**Testrapport project 1**

**Kevin van der Meer**

**Applied Artificial Intelligence – cohort 1 2022/2023**

**1064998**

**Welke scanningsmethode is het meest geschikt en waarom?**

De scanningsmethodes die we in dit project hebben vergeleken zijn Lidar en Sonar. Sonar scant de omgeving af op obstakels door middel van geluidsgolven, waar lidar gebruik maakt van lichtgolven. Kort gezegd komt het erop neer dat lidar werkt met de snelheid van het licht, waar sonar werkt op de snelheid van geluid. Lidar is dus vele malen sneller dan sonar. In onze simulatie heeft de sonar 3 scanners, en de lidar 16 scanners, die beiden een kijkhoek vooruit van 120 graden (graad voor graad) afgaan. De lidar is hierdoor veel sneller en accurater dan de sonar, en is dus per definitie het meest geschikt om een gesimuleerde auto te trainen om zelf te rijden over een circuit. In mijn model werkt sonar echter beter dan lidar (lidar heb ik zelfs niet aan de praat gekregen), en dat heeft te maken met de hogere complexiteit van de input (16 inputs in plaats van 3), maar ook met het circuit. Op het gegeven lidar circuit is er (linksonderaan) een punt waar de auto te weinig input krijgt doordat de auto zich meer dan 20 meter van de dichtstbijzijnde pion bevindt. Dit zou op te lossen moeten zijn door meer obstakels/pionnen toe te voegen aan het circuit of de simulatie zo in te stellen dat er verder vooruit gekeken wordt dan 20 meter. Als we uitgebreid de tijd zouden nemen om het circuit en/of de instellingen van de simulatie te optimaliseren, zou de lidar scanningsmethode het uiteindelijk beter doen dan de sonar.

**Welke aansturingsmethode werkt het beste en waarom?**

In mijn geval werkt de TensorFlow aansturing wel, en de SciKit Learn aansturing niet. Ik heb het idee dat Tensorflow een uitgebreidere library is, waardoor ik denk dat Tensorflow uiteindelijk het beste werkt.

**Hoe is veiligheid gegarandeerd in alle gevallen?**

De veiligheid is allereerst gegarandeerd omdat SimPyLC een simulatieomgeving is, waarin middels een header duidelijk is aangegeven dat het software is welke niet ontworpen/ bestemd is voor het ontwerpen van ‘real world’ toepassingen. Daarnaast is de hardcoded aangedreven door instructies, maar is er te weinig input bestaat de kans dat het autootje van de baan of tegen een pion op rijdt. Ook in de driving agent zijn er geen veiligheidsgaranties en kunnen obstakels geraakt worden, afhankelijk van hoe goed het model getraind is.

**Welke aansturingsmethode is het veiligst en waarom?**

Bij een voldoende getraind model (waarbij er altijd voldoende input vanuit de sensoren komt tijdens het trainen) is uiteindelijk de driving agent veiliger dan de hardcoded.

**Is de opzet van de codebase modulair? Hoe aan te tonen?**

De opzet van mijn codebase is zoveel als mogelijk modulair. Wel is de code geschreven in Python. Als ik eraan toe gekomen zou zijn om naast een Tensorflow trainingsmodel ook een goed werkend SciKit Learn model te bouwen zou ik dit met enkele aanpassingen en het importeren van andere libraries in de python file kunnen implementeren.