

Projectbeschrijving Project 3

“De maakindustrie”

Inhoudsopgave

Inleiding.....	3
Opdracht.....	3
Doel.....	3
Onderdelen.....	4
<i>Afwijkingherkenner.....</i>	<i>4</i>
<i>Acceptance Quality Limit.....</i>	<i>4</i>
<i>Statistieken.....</i>	<i>5</i>
<i>Chat functionaliteit.....</i>	<i>5</i>
Deliverables.....	6
Projectevaluatie.....	7
IT voor AI.....	7
Toegepaste wiskunde.....	7
Professionele vaardigheden.....	7
<i>Toetsing.....</i>	<i>8</i>
Tips & Tricks.....	9
Daily Kata Starter Questions.....	9
Bronnen.....	9

Inleiding

De tuinbouwsector in Zuid-Holland loopt wereldwijd voorop. Nieuwe technologieën kunnen helpen om die positie te behouden en verder te versterken. Artificial Intelligence kan hier een belangrijke rol in spelen. Verschillende AI technologieën kunnen worden ingezet om processen te versnellen en te automatiseren. Omdat de regio Zuid-Holland inzet op samenwerking en innovatie in de tuinbouw en hier veel kansen liggen voor Artificial Intelligence speelt het tweede project zich af binnen deze context.

Opdracht

Een appelverwerkingsbedrijf Pink Lady wil een aantal processen gaan automatiseren. In dit bedrijf worden grote batches met appels verwerkt om bijvoorbeeld appelmoes of appelstroop van te maken. De appels in een batch zijn echter niet altijd allemaal gezond. Afhankelijk van de hoeveel gezonde appels wordt bepaald wat er met de appels moet gebeuren. Op dit moment zijn er mensen in dienst die handmatig bekijken of appels ziek of gezond zijn, en als een appel ziek is wat voor ziekte de appel heeft. Om dit proces te automatiseren wil Pink Lady met behulp van Computer Vision bepalen wat de kwaliteit van een batch appels is om op basis van deze informatie automatisch een keuze te maken wat er met de batch gaat gebeuren. Daarnaast wil Pink Lady statistische informatie opvragen over de appels met een chatbot.

Doel

Dit project bestaat uit meerdere onderdelen:

1. Afwijkingen in appels herkennen
2. Keuring uitvoeren op een of meerdere appel *batches*
3. Statistieken over de *batches* weergeven
4. Statistieken uitvragen via chat

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

Onderdelen

Afwijkingherkenner

Voor het herkennen van de afwijkingen verwachten we dat jullie zelf een Convolutional Neural Net (CNN) ontwerpen, trainen, valideren en testen: de *bootstrap* methode. Een in de praktijk veel voorkomende handicap die ook bij deze opdracht een rol speelt, is de beschikbaarheid van slechts een beperkte hoeveelheid correct gelabelde trainingsdata. Om tot redelijke resultaten op de testset te komen, is het nodig de trainingset kunstmatig te vergroten, door de beschikbare beelden bijvoorbeeld te roteren, spiegelen, verschuiven en “zoomen” of door bijvoorbeeld ruis toe te voegen of de kleuren te veranderen. De te gebruiken neural net library PyTorch heeft hiervoor een aantal standaardfuncties. Als dit nodig blijkt mag je de dataset ook vergroten door afbeeldingen toe te voegen maar let dan wel op de bronvermelding (zeer actueel deze tijden).

Daarnaast vergelijken jullie de uitkomsten van het zelfgetrainde model met een tweede leermethode: *transfer learning*. Op basis van de inzichten van hierboven kiezen jullie een geschikt voorgetraind CNN model en hertrainen dit voor deze toepassing. Reflecteer op de verschillen en kies het meest accurate model voor deze toepassing.

Acceptance Quality Limit

Er zijn meerdere methoden voor het beoordelen van een appel batch. Voor dit project vragen we jullie om een implementatie te maken van het zogenaamde Acceptance Quality Limit (AQL) protocol. Dit protocol is een industriestandaard voor statistische kwaliteitscontroles. Informatie hierover vinden jullie in de projectbijlages en de tabel AQLChart.pdf.

We hanteren de volgende kwaliteitslabels:

Klasse 1: kwaliteit appels is goed genoeg om in supermarkt of bij groeteboer te liggen

Klasse 2: kwaliteit appels is goed genoeg voor verwerking in appelmoes

Klasse 3: kwaliteit appels is goed genoeg voor verwerking in stroop

Afgekeurd: kwaliteit is onvoldoende voor bovenstaande toepassingen

In termen van het AQL inspectieschema:

General Inspection Level: I

Lot size: een oogst batch bevat tussen 250 en 500 appels*

*Een lot size van 500 appels is natuurlijk onrealistisch laag, maar beter voor de opdracht hieronder.

Klasse 1: AQL kleiner of gelijk aan 0.4

Klasse 2: AQL tussen 0.4 en 6.5

Klasse 3: AQL tussen 6.5 en 15.0

Afgekeurd: AQL groter dan 15.0

Stel op basis van de testset een batch samen, waarop je een inspectie kunt uitvoeren. Maak bijvoorbeeld een batch met een *sample size* van 20 appels, waarvan 14 gezond, 2 met *blotch*, 1 met *rot* en 3 met *scab*, en voer dit aan je afwijkingherkenner. Hieruit moet dan het juiste kwaliteitslabel rollen. Uiteraard kun je hiervoor een compact stukje logica schrijven, maar het samenstellen mag ook met de hand. De moeilijkheid is hier om de onnauwkeurigheid van je model te relateren aan de uitkomst van het label. Kun je aangeven hoe betrouwbaar de uitkomst van het label is?

Statistieken

Naast een kwaliteitslabel (en de betrouwbaarheid van het label) toont de applicatie ook het aantal gezonde en zieke appels, de laatste uitgesplitst in de categorieën gevlekt (*blotch*), rot (*rot*) en schurft (*scab*). Dit onderdeel is verder uit te breiden in overleg.

Chat functionaliteit

De bovenstaande statistieken willen we ook uitvragen via een chatvenster. Je gebruikt hiervoor een bestaand SBERT-model of een variant hierop. Het gaat hierbij om de herkenning van de semantische overeenkomst tussen zinnen die de gebruiker intypt en standaardzinnen waarop een standaardantwoord voorhanden is. Zo'n standaardantwoord kan variabele parameters bevatten zoals aantallen. Een voorbeeld:

Vraag: "Hoeveel procent van de appels is gezond?"

Antwoord: "{X}" (bijv. 14) van de {Y} (bijv. 20) appels zijn gezond dus dat is {P}% (bijv. 70%)

Het voert te ver om de output te koppelen aan een tekstgenerator (een *transformer* als GPT), al is dit natuurlijk wel mogelijk. Text2Speech behoort tot de mogelijkheden, maar dit is een *nice to have*.

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 dinsdag 27 juni, 12:00 uur** naar:

Teams > Applied Artificial Intelligence – Make IT Work >
Jouw Private Channel > Files > Deliverables periode 3 > Project 3

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) Realisatie en testen van een effectieve netwerk-topologie
- 2) Het opsplitsen van een applicatie in "loosely coupled" onderdelen en het afzonderlijk "unit"-testen van deze onderdelen
- 3) Gebruik van PyTorch en zelfstandig bestuderen van de documentatie

Toegepaste wiskunde

- 1) Bayes classificatie en Monte Carlo methoden, experimentele, numerieke benadering (simulatie)
- 2) Steekproeven en betrouwbaarheid, intuïtieve benadering
- 3) Gebruik van vectoren als representatie van woordsemantiek

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 appelafwijkingherkenner kunnen verklaren

Tips & Tricks

Structuren – Deel je code op in losse, apart testbare classes en methods voor resp. beeldverwerking, AQL inspectie, grafische en tekstuele presentatie.

Valideer de broncode – Voer de tests uit zoals beschreven in de testspecificaties en verwerk de resultaten in een puntsgewijs, kwantitatief georiënteerd testrapport.

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 / aanpassingen 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> voor achtergrondinformatie.

Bronnen

[What is AQL?](#)

[AQL inspection levels](#)

[Convolutional Neural Networks Explained - Computerphile](#)

[Tricking AI Image Recognition - Computerphile](#)

[Inside a Neural Network - Computerphile](#)

[JPEG files & Colour - Computerphile](#)