|  |
| --- |
| **Portfolio**  Edge AI |
| **Opleiding: Elektronica-ICT, Internet of Things**  **Academiejaar: 2023-2024**  **Hulsbergen Sten**  **Haddouchi Hassan, Vanhulle Geert** |

AP.BE

**Inhoud**

[1 Theorie 2](#_Toc154574685)

[1.1 AI/ML cursussen 2](#_Toc154574686)

[1.2 Edge AI cursussen 2](#_Toc154574687)

[2 Labo’s 3](#_Toc154574688)

[2.1 Labo 1 3](#_Toc154574689)

[2.1.1 Conclusie 3](#_Toc154574690)

[2.2 Labo 2 3](#_Toc154574691)

[2.2.1 Deel 1 3](#_Toc154574692)

[2.2.2 Deel 2 3](#_Toc154574693)

[2.2.3 Conclusie 3](#_Toc154574694)

[2.3 Labo 3 4](#_Toc154574695)

[2.3.1 Deel 1 4](#_Toc154574696)

[2.3.2 Deel 2 4](#_Toc154574697)

[2.3.3 Deel 3 4](#_Toc154574698)

[2.3.4 Conclusie 4](#_Toc154574699)

[2.4 Labo 4 4](#_Toc154574700)

[2.4.1 Deel 1 4](#_Toc154574701)

[2.4.2 Deel 2 4](#_Toc154574702)

[2.4.3 Conclusie 4](#_Toc154574703)

[2.5 Labo 5 5](#_Toc154574704)

[2.5.1 Deel 1 5](#_Toc154574705)

[2.5.2 Deel 2 5](#_Toc154574706)

[2.5.3 Conclusie 5](#_Toc154574707)

[2.6 Labo 6 (Project) 5](#_Toc154574708)

[2.6.1 Stappen 5](#_Toc154574709)

[2.6.2 Details 6](#_Toc154574710)

[2.6.3 Code 6](#_Toc154574711)

# 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.

## 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.

## 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.

# 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.

## Labo 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.

## Labo 2

### Deel 1

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

### Deel 2

Stap 1: Data verzamelen en importeren

Stap 2: Data verkennen

Stap 3: Data Preprocessing

### 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.

## Labo 3

### 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.

### Deel 2

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

### Deel 3

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

### 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.

## Labo 4

### 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 word een model gemaakt met een aantal zelf in te stellen lagen. Daarna wordt het model gecompiled, getrained, geevalueerd en getest.

### 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.

### 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.

## Labo 5

### Deel 1

In dit deel word 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.

### Deel 2

In dit deel word 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.

### 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 word een CNN opgebouwd omdat een CNN het beste is om afbeeldingen te herkennen. Dit labo was best leuk maar ook niet nieuw.

## Labo 6 (Project)

### 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

### Details

Architectuur

* CNN

Training

Implementatie

Testresultaten

### Code

*---------------Model---------------*

input\_shape = example\_spectrograms.shape[1:]

print('Input shape:', input\_shape)

num\_labels = len(label\_names)

norm\_layer = layers.Normalization()

norm\_layer.adapt(data=train\_spectrogram\_ds.map(

    map\_func=lambda spec, label: spec))

model = models.Sequential([

    layers.Input(shape=input\_shape),

    layers.Resizing(32, 32),

    norm\_layer,

    layers.Conv2D(32, 3, activation='relu'),

    layers.Conv2D(64, 3, activation='relu'),

    layers.Conv2D(128, 3, activation='relu'),

    layers.MaxPooling2D(),

    layers.Dropout(0.25),

    layers.Flatten(),

    layers.Dense(32, activation='relu'),

    layers.Dropout(0.5),

    layers.Dense(num\_labels),

])

model.summary()

model.compile(

    optimizer=tf.keras.optimizers.Adam(),

    loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

    metrics=['accuracy'],

)

history = model.fit(

    train\_spectrogram\_ds,

    validation\_data=val\_spectrogram\_ds,

    epochs=EPOCHS,

    callbacks=tf.keras.callbacks.EarlyStopping(verbose=1, patience=2),

)