­

Projectbeschrijving Project 2

**“Haven en transport”**

Inhoudsopgave

[Inleiding 3](#_Toc132244770)

[Smart Digital Ports Of The Future 2023 conference 16th - 17th May 2023 3](#_Toc132244771)

[Opdracht 3](#_Toc132244772)

[Doel 3](#_Toc132244773)

[Onderdelen 4](#_Toc132244774)

[Data verzamelen 4](#_Toc132244775)

[Data interpreteren 4](#_Toc132244776)

[Neuraal netwerk ontwerpen 4](#_Toc132244777)

[Trainen, testen en valideren 4](#_Toc132244778)

[Zorgdragen voor de veiligheid 5](#_Toc132244779)

[Deliverables 6](#_Toc132244780)

[Projectevaluatie 7](#_Toc132244781)

[IT voor AI 7](#_Toc132244782)

[Toegepaste wiskunde 7](#_Toc132244783)

[Professionele vaardigheden 7](#_Toc132244784)

[Toetsing 8](#_Toc132244785)

[Hints, Tips & Tricks 9](#_Toc132244786)

[Daily Kata Starter Questions 9](#_Toc132244787)

[Bronnen 9](#_Toc132244788)

# Inleiding

Als je aan Rotterdam denkt dan denk je onder andere aan de Rotterdamse Haven. De haven van Rotterdam is het grootste haven- en industriecomplex van Europa. Tussen 1962 en 2004 was de haven van Rotterdam zelfs de grootste haven van de wereld. Rotterdam heeft de ambitie om de ‘smartest port’ te zijn en hiermee neemt de stad het voortouw in de digitale transformatie van haven en logistiek. Daarnaast heeft de haven als doel om in 2030 schepen autonoom door de haven te laten varen. Dit tweede project richt zich daarom ook op het thema Haven en Transport.

## Smart Digital Ports Of The Future 2023 conference 16th - 17th May 2023

Wellicht een job opportunity of gewoon interessant… Geef je op via [https://sdp.ptievents.com](https://sdp.ptievents.com/) (overlapt met lessen, in overleg met de docenten)

# Opdracht

Het autonoom laten rijden van voertuigen en varen van schepen is een belangrijk doel om het werk in de haven efficiënter te maken. Hiermee kunnen kosten bespaard worden en kan er meer werk worden verzet. Maar het autonoom laten rijden van voertuigen en varen van schepen moet wel veilig zijn. Tricky: wie is er verantwoordelijk als er iets misgaat? In dit project krijg je de opdracht een voertuig autonoom veilig te laten rijden in een digitale omgeving. Het autonome voertuig, een auto in dit geval, kan tijdens het rijden data verzamelen door middel van twee typen sensoren, een LIDAR en een SONAR. Je verzamelt de data van de auto om hiermee een neuraal netwerk te trainen, zodat de auto zelfstandig en zo veilig mogelijk rond kan rijden in de simulatieomgeving.

# Doel

Dit project bestaat uit meerdere onderdelen:

1. Het verzamelen van LIDAR en SONAR data
2. Het interpreteren van de verzamelde data
3. Het ontwerpen van een effectief neuraal netwerk
4. Het trainen, testen en valideren van het rijgedrag van de autonome auto
5. Zorgdragen voor de veiligheid van de auto

Het doel is om deze onderdelen stap voor stap uit te werken en ze op een logische manier aan elkaar te knopen in een bruikbaar *proof of concept*.

# Onderdelen

## Data verzamelen

Om snel te kunnen starten vind je in de aangeleverde [broncode](https://github.com/AlxcNL/MakeAIWork2/tree/main/project/simpylc) een tweetal voorgedefinieerde tracks ([*lidar.track*](https://github.com/AlxcNL/MakeAIWork2/blob/main/project/simpylc/simulations/car/lidar.track) en [*sonar.track*](https://github.com/AlxcNL/MakeAIWork2/blob/main/project/simpylc/simulations/car/sonar.track)), en een voorgeprogrammeerde autonome auto ([*hardcoded\_client.py*](https://github.com/AlxcNL/MakeAIWork2/blob/main/project/simpylc/simulations/car/control_client/hardcoded_client.py)). Deze voorgeprogrammeerde autonome auto rijdt al zelfstandig en verzamelt tijdens het rijden data. Afhankelijk van de configuratie verzamelt de auto datapunten via LIDAR (16 scanning beams) en SONAR (3 scanning beams). Deze data wordt opgeslagen in een json-bestand. Bestudeer de code voor meer details, en bestudeer ook de hardcoded client.

Als uitbreiding is er ook een [handmatig te besturen auto beschikbaar](https://github.com/hakoptak/SimPyLC/tree/Manual-Car/simpylc/simulations/car_manual_flex_track) (via de pijltjestoetsen). Deze verzamelt ook data via LIDAR en SONAR en stelt je in staat om je autonome auto te laten rijden als een mens. Je kunt hiermee bijvoorbeeld bochten afsnijden, schokkend optrekken en afremmen of als een dronkenman over de weg slingeren. Let op: begin hier pas aan als je een systematische aanpak hebt uitgewerkt op basis van de voorgeprogrammeerde auto. Leren op basis van handmatige besturing heeft ook invloed op het ontwerp van je neurale netwerk.

## Data interpreteren

Bestudeer de verzamelde data om beter inzicht te krijgen in wat de rijdende auto ‘ziet’ en hoe de auto hierop reageert (sturen, gasgeven).

## Neuraal netwerk ontwerpen

Ontwerp op basis van bovenstaande analyse een aantal *fully connected* neuraal netwerken. Dit mag met PyTorch maar als je liever zelf een model programmeert dan mag dat ook. Het doel is hier om een effectief netwerk te vinden dat de opdracht kan uitvoeren. Begin daarom met een eenvoudig netwerk en breid het netwerk uit als daar een noodzaak toe is. Meer *weights* & *biases* betekent meer berekeningen en dus meer energieverbruik en een langere *inference* tijd (al zal dat hier meevallen).

## Trainen, testen en valideren

Train je ontworpen netwerk(en) (het brein van je autonome auto) op basis van de verzamelde data (*supervised learning*). Houd hierbij rekening met onder- en overfitten. Houd systematisch bij welke *hyperparameters* je gebruikt voor welk ontwerp en op welke track. De ultieme test is natuurlijk om de auto te laten rijden om een zelfontworpen track. Maak hiervoor één of meerdere *.track* bestanden aan.

## Zorgdragen voor de veiligheid

De auto kan zich op twee manieren orienteren in de ruimte: via LIDAR en via SONAR. Welke methode (LIDAR of SONAR) is het nauwkeurigst en daarmee het meest betrouwbaar?

Het kan gebeuren dat de zelfrijdende auto na een aantal ronden alsnog de weg kwijt raakt. Dat is natuurlijk onwenselijk maar nog belangrijker: kun je dit ook detecteren? En zo ja: hoe ga je hier dan mee om? Kun je de auto vlot afremmen als hij buiten de gebaande paden gaat? Of scherp bijsturen? Of…?

# Deliverables

* Vastlegging van inzichten en experimenten in een Jupyter Notebook
* Korte onderbouwing van gemaakte keuzes (ontwerp, implementatie)
* Overzicht van de testresultaten (nauwkeurigheid, foutmarges) en advies (bruikbaarheid?)
* Uitdraai van de git historie (geeft een beeld van het software development proces)
* De uiteindelijke broncode moet als zip-bestand worden ge-upload naar je eigen Teams kanaal. Volg hiervoor de stappen op <https://github.com/AlxcNL/MakeAIWork2/blob/main/PROJECT_EXPORT.md>

Upload bovenstaande onderdelen **vóór de deadline van maandag 8 mei, 12:00 uur** naar:

Teams > Applied Artificial Intelligence - Make IT Work >  
Jouw Private Channel > Files > Deliverables periode 2 > Project 2

Zorg daarbij voor de volgende mappenstructuur:

* Notebooks
* Onderbouwing
* Advies
* Git-historie
* Broncode

# Projectevaluatie

Het doel van de projectevaluatie is tweeledig. Aan de ene kant wordt in dit document aandacht besteed aan de meta-functionele overwegingen die gemaakt worden tijdens het project, daarnaast wordt in dit document aandacht besteed aan de individuele ervaringen gedurende dit project. Meta functionele eisen: het gaat hier om bedrijfsmatige, ethisch-maatschappelijke, juridische, regulerings- en technische kaders die de directe functionele eisen overstijgen. Bij toetsing ligt de nadruk op de volgende onderwerpen c.q. activiteiten:

### IT voor AI

1. Git(hub) projectmatig inzetten
2. Basis operaties Linux, waaronder bash scripts

### Toegepaste wiskunde

1. Begrip van de topologie van een fully connected neuraal netwerk
2. Begrip van het leerproces van een neuraal netwerk
3. PyTorch toepassen in specifieke use case

### Professionele vaardigheden

1. Kennis putten uit onvolmaakte bronnen door kritisch te lezen en meerdere bronnen te combineren
2. Planmatig en zelfstandig werken
3. Het vinden van een werkbare balans tussen snel versus beheersbaar ontwikkelen bij een proof of concept

## Toetsing

De toetsing is formatief. Het doel is niet beoordeling, maar het geven van feedback en adviezen door de docent(en). Hierbij ligt de nadruk op de volgende zaken:

1. Uitwerking van opdrachten die onderdeel zijn van de lessen
2. Uitwerking van de project-opdracht, inclusief inhoud onderzoeksrapportage  
   (Jupyter Notebook), broncode, git historie, testresultaten en advies
3. Je moet de uitkomsten van je autonome auto kunnen verklaren

# Hints, Tips & Tricks

**Simulatie** – Gegeven is een eenvoudige simulatie van een auto, die wel aan de natuurwetten voldoet: Als je te hard rijdt, vlieg je uit de bocht. De gesimuleerde auto kan worden bestuurd via een socket-verbinding. Ook de data stroomt terug via de socket-verbinding. Begrip van de werking van de simulator zelf is voor uitvoering van dit project niet nodig.

**Hardcoded client** **–** Je vindt de code van de voorgeprogrammeerde auto in het bestand *hardcoded\_client.py* in de folder *car/control\_clients*. Gebruik deze code tevens als voorbeeld voor communicatie met de gesimuleerde auto, die de rol van server vervuld.

**Hulp vragen** – Schroom niet gerichte vragen te stellen aan docenten en mede-cursisten. Zorg ervoor dat je de hulpvraag goed formuleerd en uiteen kunt zetten wat je stappen tot dan toe zijn geweest.

Zorg dat je zo snel mogelijk een werkend prototype hebt en ga die vervolgens tunen. Maak voor experimenten / aanpassingenen een feature branch zodat de main branch altijd een werkende versie bevat.

# Daily Kata Starter Questions

1. Wat is het einddoel? Wat heb en kun je aan het einde van het project?
2. Waar sta je nu?
3. Zijn er obstakels die je verhinderen om het doel te bereiken?   
   Zo ja, van welke ondervind je op dit moment hinder?
4. Gegeven het einddoel en waar je nu staat, wat is de volgende stap?
5. Hoe kun je de stap dusdanig klein maken dat je die vandaag in zijn geheel kunt nemen?

Zie [https://traccsolution.com/blog/toyota-kata](https://traccsolution.com/blog/toyota-kata/) voor achtergrondinformatie.

# Bronnen

<https://www.wired.co.uk/article/rotterdam-port-ships-automation>