2 m	21,4	3	64,2
staal driehoek dik 2 m	18,6	2	37,2
staal staaf 2 m	9,75	4	39
staal driehoek dun 2 m	14,6	2	29,2
staal vierkant 2 m	14,5	4	58
gocart	50	1	50
lineaire motor DC	440	2	880
DC-motor	880	1	880
drone	119,6	1	119,6
respeaker	58,9	2	117,8
box	340	1	340
peltier-element	64,21	1	64,21
koeling	5,6	4	22,4
scherm lcd	162,8	4	651,2
ultrasonische sensor	6,58	4	26,32
kabels	40	1	40
DC-spanning convertor	128	1	12,8
totaal	4 933,13 euro		

Algemeen besluit

Onward is een elektrisch voertuig van 2,10 bij 1,50 meter dat zelfstandig rijdt. Ik ben trots om te zeggen dat het een level 5 behaalt op autonoom rijden. Het voertuig leert rijden op basis van voorbeelden en wordt gecoacht door de drone. Na verloop van tijd kan het in ongeziene situaties zelf keuzes maken gebaseerd op zijn ervaring. Onward is doorheen het jaar veel aangepast en uitgebreid. Ik ben blij dat ik ook gezichtsherkenning en de link met de drone als extra kon toevoegen. Ik probeerde ook zoveel mogelijk te recyclen. Ik haalde de koeling en schermen uit kapotte laptops. De lineaire motoren komen uit oude bureaustanden en het chassis is een oude gokcart.

Terwijl ik dit project uitwerkte, leerde ik veel bij. Ik heb leren lassen, nieuwe modules gebruiken, productiecode en nieuwe AI-modellen leren schrijven. Bij elk onderdeel botste ik op nieuwe moeilijkheden zoals bibliotheekversies die elkaar niet ondersteunden en motor drivers die in mijn handen ontploften. Zo moest ik telkens op zoek naar een creatieve oplossing. Dit duurde soms een dag of soms helaas wel een maand.

Ik hoopte wel verder geraakt te zijn. Ik zou nog enorm veel dingen willen verbeteren en toevoegen. Het zelfrijdende deel is nog in de beginfase. Het voertuig is minder goed afgewerkt qua design en metaalwerk. De voice-assistent is momenteel niet bruikbaar terwijl dit een supertoffe extra was. De superimpose software en het AI-model om hoeken te voorspellen kon ik niet gebruiken door tekort aan rekenkracht.

Wat ik zou verbeteren is dan net zoals Tesla een digital twin maken. Om dit waar te maken zou ik overal sensoren plaatsen. Dit maakt het mogelijk om situaties te moduleren en te voorspellen wanneer onderdelen aan onderhoud toe zijn. Tesla's zijn rijdende supercomputers. Ook op dit vlak wil ik verbeteren wat ik met mijn Kubernetes software zou realiseren. Ik ben tegen het verspillen van energie dus wil ik ook de warmte via een Peltier element en de zon via een zonnepaneel terug in elektriciteit omzetten.

Na een bijna volledig schooljaar kan ik vast en zeker, desondanks een paar teleurstellingen, zeggen dat ik trots ben op mijn gip en vooral het proces hierachter.

Bijlagen

Bijlage 1: curriculum vitae

Bijlage 2: lijst van afbeeldingen

Bijlage 3: schema's


Bijlage 4: datasheets


Bijlage 5: code


Bijlage 1: curriculum vitae


Henri Victor Lahousse


Personalia


 Henri Victor Lahousse

 henri.lahousse@leerling.pti.be

 0478 10 66 16

 Watermolen straat 49
8500 Kortrijk

 17 maart 2004

 Kortrijk

Vaardigheden

A.I. (artificial intelligence)

M.L. (machine learning)

IOT

elektronica

Python


Talen


Nederlands

Engels


Frans


Hobby's en interesses


 Chiro


 Trompet


Eigenschappen

 leergierig

 team player

 problemen definiëren

 oplossingen zoeken

 creatief

Opleidingen

elektriciteit elektronica

PTI, Kortrijk

basiskennis opgedaan van elektriciteit, elektronica, automatisatie en regeltechnieken Mijn gip heet Onward, een zelfrijdende auto. Met speciaal voor dit project ontwikkelde stemherkenning, gezichtsherkenning en sound source location beleef je een werkelijk unieke ervaring.

sep. 2020 - jun. 2022

Latijn wiskunde

Guldensporencollege, Kortrijk

sep. 2016 - jun. 2020

Werkervaring

junior software engineer

Johan Langenbick, Gent

Mixly verhuurt toestellen die gepersonaliseerde cocktails in minder dan twee seconden uitschenkt en ik hielp bij het toegankelijker maken van deze toestellen. Ik voegde een extra functie toe, waardoor klanten via qr-codes met het toestel kunnen communiceren.

dec. 2020 - heden

electronic engineer

Bernard Lahousse, Gent

In het labo van Foodparing hadden ze nood aan een toestel dat op de seconde en de millimeter precies substanties kan afnemen.

nov. 2020 - dec. 2020

Cursussen

A.I. (artificial intelligence)

gevolgd bij Datacamp en certificaat behaald

M.L. (machine learning)

gevolgd bij Datacamp en certificaat behaald

Python

gevolgd bij Datacamp en certificaat behaald

Kubernetes

heden

SQLite

heden

Profiel

Sinds mijn 10de verjaardag ben ik continu bezig geweest met eigen projecten. Dit, samen met mijn studies en het samen oplossen van problemen met vrienden heeft me een hele reeks vaardigheden opgeleverd. Ik ga er altijd 110 procent voor en wil de rest van mijn leven nieuwe dingen blijven bijleren. U kunt volledig op mij rekenen en erop vertrouwen dat ik voor alle problemen een zo goed mogelijke oplossing zal vinden.

Certificaten

Challenge for AI&Business

Aan de topuniversiteit Esade in Barcelona behaalden we de eerste plaats. We wonnen door je smartwatch en smartfrigo slim te laten communiceren. Zo genereerden we gepersonaliseerde gerechten op basis van je humeur, gezondheid en frigo-inhoud.

Onward

| 40 |

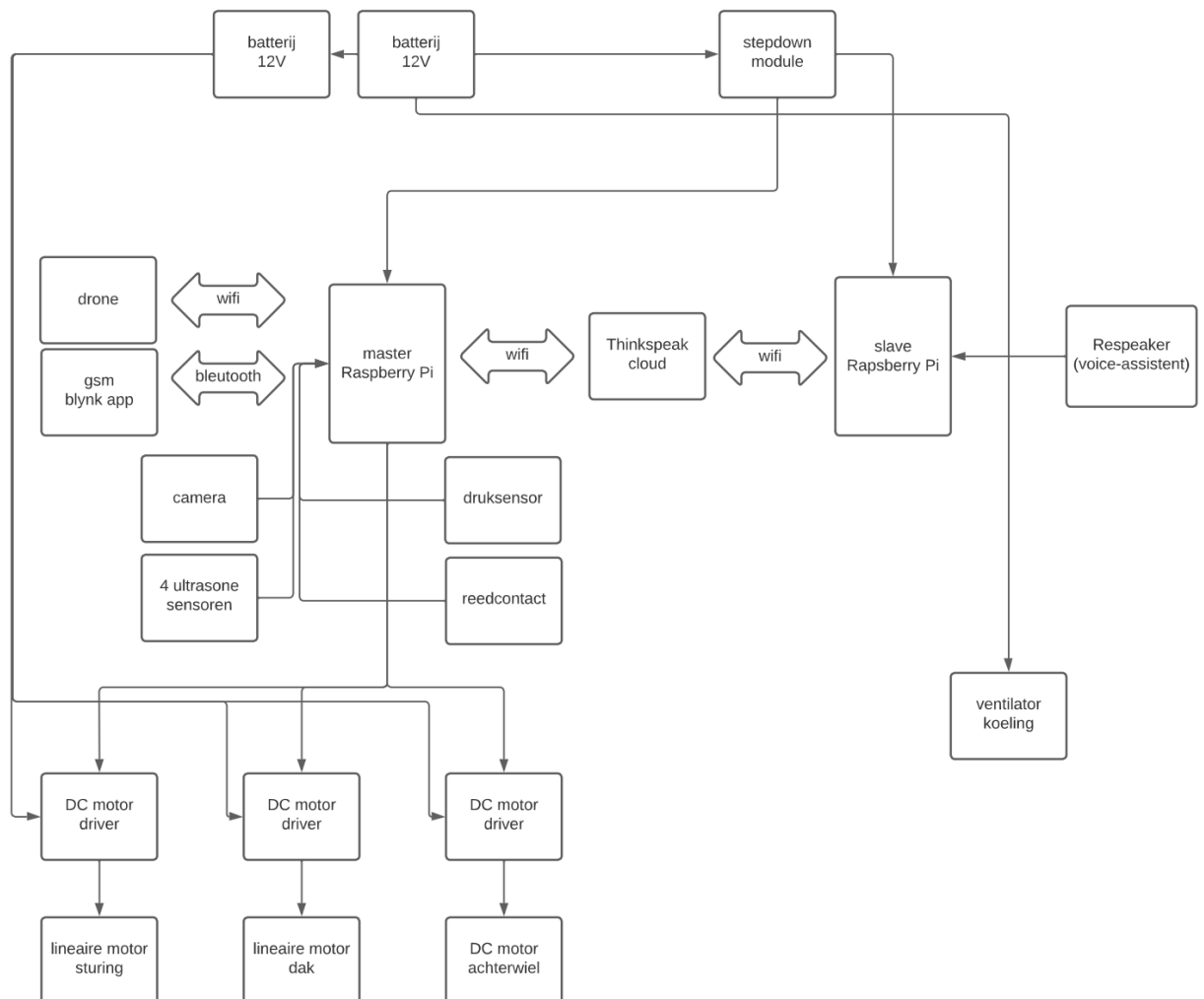
Bijlage 2: lijst van afbeeldingen

Figuur 1: levels selfdriving	1
Figuur 2: schets	5
Figuur 3: RPI-camera	6
Figuur 4: respeaker	7
Figuur 5: ultrasonische sensor	8
Figuur 6: Tello drone	8
Figuur 7: druksensor.....	9
Figuur 8: reedcontact	9
Figuur 9: Raspberry Pi 4.....	10
Figuur 10: DC-motor.....	11
Figuur 11: lineaire motor.....	12
Figuur 12: lineaire motoren setup	12
Figuur 13: motor driver 20 A 30 V	13
Figuur 14: motor driver 10 A 30 V	13
Figuur 15: loodzuur-batterij	14
Figuur 16: spanningsregelaar.....	14
Figuur 17: Deepface	15
Figuur 18: lineaire motoren achterwielaandrijving	16
Figuur 19: kleurfiltering.....	17
Figuur 20: EDA/wolf	18
Figuur 21: confusion matrix.....	19
Figuur 22: respeaker	21
Figuur 23: superimpose	25
Figuur 24: drone objecttracking	26
Figuur 25: ThingSpeak	27
Figuur 26: Kubernetes	27
Figuur 27: K3S Onward	29
Figuur 28: deliveryvoertuig.....	30
Figuur 29: gocart	31
Figuur 30: gesneden chassis	31
Figuur 31: verlengd chassis.....	32

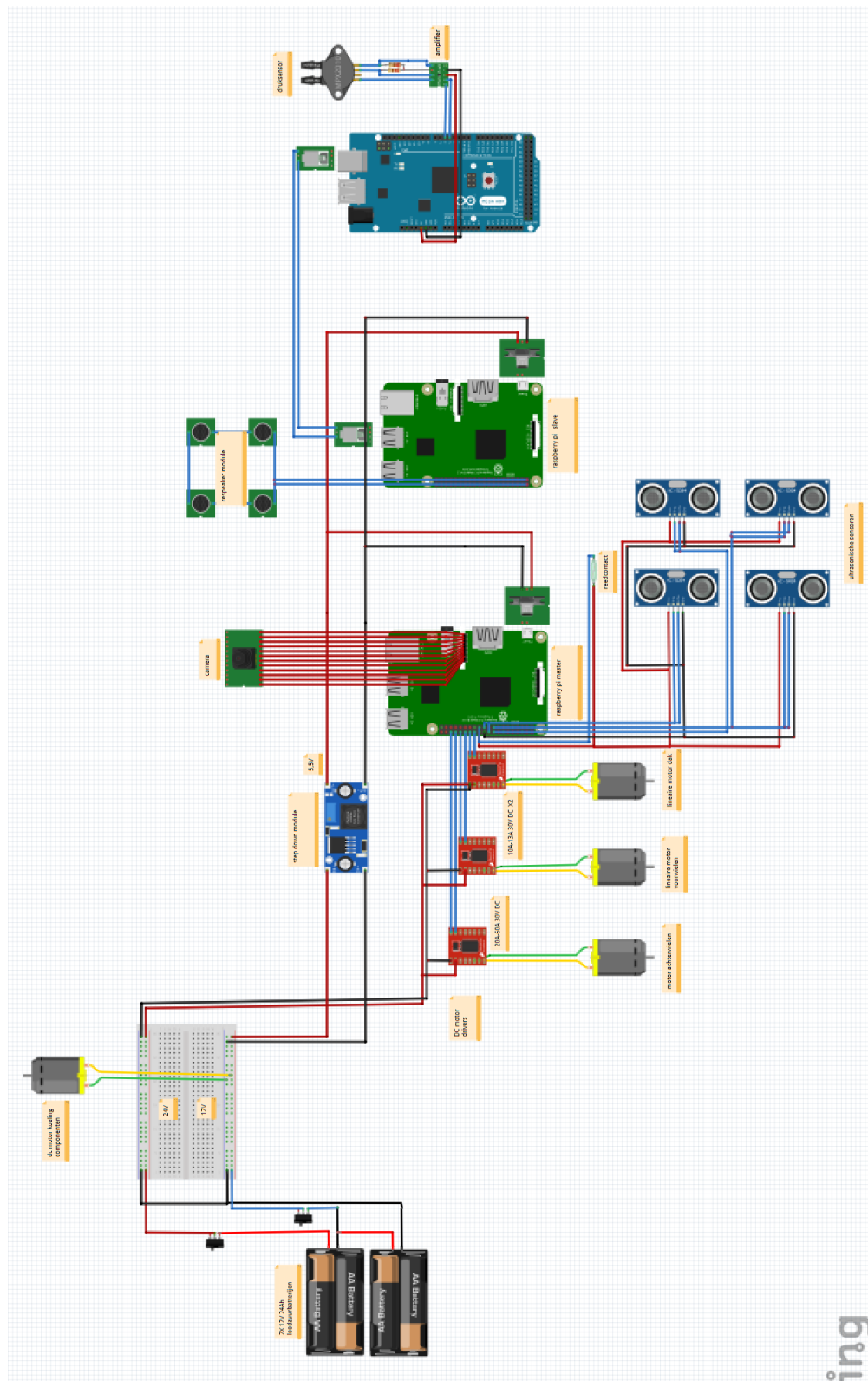
Figuur 32: stalen frame	33
Figuur 33: stalen frame 2	33
Figuur 34: voorlopig eindresultaat	34
Figuur 35: nylon overtrek	35

Bijlage 3: schema's

Blokschema



Getailleerd schema



Onward

Bijlage 4: datasheets

Raspberry Pi 4



Loodzuur-batterijen 12 V 24 Ah



Cytron motor driver 20 A-60 A 30 V DC



Cytron motor driver 10 A-13 A 30 V DC



Bijlage 5: code

Scan de QR-code om de code van het project te zien.



Bronvermelding

Internetbronnen

- https://wiki.seeedstudio.com/ReSpeaker_4_Mic_Array_for_Raspberry_Pi/(2021-09-01/2022-05-06)
- <https://www.sqlitetutorial.net/sqlite-python/>(2021-09-01/2022-05-06)
- <https://www.youtube.com/watch?v=X9fSMGkjtug&t=471s>(2021-09-01/2021-12-20)
- <https://stackoverflow.com/>(2021-09-01/2022-05-06)
- <https://app.datacamp.com/learn>(2021-09-01/2022-05-06)
- <https://thingspeak.com/>(2021-09-01/2022-05-06)
- <https://picovoice.ai/docs/>(2021-09-01/2022-05-06)
- <https://console.picovoice.ai/>(2021-09-01/2022-05-06)
- <https://github.com/>(2021-09-01/2022-05-06)

Afbeeldingen

- Figuur 1: <https://www.synopsys.com/automotive/autonomous-driving-levels.html>
- Figuur 2: eigen illustratie
- Figuur 3: <https://thepihut.com/products/raspberry-pi-camera-module>
- Figuur 4: <https://www.kiwi-electronics.nl/nl/respeaker-4-mic-array-voor-raspberry-pi-3288>
- Figuur 5: <https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/>

- Figuur 6: <https://www.derekenwinkel.nl/dji-tello-drone-edu-uitvoering.html>
- Figuur 7: <https://forum.arduino.cc/t/hx711-amplifier-with-50kg-load-cell-values-fluctuating-constantly/931426>
- Figuur 8: <https://www.amazon.nl/profitec-SAS-4-Reed-contact/dp/B000NZIM0W>
- Figuur 9: <https://www.reichelt.com/nl/nl/raspberry-pi-4-b-4x-1-5-ghz-4-gb-ram-wlan-bt-ram-ram-wlan-rasp-pi-4-b-4gb-p259920.html?CCOUNTRY=662&LANGUAGE=nl&&r=1>
- Figuur 10: https://en.wikipedia.org/wiki/DC_motor
- Figuur 11: <https://www.firgelliauto.com/nl/blogs/news/inside-a-linear-actuator-how-a-linear-actuator-works>
- Figuur 12: eigen illustratie
- Figuur 13: <https://www.cytron.io/p-20amp-6v-30v-dc-motor-driver>
- Figuur 14: <https://www.robotshop.com/be/nl/cytron-13a-5-30v-enkele-gelijkstroommotorbesturing.html>
- Figuur 15: <https://benl.rs-online.com/web/p/lead-acid-batteries/1501559>
- Figuur 16: https://www.conrad.be/nl/p/makerfactory-spanningsregelaar-mf-6402402-1-stuk-s-2134134.html?t=1&utm_source=google&utm_medium=surfaces&utm_term=2134134&utm_content=free-google-shopping-clicks&utm_campaign=shopping-feed&vat=true&gclid=CjwKCAjwve2TBhByEiwAaktM1Jnzhdj4XpmKVXiJpBYr88dTRdMpzukdbBzcfJBZHy2ouTL_2e9gMxoCiawQAvD_BwE&tid=16860426636_pla-2134134&WT.srch=1
- Figuur 17: <https://github.com/serengil/deepface>
- Figuur 18: eigen illustratie
- Figuur 19: eigen illustratie
- Figuur 20: https://stringfixer.com/nl/European_wolf

- Figuur 21: <https://shravantandale456.medium.com/importance-of-confusion-matrix-in-machine-learning-and-cybersecurity-80e67f5858fb>
- Figuur 22: eigen illustratie
- Figuur 23: eigen illustratie
- Figuur 24: eigen illustratie
- Figuur 25: <https://www.home-assistant.io/integrations/thingspeak/>
- Figuur 26: <https://www.combell.com/nl/blog/wat-is-kubernetes/>
- Figuur 27: eigen illustratie
- Figuur 28: eigen illustratie
- Figuur 29: eigen illustratie
- Figuur 30: eigen illustratie
- Figuur 31: eigen illustratie
- Figuur 32: eigen illustratie
- Figuur 33: eigen illustratie
- Figuur 34: eigen illustratie
- Figuur 35: eigen illustratie

