

Edge AI Portfolio

Yves Braeckman

26 december 2025

Inhoudsopgave

1 Inleiding	3
2 Portfolio: Lessen en Labo's	4
2.1 Labo 1.1	4
2.1.1 Beschrijving en Aanpak	4
2.1.2 Persoonlijke Inzichten en Reflectie	4
2.1.3 Conclusie	4
2.2 Labo 4	5
2.2.1 Beschrijving en Aanpak	5
2.2.2 Persoonlijke Inzichten en Reflectie	5
2.2.3 Conclusie	5
3 Edge AI Project: Rapportage	6
3.1 Projectintroductie	6
3.2 Eigen Bijdrage en Leerproces	7
3.2.1 Wat heb ik bijgedragen?	7
3.2.2 Wat heb ik geleerd?	7
A Appendix: Technical Project Guide	8
A.1 Benodigde Hardware	8
A.2 Software en Dependencies	8
A.3 Installatie-instructies	8
A.4 Opstartprocedure	8
A.5 Testinstructies en Model Performantie	9
A.5.1 Model Analyse	9
A.5.2 Foutafhandeling en Beperkingen	9

1 Inleiding

2 Portfolio: Lessen en Labo's

2.1 Labo 1.1

2.1.1 Beschrijving en Aanpak

In dit eerste labo heb ik een data analyse gedaan op verschillende datasets, om zo een beter begrip van de data te krijgen. Dit zorgt er voor dat de volgende stap, het maken van een ML-model, gebeurt op een correcte manier.

Aanpak: Tijdens het maken van het labo heb ik volgende stappen doorlopen:

1. **Data import:** Ik heb het .csv-bestand gedownload en geïmporteerd in een Pythonomgeving met behulp van de pandas library. Dit creëerde een DataFrame waar ik bewerkingen op kon doen.
2. **Analyse:** Voor de analyse heb ik met behulp van pandas en numpy functies het gemiddelde, de mediaan, de standaardafwijking en de kwartieLEN berekend.
3. **Visualisatie:** De volgende stap was de data visualiseren. Hiervoor heb ik gebruikgemaakt van de matplotlib library. Ik heb histogrammen gegenereerd om de vorm van de distributie te controleren en boxplots gemaakt om eventuele uitschieter te vinden.
4. **Interpretatie:** Ik was bij het maken van het labo niet aan deze stap geraakt en ik heb de correctiesleutel geraadpleegd voor deze stap.

2.1.2 Persoonlijke Inzichten en Reflectie

2.1.3 Conclusie

2.2 Labo 4

2.2.1 Beschrijving en Aanpak

In dit labo was het de bedoeling om zelf een netwerk te bouwen en vervolgens dit netwerk te trainen om geschreven letters van a tot z te herkennen. Als laatste stap moet de performantie van het netwerk getest worden.

Aanpak: Tijdens het maken van het labo heb ik volgende stappen doorlopen:

1. **Data Acquisitie en Preprocessing:** Als eerste stap heb ik scaling toegepast op de dataset zodat alle afbeeldingen van uint8 [0, 255] naar float32 [0.0, 1.0] werden omgezet.
2. **Training en testen:** De tweede stap was het verdelen van de data in een training stuk en een test stuk om op het einde de performantie te kunnen testen.
3. **Model bouwen:** Na een beetje opzoekwerk had ik gelezen dat een Conv2D laag beter werkt voor afbeeldingen dan een Dense laag. Ik heb vervolgens verschillende combinaties van lagen geprobeerd en verschillende groottes van lagen.
4. **Training en Compilatie:** Het model is gecompileerd met de Adam optimiser. Ik heb hier gekozen voor een heel lage learning rate omdat het model anders stopte met leren. Het model is getraind gedurende 15 epochs.

Gebruikte Tools:

- **TensorFlow en Keras:** Voor het bouwen, trainen en evalueren van het neurale netwerk.
- **TensorFlow Datasets:** Voor het inladen van de EMNIST dataset.

2.2.2 Persoonlijke Inzichten en Reflectie

Ik vond dit labo interresant omdat dit alles van de vorige labo's een beetje samenbracht. Ik vond het wel lastig om het model te bouwen omdat we hier niet veel over gezien hebben. Dit was dus een trial-en-error process en liep niet vlot.

Ik heb uiteindelijk online opgezocht en gebruik gemaakt van LLM's om te begrijpen wat alle parameters doen bij het bouwen van een model. Zo heb ik bijgeleerd over de learning rate en het belang van dit getal bij de training. Wanneer ik dit te hoog koos, ging het model in volledige chaos en leerde het niet, maar als ik het laag koos bleef het model steken op een bepaalde accuracy.

2.2.3 Conclusie

3 Edge AI Project: Rapportage

3.1 Projectintroductie

Motivatie

Voor het project van Edge AI, heb ik, samen met mijn groep, gekozen om een afvalsorteersysteem te maken op basis van machine learning. Ik vond dit onderwerp interessant omdat dit een brug bouwt tussen sustainable IT en Edge AI, we gebruiken IT om duurzamer te zijn. Ook vond ik het mooi een dagelijks probleem te kunnen aanpakken, aangezien er vaak fouten tegen gemaakt worden en het veel efficienter zou kunnen wanneer dit automatisch gebeurt.

Doel

Het doel van dit project is het ontwerpen en trainen van een machine learning model dat in staat is om vijf verschillende afvalcategorieën correct te classificeren. Daarnaast is het doel om dit model te implementeren op een Coral Dev Board. Hiermee willen we aantonen dat het systeem in staat is om op basis van visuele input een fysieke actie aan te sturen. Binnen dit project simuleren we de aansturing van een sorteermachine visueel met behulp van vijf LED's, waarbij elke LED correspondeert met een specifieke afvalstroom.

Aanpak

Voor de praktische uitwerking van het project is er gebruik gemaakt van Roboflow voor het uitbreiden en cleanen van de dataset, Python met het TensorFlow library voor het ontwerpen en trainen van het model en een Coral Dev Board met een webcamera.

Opzet van het systeem

De technische structuur van het systeem is opgebouwd als een eindeloze lus in Python die op het Coral Dev Board draait.

1. **Initialisatie:** Bij het opstarten laadt het script het geconverteerde .tflite-model in het geheugen van de Edge TPU en worden de GPIO-pinnen voor de vijf LED's geconfigureerd als output.
2. **Input** De lus begint met het ophalen van een frame van de live webcam-feed. Dit beeld wordt voorbewerkt door het te schalen naar de input-resolutie die het model verwacht.
3. **Inferentie** De Edge TPU voert de berekeningen uit op het beeld. Het resultaat is een lijst met waarschijnlijkheidsscores voor elk van de vijf afvalcategorieën.
4. **Output:** De code controleert of de hoogste score boven een vooraf ingestelde drempelwaarde ligt. Als een van de score's boven de drempel ligt, dan wordt via de GPIO-interface de LED aangestuurd die overeenkomt met de herkende categorie. Indien niets met zekerheid wordt herkend, blijven de LED's uit.

Dit proces herhaalt zich heel snel, wat zorgt voor een real-time reactie wanneer een gebruiker afval voor de camera houdt.

3.2 Eigen Bijdrage en Leerproces

3.2.1 Wat heb ik bijgedragen?

3.2.2 Wat heb ik geleerd?

A Appendix: Technical Project Guide

A.1 Benodigde Hardware

Hardware Component	Specificatie / Opmerking
Google Coral Dev Board	Om code op te draaien
USB-C Kabel	Voor stroomtoevoer
Ethernet Kabel	Voor SSH
USB Webcam	Input voor beeldherkenning
5x LED	Rood, Groen, Geel, Blauw, Wit
5x Weerstand	330Ω
Breadboard	Voor het prototype circuit
Jumper Wires	Male-to-Male / Male-to-Female
PC / Laptop	Voor SSH-toegang en opstarten

Tabel 1: Lijst van benodigde hardware

A.2 Software en Dependencies

A.3 Installatie-instructies

A.4 Opstartprocedure

1. Hardware Opstelling

Voordat het bord wordt ingeschakeld, moeten alle componenten correct verbonden zijn:

1. **Camera:** Sluit de USB-webcam aan op de USB-A poort van het Coral Dev Board. Richt de camera op een witte achtergrond met voldoende belichting.
2. **Circuit:** Sluit de LED's en weerstanden aan op het breadboard en verbind deze met de GPIO-pinnen volgens het bedradingsschema.
3. **Netwerk:** Verbind de ethernetkabel met het bord en laptop voor SSH.
4. **Stroom:** Sluit als laatste de USB-C voedingskabel aan op de poort met het label PWR.

2. Verbinding en Uitvoering

Stap A: Verbind via SSH

Open een terminal en maak verbinding met het bord:

```
ssh mendel@192.168.1.100
```

Stap B: Navigeer naar de projectmap

Ga naar de folder waar de broncode en het model staan:

```
cd /home/mendel/Edge-AI-Project
```

Stap C: Start de applicatie

Voer het Python-script uit om de detectie te starten.

```
python3 main.py
```

Stap D: De applicatie gebruiken

Ga op de laptop/PC naar de browser en surf naar 192.168.1.100:5000

3. Stoppen van het systeem

Om het programma te stoppen, druk je in de terminal op:

CTRL + C

A.5 Testinstructies en Model Performantie

A.5.1 Model Analyse

A.5.2 Foutafhandeling en Beperkingen