# Portfolio

## Edge AI

---

**Opleiding: Elektronica-ICT, Internet of Things**

**Academiejaar: 2023-2024**

Hulsbergen Sten

Haddouchi Hassan, Vanhulle Geert

# Inhoud

# 1 Theorie

Bij de theorie waren we met drie die de theorie niet bij hoefden te wonen, waaronder ik één van ben. De cursus die gebruikt is komt van de Minor AI, wat ik dus opgenomen heb. Dit betekend dus dat ik dit al gezien heb. Daarnaast heb ik vorig jaar als extra vak ML-Principles opgepakt voor studiepunten op te vullen. Deze cursus was toen ook de cursus van AI-Principles uit de Minor AI. Voor beide vakken ben ik ook geslaagd geweest, wat betekend dat zo veel op theoretisch vlak al goed gekend is.

## 1.1 AI/ML cursussen

Wat wel nieuw is, is de verder uitgewerkte ML cursus en de twee cursussen over Edge AI. Ik heb de ML cursus eens gelezen en deze bevat een grotere uitleg over onderdelen die kort of zelfs nog niet aan bod zijn gekomen in de AI-Principles cursus. Persoonlijk vind ik vooral de extra informatie over Convolutionele Neurale Netwerken, het trainen en testen en Deep Reinforcement Learning in deze cursus het interessants.

## 1.2 Edge AI-cursussen

De twee cursussen over Edge zijn voor mij alleszins het interessants aangezien we veel gebruik maken van MCU’s en SBC’s. De verschillen tussen een CPU, GPU en TPU waren al grotendeels bekend vanwege voorkennis en persoonlijke interesse, maar de wat Edge AI is, de gebruiksmogelijkheden en voor- en nadelen zijn altijd leuk om te weten.

# 2 Labo's

Aangezien bij de Minor AI ook een vak Neural Networks zit, was het maken van deze labo's niet moeilijk aangezien dit niet nieuw was.

## 2.1 Labo 1

### 2.1.1 Conclusie

Dit labo was kennis maken met Tensorflow en het werken met Matrixjes van Numpy. Hier ben ik snel doorheen gekomen, er zat niets nieuw in.

## 2.2 Labo 2

### 2.2.1 Deel 1

Dit labo bevat niets nieuw, hier ben ik snel doorheen gegaan. Dit labo had ik ook gemaakt in R voor AI-Principles.

### 2.2.2 Deel 2

Stap 1: Data verzamelen en importeren

Stap 2: Data verkennen

Stap 3: Data Preprocessing

### 2.2.3 Conclusie

Hier ben ik vergeten om bij stap 2 het gemiddelde, de mediaan, standaardafwijking en percentielen te berekenen. Wederom was er niet echt iets nieuw in dit labo buiten het gebruik maken van *"StandardScaler()".* Hier ging ik ook snel doorheen.

## 2.3 Labo 3

### 2.3.1 Deel 1

In dit deel moest ik een aantal vragen oplossen en de screenshots ervan doorsturen. Hierin moest ik data exploratie doen, een model maken en trainen, valideren of het model klopt en als laatste het model aanpassen door extra variabelen te gebruiken voor betere voorspellingen.

### 2.3.2 Deel 2

In dit deel werd het inladen van verschillende soorten data, waaronder afbeeldingen, CSV-data en Numpy data, bekeken.

### 2.3.3 Deel 3

In dit deel wordt het hele process van data exploratie, model trainen, evalueren en het plaatsen in Google Cloud omgeving bekeken.

### 2.3.4 Conclusie

Hier zijn deel 1 en 2 niet nieuw maar deel 3 wel. Het is best interessant om te weten hoe een model in de Cloud wordt geplaatst en dat dat zelfs een mogelijk is.

## 2.4 Labo 4

### 2.4.1 Deel 1

In dit deel van het labo werd a.d.h.v. een dataset van keras, dat bestaat uit getallen 0 tot 9, een model gemaakt. Hier moest ik de mnist data inladen, vervolgens de data opsplitsen in train en testdata en daarna wordt een model gemaakt met een aantal zelf in te stellen lagen. Daarna wordt het model gecompileerd, getraind, geëvalueerd en getest.

### 2.4.2 Deel 2

In dit deel van het labo wordt hetzelfde gedaan als deel 1, maar met kleine aanpassingen wel. Deze keer werd een extra dataset van keras gebruikt, dat bestaat uit het alfabet, namelijk emnist.

### 2.4.3 Conclusie

Ik heb het model van deel 1 rond de 97% en het model van deel 2 rond de 90% gekregen, dit was uiteindelijk niets nieuw waardoor ik hier snel doorheen ging.

## 2.5 Labo 5

### 2.5.1 Deel 1

In dit deel wordt een dataset van afbeeldingen van 10 dingen ingeladen van keras. Een model opgebouwd, getraind, gecompileerd en getest. Als laatste wordt het model opgeslagen en ingeladen om ergens anders het model te kunnen gebruiken.

### 2.5.2 Deel 2

In dit deel wordt een dataset van afbeeldingen van 100 dingen ingeladen van keras. Een model opgebouwd, getraind, gecompileerd en getest. Als laatste wordt het model opgeslagen en ingeladen om ergens anders het model te kunnen gebruiken.

### 2.5.3 Conclusie

Bij deel 1 kwam ik voor accuracy rond de 80% maar dat was bij deel 2 niet zo, na heel veel de lagen aangepast te hebben was het beste dat ik kon krijgen was een accuracy van 37,7%. In beide delen wordt een CNN opgebouwd omdat een CNN het beste is om afbeeldingen te herkennen. Dit labo was best leuk maar ook niet nieuw.

## 2.6 Labo 6 (Project)

### 2.6.1 Stappen

Stap 1: Model maken a.d.h.v. data uit testdataset en persoonlijke dataset maken

Stap 2: Model omzetten naar zelfgemaakte dataset

Stap 3: Model finetunen voor hogere accuracy

Stap 4: Microfoonscript maken

Stap 5: Testen van model met microfoonoutput

Stap 6: Coral DEV board instellen

Stap 7: Model omzetten naar tflite

Stap 8: Code op Coral DEV board plaatsen

Stap 9: Testen van code

### 2.6.2 Details

Architectuur

- Het model dat gebruikt wordt is CNN, omdat een WAV-bestand wordt omgezet naar een afbeelding van de samples. Een CNN is het beste voor afbeeldingen te herkennen, daarom wordt dit hiervoor gebruikt.
- Dit model wordt na het maken opgeslagen met tflite.

Training

1. Een model maken met een al bestaande dataset van een paar Engelse woorden.
2. Vervolgens is het model aangepast geweest om accurater te zijn.
3. Daarna is ons eigen dataset geïmplementeerd en het model aangepast.

Implementatie

- Voor de code op het Coral DEV Board te plaatsen is gebruik gemaakt van een GitHub repository.
- Het *"mic.py"* script bevat de opname van de microfoon, de output ervan wordt opgeslagen in een WAV-bestand.
- De lengte van het WAV-bestand is één seconde lang wat voor ons 44100 samples bevat. Als de input ook één seconde lang wordt, of m.a.w. 44100 samples, kunnen er makkelijk voorspellingen gemaakt worden.
- De LED's worden aangesproken *"periphery"* library van Coral in het bestand *"gpio.py".*

Testresultaten

- Het model is uiteindelijk op 100% accuracy gekomen.
- Een probleem dat we ondervonden is de input WAV voor de voorspelling, deze is op dit moment nog geen seconde lang en bevat alleen het gesproken woord (wat korter dan een seconde is).
- Ook al is de input WAV niet volledig correct, kunnen er nog steeds goede voorspellingen gemaakt worden.
- Al de rest werkt zoals behoren, als in, alle LED's kunnen geschakeld worden met alle woorden die gebruikt zijn bij het trainen.

### 2.6.3 Code