Projectbeschrijving Project 2

**“De maakindustrie”**

Inhoudsopgave

[Inleiding 3](#_Toc117063273)

[Opdracht 3](#_Toc117063274)

[Doel 3](#_Toc117063275)

[Requirements 4](#_Toc117063276)

[Afwijkingherkenner 4](#_Toc117063277)

[Acceptance Quality Limit 4](#_Toc117063278)

[Statistieken 5](#_Toc117063279)

[Chat functionaliteit 5](#_Toc117063280)

[Deliverables 6](#_Toc117063281)

[Projectevaluatie 7](#_Toc117063282)

[IT voor AI 7](#_Toc117063283)

[Toegepaste wiskunde 7](#_Toc117063284)

[Professionele vaardigheden 7](#_Toc117063285)

[Toetsing 8](#_Toc117063286)

[Tips & Tricks 9](#_Toc117063287)

[Daily Kata Starter Questions 9](#_Toc117063288)

[Bronnen 9](#_Toc117063289)

# 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*.

# Requirements

## 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 TensorFlow/Keras heeft hiervoor een aantal standaardfuncties. Om deze te kunnen gebruiken dienen documentatie en voorbeelden te worden bestudeerd, eveneens een veel voorkomende situatie in de praktijk.

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 bestaand 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 ([aql\_poster.pdf](https://github.com/MitwRdam/MakeAIWork/blob/main/projects/p2/cursus_materiaal/achtergrond_informatie/aql_poster.pdf)).

We hanteren de volgende kwaliteitslabels:

**Klasse 1**: kwaliteit appels is goed geoeg 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**: 1

**Lot size**: een oogst batch bevat tussen de 5.000 en 10.000 appels

**Klasse 1**: AQL kleiner of gelijk aan 0.4

**Klasse 2**: AQL tussen 0.4 en 2.5

**Klasse 3**: AQL tussen 2.5 en 0.65

**Afgekeurd**: AQL groter dan 0.65

Schrijf ook een compact programma dat op basis van de validatie- en test-set van de appels een batch samenstelt, waarop je een inspectie kunt uitvoeren. Genereer bijvoorbeeld een batch met een *sample size* van 80 appels, waarvan 72 gezond, 4 met *blotch*, 1 met *rot* en 3 met *scab*, en voer dit aan je afwijkingherkenner. Hieruit moet dan het juiste kwaliteitslabel rollen.

## Statistieken

Naast een kwaliteitslabel 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 hertraint hiertoe een bestaand SBERT model voor deze specifieke toepassing (wederom via transfer learning). Het gebruik ervan in dit project focust zich op herkenning van de semantische overeenkomst tussen zinnen die de gebruiker intypt en standaard-zinnen waarop een standaard-antwoord voorhanden is. Zo’n standaard antwoord kan wel variabele parameters bevatten zoals aantallen. Een voorbeeld hiervan is het antwoord op de vraag: “Hoeveel procent van de appels is gezond”.

Het voert te ver om de output te koppelen aan een tekstgenerator (een *transformer* model als GPT-2), 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>

# 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 aan de hand van een gegeven, niet geheel eenduidig en nodeloos complex wetenschappelijk artikel aangevuld met zelfstandig te zoeken en gebruiken bronnen
2. Het opsplitsen van een applicatie in “loosely coupled” onderdelen en het afzonderlijk “unit”-testen van deze onderdelen
3. Gebruik van MatPlotLib en zelfstandig bestuderen van de documentatie, aangevuld met gerichte vragen

### Toegepaste wiskunde

1. Bayes classificatie en Monte Carlo methoden, experimentele, numerieke benadering (simulatie), sporadisch aangevuld met formele notatie
2. Steekproeven en betrouwbaarheid, intuitieve 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 programma’s of modules 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 georienteerd 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 / aanpassingenen 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](https://traccsolution.com/blog/toyota-kata/) voor achtergrondinformatie.

# Bronnen

[Projectbenodigdheden](https://github.com/AlxcNL/MakeAIWork2/tree/main/projects/apple_disease_classification) (bijlages, data)

[Convolutional Neural Networks Explained - Computerphile](https://mediasite.hro.nl/Mediasite/MyMediasite/presentations/5fc68f8dde054c279af95f5e61e0884b1d)

[Tricking AI Image Recognition - Computerphile](https://mediasite.hro.nl/Mediasite/MyMediasite/presentations/c52acf9def7b4ffab2795f5f1c6f47c21d)

[Inside a Neural Network - Computerphile](https://mediasite.hro.nl/mediasite/Play/a5fee00899204852947a0c5aa861156c1d)

[JPEG files & Colour - Computerphile](https://mediasite.hro.nl/mediasite/Play/6c1caa39363c4c06991986058fc53e531d)