

Het suggereren van sportschema's door middel van objecttherkenning en generatieve AI.

Maurice Cantaert.

Scriptie voorgedragen tot het bekomen van de graad van
Professionele bachelor in de toegepaste informatica

Promotor: Mevr. M. Van Audenrode

Co-promotor: Dhr. J. Van Bossuyt

Academiejaar: 2023–2024

Tweede examenperiode

Departement IT en Digitale Innovatie .

**HO
GENT**

Woord vooraf

Als ik terugblik op de conclusie van mijn bachelorproef en hoe ik die heb bereikt, kan ik zeggen dat ik tevreden ben met het eindresultaat. Het waren enkele vermoeiende maanden, vooral in combinatie met mijn stage bij Inetum-Realdolmen. De periode voorafgaand waarin ik op zoek was naar een onderwerp en een co-promotor was wat lastig, waardoor ik met minder motivatie kon beginnen. Maar goed begonnen is half gewonnen, en dat merkte ik zeker in dit geval. Hierdoor kon ik deze proef met een gerust hart uitwerken en voordragen.

Artificiële intelligentie evolueert steeds sneller en het gebruik daarvan neemt toe, ook diensten zoals ChatGPT zijn niet meer weg te denken. Hiermee kreeg ik de interesse om te onderzoeken in welke gebieden dit een meerwaarde zou kunnen bieden, en met dank van Kilian Ostijn kwam ik uit op de sportsector. Ik wilde ook meer leren over deep learning en hoe deze gebruikt wordt om recente innovaties binnen de (toegepaste) informatica te realiseren.

Het onderwerp is doorheen de uitwerking van de proef geëvolueerd van een algemene proof-of-concept gebruikmakend van AI naar een specifieke oplossing op maat voor personal trainers, waaronder mijn co-promotor. Hiervoor wil ik graag mijn promotor bedanken. Enerzijds voor de hands-off approach, waarmee ik meer creatieve vrijheid had, en anderzijds voor het leren om de proef niet alleen vanuit een onderzoekend en technisch perspectief te bekijken, maar ook vanuit een meer praktisch haalbaar perspectief met oog op business value.

Ook wil ik graag mijn co-promotor, Jordi Van Bossuyt, bedanken voor alle hulp en advies doorheen de uitwerking van deze thesis. Zonder zijn hulp zou het veel moeilijker geweest zijn om een resultaat te bereiken dat een echte meerwaarde biedt aan mijn doelgroep, de personal trainingsector. Daarnaast wil ik hem en mijn vrienden bedanken voor de foto's die ik kon gebruiken bij het testen van de applicatie. Tot slot wil ik mijn ouders en broers bedanken voor de motivatie en steun, zonder hen zou ik deze thesis nooit op tijd afgewerkt gekregen hebben.

Samenvatting

Grote taalmodellen binnen de kunstmatige intelligentie, en in het verlengde hiervan multimodale taalmodellen, brengen dag-na-dag steeds meer mogelijkheden ter tafel. Technologische giganten zoals Google en Apple zien het potentieel ervan en bieden daarmee steeds meer functionaliteiten die versterkt zijn door deze taalmodellen. Deze bachelorproef onderzoekt of deze recente innovaties een oplossing kan bieden in de sportsector, en met name de personal training.

Sportparticipatie binnen Vlaanderen scoort hoog met een gemiddelde van een op vier Vlamingen die (bijna) dagelijks een sport beoefent, echter kan hetzelfde niet gezegd worden over de verhouding beschikbare trainers. Clubs blijven kampen met een trainerstekort, ook komt het naar voren dat gemiddeld 26 procent van jonge trainers afhaken door de werkdruk. Hiermee ontstaat de vraag of de recente innovaties binnen artificiële intelligentie een oplossing kan bieden voor dit knelpunt: Is het mogelijk om generatieve AI in te zetten om een deel van de verantwoordelijkheid van personal trainers over te nemen?

Deze bachelorproef richt zich erop te onderzoeken of een nieuw platform opgezet kan worden met behulp van bestaande taalmodellen en datasets. Het platform hoort enerzijds de werkdruk te verminderen en anderzijds een positieve meerwaarde te bieden aan de behoeften van klanten. Om dit te bereiken komt een literatuurstudie aan bod dat de huidige stand van zaken rond objectherkenning en generatieve AI in kaart brengt. Alvorens een proof-of-concept uit te werken zal er zich eerst een korte requirements-analyse plaats vinden. Op basis van de resultaten uit deze analyse kan er een selectie gemaakt worden van de gebruikte technologieën.

De proof-of-concept bewijst de mogelijke voordelen van kunstmatige intelligentie in de sportsector. Vooropgestelde vereisten zoals het genereren van suggesties wordt overtroffen door constructieve en gestructureerde suggesties in Markdown formaat waar de coach verder op kan inspelen. Ondanks de benodigde maatregelen om mogelijke foutieve inschattingen kunnen enkele foutieve metingen plaatsvinden. Toch kan het implementeren van generatieve AI een meerwaarde bieden, zeker met het mogelijke vooruitzicht om biometrische data toe te passen om suggesties gepersonaliseerder te maken.

Inhoudsopgave

Lijst van figuren	viii
Lijst van Codefragmenten	ix
Lijst van Codefragmenten	ix
1 Inleiding	1
1.1 Probleemstelling	1
1.2 Onderzoeksvraag	2
1.3 Onderzoeksdoelstelling	2
1.4 Opzet van deze bachelorproef	2
2 Stand van zaken	4
2.1 Selectie van bronnen	4
2.1.1 De CRAAP-test	4
2.1.2 Gebruikte zoekstrategieën	5
2.1.3 Selectiecriteria	5
2.1.4 Bronnen buiten de literatuurstudie	5
2.2 Klassieke objectherkenning	5
2.2.1 Computer visie	6
2.2.2 Beeldfragmenten in datavorm	7
2.2.3 Het verwerken van data uit multimedia	8
2.2.4 Klassieke objectherkenningsalgoritmen	9
2.2.5 Uitdagingen en beperkingen	10
2.3 Machine learning	11
2.3.1 Hoe machine learning en deep learning werkt	11
2.3.2 Deep learning	11
2.3.3 Moderne objectherkenningsalgoritmen	12
2.4 De werking van grote taalmodellen	13
2.4.1 Vooruitgang van taalmodellen	13
2.4.2 Versterkingsleren via menselijke feedback	14
2.4.3 Generatieve AI en deep learning	14
3 Methodologie	16
3.1 Requirements-analyse	16

3.2 Selectie van tools	16
3.3 Proof-of-concept	16
3.3.1 Visie	17
3.4 Conclusie en vooruitzichten	18
4 Selectie van tools	19
4.1 Android-app met Jetpack Compose	19
4.1.1 Motivering voor Jetpack Compose	19
4.2 Achterliggende service met Quarkus	20
4.2.1 Modern raamwerk	20
4.2.2 Quarkus Dev Services	20
4.2.3 Quarkus AI met Langchain4j	20
4.3 Artificiële intelligentie met Vertex AI	21
4.3.1 Gemini 1.5 Pro	21
4.4 Build tool	21
4.4.1 Motivering	22
4.4.2 Gradle als buildtool	22
4.5 Gekozen fitnessinstrument	23
5 Proof-of-concept	24
5.1 Voorvereisten	24
5.1.1 Quarkus CLI	24
5.1.2 Java Software Development Kit	25
5.1.3 Docker	25
5.2 Omgevingen opzetten en testen	26
5.2.1 Opzetten Google Gemini omgeving	26
5.2.2 Opzetten van de Quarkus omgeving	26
5.2.3 Opzetten Jetpack Compose Android-app omgeving	27
5.2.4 Gemini AI testen	28
5.3 Functionaliteiten uitwerken	31
5.3.1 Domeinmodel opzetten	31
5.3.2 Functionaliteiten van de gebruiker	32
5.3.3 Functionaliteiten van de personal trainer uitwerken	35
5.3.4 Opmerkingen	35
5.4 Reproducieren van de proof-of-concept	36
5.5 Resultaten	36
6 Conclusie	37
6.1 Antwoord op de onderzoeksvragen	37
6.2 Reflectie	38

6.3 Vooruitzichten	38
A Onderzoeksvoorstel	39
A.1 Introductie	40
A.2 Literatuurstudie	40
A.2.1 Artificiële intelligentie	40
A.2.2 De technologische vooruitgang van objectdetectie in de cloud	41
A.2.3 Lokale machine learning	41
A.3 Methodologie	41
A.3.1 Voorafgaande literatuurstudie	42
A.3.2 Requirements-analyse	42
A.3.3 Selectie van tools	42
A.3.4 Proof-of-concept	42
A.4 Verwacht resultaten en conclusie	43
Bibliografie	44

Lijst van figuren

2.1	Visualisatie van een LiDAR-systeem met objectherkenning in auto's (Baddoni, 2021)	7
2.2	Visualisatie van pixel binning (Jin & Hirakawa, 2012)	8
2.3	Visualisatie van het verschil in Gaussianen (Lowe, 2004)	10
2.4	Vergelijking tussen GPT 3.5 en GPT 4 bij het afleggen van verschillende academische en professionele examens. Bijkomende computer visie mogelijkheden helpen GPT-4 om betere resultaten te halen (OpenAI, 2023)	14
4.1	Quarkus AI is een dienst waarmee Quarkus applicaties kunnen communiceren met grote taalmodellen. De document store stelt hierbij de databank voor met de geschiedenis van gebruikers in het opvragen van suggesties rond fitnessstoestelen. (Lowe, 2004)	21
4.2	De drie stappen in het Gradle bouwproces (Gradle, g.d.)	23
5.1	De bovenkant van een 16 pond dumbbell, de foto werd genomen met een Apple iPhone 14.	29
5.2	De zijkant van twee 8 kilogram dumbbells, de foto werd genomen met een OnePlus 8T.	30
5.3	Een zijaanzicht van een 5 kilogram dumbbell, de foto werd genomen met een Samsung S23 Ultra.	31
5.4	Formaat om data op te slaan in de databank. Een cliënt vraagt een verzoek aan om meer informatie te krijgen over een object (equipment_request), op basis hiervan krijgt de cliënt een suggestie terug met optionele invloed van de personal trainer (trainer_input).	32

Lijst van Codefragmenten

5.1	Commando om het Quarkus project met bijhorende extensies en configuratie te genereren	27
5.2	Resulterende JSON-data van de eerste computer visie test.	28
5.3	Resulterende JSON-data van de tweede computer visie test.	29
5.4	Resulterende JSON-data van de laatste computer visie test.	30
5.5	Instructies van de bron om een afbeelding te analyseren en daarvoor suggesties te genereren	33
5.6	Instructies om computer visie uit te voeren op een afbeelding om het object te analyseren en vervolgens uit te gieten in een vooraf gedefinieerd JSON-bestand	34
5.7	Omgevingswaarden met informatie over Gemini	36

Inleiding

Sport kent een prominente rol in de levensstijl van Vlamingen, zo blijkt sterk in de statistieken van Statistiek Vlaanderen (2024). Hoewel sportparticipatie van Vlamingen hoog scoort met een gemiddelde van een op vier Vlamingen die (bijna) dagelijks een sport beoefent, kan hetzelfde niet gezegd worden over verhouding beschikbare trainers. Ondanks de stijging van het aantal gediplomeerde trainers blijkt in het jaarverslag van Sport Vlaanderen (2023) blijkt dat sportclubs blijven kampen met een trainerstekort. Sterker nog, het komt naar voren dat gemiddeld 26 procent van de trainers jonger dan 30 jaar afhaken, waardoor de verhouding van één trainer op twintig sporters geen positieve evolutie kent.

Het is duidelijk dat er niet alleen werk gemaakt moet worden om meer coaches op te leiden, maar ook om tewerkgestelde trainers bij te staan in hun takenpakket. Met de voortdurende verbeteringen binnen de artificiële intelligentiesector en de toename in capaciteiten van generatieve AI ontstaan mogelijkheden om de werkdruk van trainers te verlichten. Deze paper richt zich erop een proof-of-concept Android-app te ontwikkelen om sporters op weg te helpen met behulp van generatieve AI en objectherkenning. Trainers kunnen vervolgens bijsturen waar nodig, waarmee de werkdruk om startende sporters bij te staan verkleind kan worden.

1.1. Probleemstelling

Jordi Van Bossuyt is een jonge zelfstandige personal trainer en geeft momenteel training aan cadetten in de Koninklijke Atletiekclub Eendracht Aalst. Met een interesse voor een meer gestroomlijnde manier om startende sporters bij te staan komt het idee van een combinatie van een mobiele app met generatieve AI aan bod. Concreet is het doel om een app te ontwikkelen waarbij er minimale frictie moet zijn om aan de slag te gaan met sportmateriaal. Gebruikers horen de mogelijkheid te krijgen om sportgerelateerde objecten en omgevingen zoals fitnessinstrumenten

of looppistes in te scannen met de camera van de telefoon en hiervoor suggesties te krijgen. Met een proof-of-concept moet aangetoond kunnen worden dat dit mogelijk is door middel van een vooraf-getrainde dataset en bestaande generatieve AI-modellen om zelf suggestieve trainingsschema's te genereren.

1.2. Onderzoeksvraag

Hoe kan een bestaande kunstmatige intelligentieplatform gebruikt worden om aan nauwkeurige objectherkenning te doen? Kan het platform vervolgens suggesties van activiteiten voorstellen aan de gebruiker? Op welke manier kan de trainer inspelen op het de suggesties die gegenereerd worden door de app?

1.3. Onderzoeksdoelstelling

Deze bachelorproef zal zich eerst richten op het onderzoeken van de manier waarop aan objectherkenning en het suggereren van oefeningen kan gedaan worden in de vorm van een literatuurstudie. Hierin komt tevens ook het trainen van datasets in context van machine en deep learning aan bod om een beter begrip te krijgen op de manier waarop kunstmatige intelligentiemodellen getraind en gebruikt worden. Met dit basis begrip zal een proof-of-concept uitgewerkt worden om de haalbaarheid van een mobiele app ter ondersteuning van trainers te illustreren.

De proof-of-concept moet hierbij voldoen aan enkele criteria van de opdrachtgever:

- De app hoort op een kostenefficiënte manier ontwikkeld te worden, waarbij er geen vereiste is om zelf datasets te trainen.
- De gebruikte technologieën horen zo dicht mogelijk bij elkaar aan te sluiten en met elkaar te integreren, er is dus één platform voor zowel de personal trainer als de cliënt.
- Suggesties moeten gegenereerd kunnen worden zonder de input van trainers, maar met de mogelijkheid voor trainers om deze suggesties te kunnen bijsturen.
- Gebruikers horen zo weinig mogelijk frictie te ervaren bij het inzenden van foto's en het ontvangen van suggesties.

1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen. Ook de visie van het technisch luik komt hier aan bod.

In Hoofdstuk 4 worden de gekozen technologieën en platformen toegelicht die gebruikt zullen worden bij het ontwikkelen van de proof-of-concept.

In Hoofdstuk 5 wordt de uitwerking van de proof-of-concept uitgelegd samen met een demonstratie van hoe de app gebruikt kan worden.

In Hoofdstuk 6, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2

Stand van zaken

2.1. Selectie van bronnen

Om deze literatuurstudie en deze bachelorproef aan sich als volwaardig en kwalitatief te kunnen aanschouwen volgt eerst een kleine uitleg rond de keuze voor ondersteunende bronnen. De selectiecriteria bouwt hierbij verder op de CRAAP-test, opgesteld door Blakeslee (2004).

2.1.1. De CRAAP-test

De CRAAP-test bestaat uit volgende criteria, zoals gezien in de syllabus IT-component van het opleidingsonderdeel Research Methods (Bert e.a., 2023):

- **Currency** of actualiteit: de aangehaalde bron is recent genoeg voor het onderwerp.
- **Relevance** of relevantie: het doelpubliek van de bron past bij het doelpubliek van deze paper, bovendien past de geciteerde inhoud bij de literatuurstudie.
- **Authority** of autoriteit: er is geen mogelijke belangenconflict van de auteur van de bron. De auteur heeft eveneens een gepaste achtergrond als vakexpert of is verbonden met een betrouwbare instantie.
- **Accuracy** of correctheid: de bron werd onderwerpen aan een peer review-proces, heeft zelf verifieerbare bronnen of resultaten en heeft genoeg diepgang.
- **Purpose** of doel: het doel van de auteur is om een volwaardige tekst te schrijven zonder vertekend beeld. Hierbij is duidelijk of de inhoud gaat over een mening of over feiten, tevens sluit het doel van de bron aan bij het doel van deze paper.

2.1.2. Gebruikte zoekstrategieën

Zoekmachines en publicatieplatformen zoals Google Scholar en ResearchGate maken het mogelijk om allerlei publicaties te vinden rond het gepaste onderwerp. Artificial intelligence kent een voortdurend proces van verbeteringen vergelijkbaar met het agile werken in de rest van de softwareontwikkelingsindustrie. Hierdoor is het belangrijk dat bronnen geëvalueerd worden volgens hun actualiteit. De focus ligt hierdoor op bronnen die de afgelopen 3 jaar gepubliceerd werden, tevens wordt gekeken of de inhoud nog actueel is aan recentere ontwikkelingen. Een uitzondering wordt gemaakt voor onderzoeken over lang bestaande onderwerpen waarvan de inhoud niet vaak zal veranderen, in het bijzonder klassieke objectherkenningsalgoritmen. Bronnen dienen eveneens relevant te zijn aan deze paper, daarvoor wordt gekozen voor bronnen waarvan het onderwerp overeenstemt met artificiële intelligentie.

2.1.3. Selectiecriteria

Na het filteren van irrelevante en verouderde bronnen komt het beoordelen van de autoriteit en het doel van de bron aan bod. Waar mogelijk is de bron geschreven met als doel om aan een conferentie of universitaire colleges voor te dragen. Eveneens zijn bronnen die gepubliceerd werden in een betrouwbare wetenschappelijk tijdschrift aanvaardbaar. Een wetenschappelijke tijdschrift kan als betrouwbaar aanschouwd worden als publicaties door vakgenoten beoordeeld worden. Tevens worden ook bronnen die reproduceerbaar zijn als geldige, betrouwbare aanschouwd.

2.1.4. Bronnen buiten de literatuurstudie

Bij het opstellen van de shortlist met gebruikte technologieën en het uitwerken van de proof-of-concept zal er voornamelijk naar technische bronnen verwezen worden, dit in tegenstelling tot de academische bronnen die in deze literatuurstudie aangehaald worden. De bovenstaande selectiecriteria zijn hierbij ook van toepassing, met de uitzondering dat er gekeken zal worden naar documentatie van vertrouwde omgevingen in plaats van wetenschappelijke tijdschriften. Een omgeving is vertrouwd als het gaat om een bewezen ecosysteem met actieve ontwikkelaars en stabiel genoeg is om in productie genomen te worden.

2.2. Klassieke objectherkenning

Omdat objectdetectie de kern van de proof-of-concept zal vormen, is het van essentieel belang om eerst een grondig begrip te definiëren van wat deze techniek precies inhoudt. In dit hoofdstuk zal eerst computer visie aan bod komen, een van de gebieden van hedendaagse artificiële intelligentie, waarbinnen objectherkenning zich plaats vindt. Dit zal verduidelijkt worden aan de hand van enkele prak-

tische toepassingen die vandaag al in gebruik genomen worden met behulp van computer visie en objectherkenning. Daarna volgt een uitleg van de processen waarop beelden opgeslagen en verwerkt worden ter voorbereiding op het toepassen van objectherkenning. Vervolgens worden enkele traditionele objectdetectie algoritmen besproken die de verwerkte data kunnen interpreteren. Tot slot komen enkele uitdagingen en limitaties van klassieke algoritmen aan bod, waar het volgende hoofdstuk een mogelijk antwoord op kan bieden.

2.2.1. Computer visie

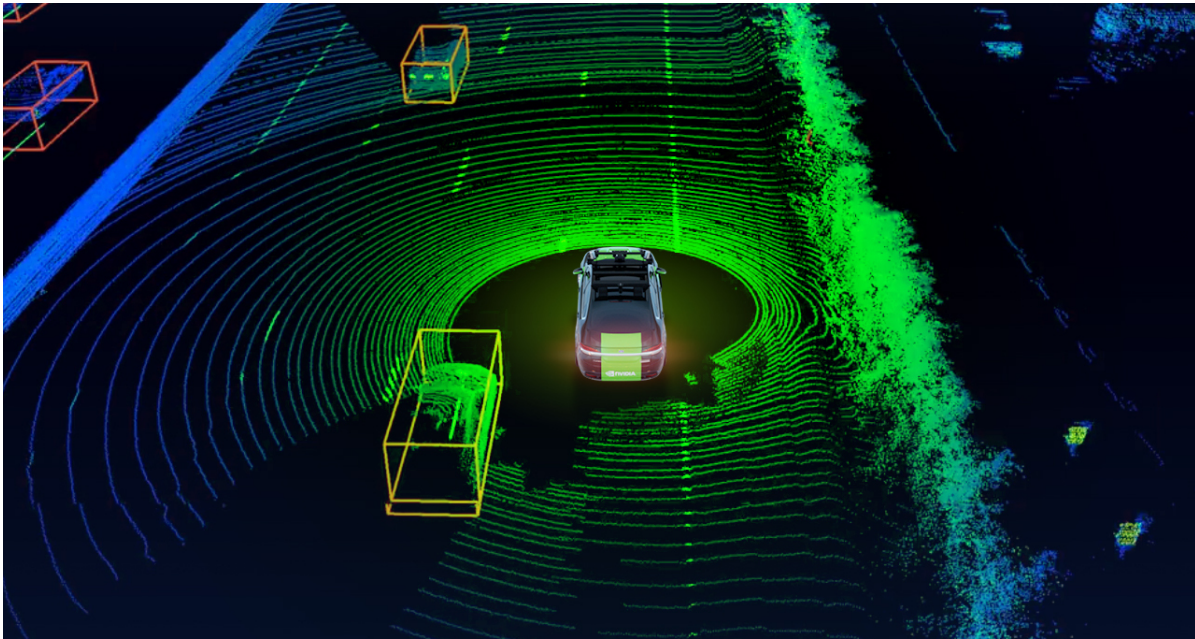
Computer visie is een van de vele vakgebieden binnen de kunstmatige intelligentie en richt zich op het interpreteren van multimedia (Moin, 2023). Met de komst van dit vakgebied ontstaat een grote waaier aan mogelijkheden voor computers om bij te staan bij complexere taken zoals gezichts- en emotieherkenning, scèneanalyse en objectherkenning. Hiermee beschrijven Tasnim en Qi (2023) dat objectdetectie een fundamentele taak omvat binnen dit vakgebied waarmee het identificeren en lokaliseren van objecten mogelijk wordt op (bewegende) beelden. De ontwikkelingen in dit vakgebied worden vandaag steeds meer gebruikt in allerlei sectoren:

Computer visie in de landbouwsector

Radojicic en Cvetković (2023) beschrijven in zijn conferentiepapier voor de 15de Internationale Landbouwsymposium 'AGROSYM 2023' conferentie het volgende over computer visie in de landbouw: "De integratie van robotica, slimme landbouw en computervisietechnologieën in de landbouw heeft een transformerende verschuiving in de industrie teweeggebracht. Het gebruik van computervisiealgoritmen heeft een cruciale rol gespeeld in het realtime monitoren en beheren van gewassen. Door afbeeldingen en video's te analyseren, bieden deze algoritmen boeren tijdige en nauwkeurige informatie over de groei en ontwikkeling van hun gewassen, waardoor ze geïnformeerde beslissingen kunnen nemen over bemesting, irrigatie en pesticidegebruik." Deze resultaten binnen de landbouwsector worden gerealiseerd door middel van objectherkenning binnen computervisie. Om het gebruik hiervan mogelijk te maken worden deep learning algoritmes zoals convolutionele neurale netwerken (CNN) toegepast, meer hiervan komt aan bod in het hoofdstuk [Machine learning](#).

Computer visie in zelfrijdende auto's

Een tweede voorbeeld is het toepassen van computer visie in de transportsector. Met behulp van computer visie-algoritmen zoals kleurenruimte transformatie, Canny-randdetectie en Hough-lijntransformatie wordt het mogelijk voor zelfrijdende auto's om op een goedkopere wijze berekeningen te maken (Gajjar & Sanyal, 2023). Door middel van deze algoritmen verdwijnt de noodzaak om te ondersteunen op duurdere systemen, met name Lichtdetectie- en afstandmetingssensoren (LiDAR) die gebruikt worden om een virtuele driedimensionele map te maken van de direct

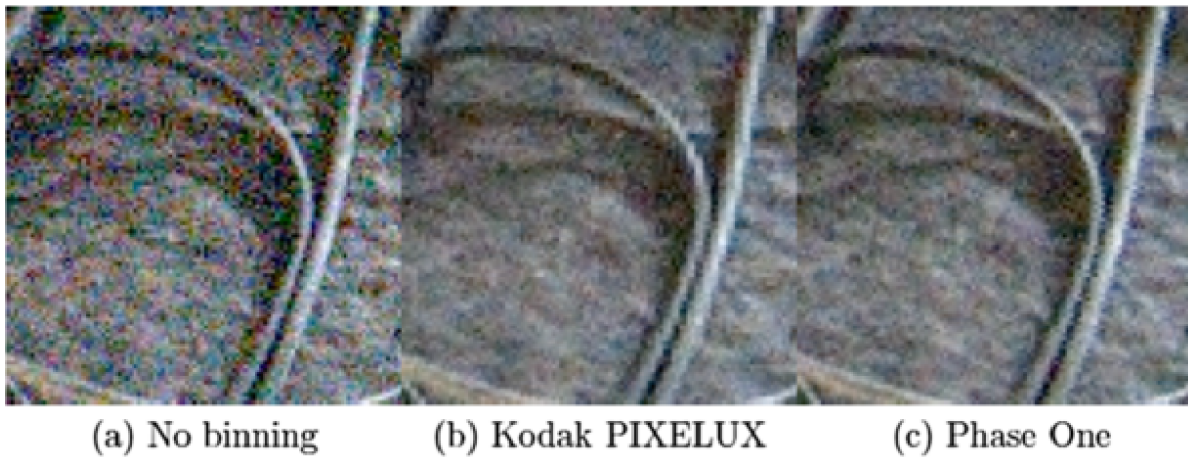


Figuur 2.1: Visualisatie van een LiDAR-systeem met objectherkenning in auto's (Badoni, 2021)

omgeving. Beide systemen kennen een vorm van objectherkenning om objecten zoals verkeersborden of voertuigen in de omgeving te detecteren, een visualisatie van het LiDAR-systeem in zelfrijdende auto's is terug te vinden op afbeelding 2.1. De implementatie hiervan kent echter de moeilijkheid van data-analyse in donkere momenten zoals na zonsondergang, waardoor de kwaliteit van binnenkomende data afzwakt door het lage licht. Een mogelijke oplossing hierbij is het toevoegen van infrarood lampen gecombineerd met een infraroodcamera. Ook hierbij wordt het mogelijk om via CNN-algoritmes te bepalen wanneer overgestapt moet worden op een alternatief systeem.

2.2.2. Beeldfragmenten in datavorm

Concreet worden beelden digitaal opgeslagen in tweedimensionale tabellen van pixels, waarbij elke pixel kleurdata bevat. De resolutie, of scherpheid, van een beeldfragment wordt vaak hierin uitgedrukt. Zo kent een standaard Full HD (FHD)-computerscherm volgens de specificaties van Video Electronics Standards Association (2013) een resolutie van 1920 pixels bij 1080 pixels. Deze resolutie wordt volgens statistieken van Valve Corporation (2024) gebruikt door meer dan de helft van haar gebruikers. Om de kwaliteit van een beeldfragment uit te drukken wordt regelmatig gebruik gemaakt van de meeteenheid megapixels, waarbij één megapixel één miljoen pixels voorstelt. Hiermee ontstaat de vaststelling dat de voorafgaande en meest gebruikte specificatie van Full HD een beeldkwaliteit van 2,1 megapixels kan weer geven aan gebruikers.



Figuur 2.2: Visualisatie van pixel binning (Jin & Hirakawa, 2012)

Kwaliteit van hedendaags camera's

Moderne smartphones zoals de Galaxy S24 nemen volgens Samsung (2024) foto's aan een kwaliteit van 50 megapixels en bieden daarmee ongeveer 24 keer hogere kwaliteit dan een Full HD-computerscherm kan weergeven. Het is hierbij belangrijk om op te merken dat het aantal megapixels niet de enige factor speelt in het bepalen van beeldkwaliteit. Deze meeteenheid bepaalt niet alleen de totale grootte van de data dat een beeldfragment bevat, maar ook ongewenste data zoals ruis of andere artefacten. Om het voorgenoemde fenomeen te mitigeren wordt in vele gevallen aan pixel binning gedaan, een proces waarbij pixels gegroepeerd worden tot zogenaamde superpixels (Jin & Hirakawa, 2012).

2.2.3. Het verwerken van data uit multimedia

Om de nauwkeurigheid en snelheid van objectherkenningsalgoritmen zo veel mogelijk te bevorderen worden eerst enkele stappen uitgevoerd ter voorbereiding. Deze algoritmen kennen immers een langere verwerkingstijd bij grotere bestanden. Beeldfragmenten ondergaan hiervoor verschillende processen, waaronder het voorgenoemde pixel binningsproces.

Pixel binning

Zoals eerder aangehaald kiezen moderne smartphones ervoor om pixels te groeperen in superpixels waarop vervolgens aan pixel binning gedaan wordt. Het effect hiervan is het verzachten van observeerbare ruis, zoals zichtbaar op afbeelding 2.2. Beelden die genomen zijn in donkere plekken kunnen daarmee een hogere waarneembare beeldkwaliteit krijgen. Een bijkomend effect van dit proces is de kleinere voetafdruk van de verwerkte beelden doordat er een kleiner aantal pixels overblijven. In het geval van smartphones wordt vaak gekozen om te binnen richting 12 megapixels, wat tussen de schermresoluties 4K Ultra HD (ongeveer 8 megapixels) en 8K Ultra HD (ongeveer 33 megapixels) ligt. Deze resoluties worden steeds vaker

gebruikt in televisies (Statista, 2019).

Transformatie van beelden

Beeldfragmenten kunnen naast pixel binning op andere manieren bewerkt worden om meer specifieke informatie te benadrukken en op te slaan. Een voorbeeld hiervan is Fouriertransformatie, zoals beschreven in het onderzoek van Olaoye e.a. (2024). Door middel van deze techniek worden dominante ruimtelijke frequenties benadrukt wat het identificeren van patronen, randen en texturen van beelden mogelijk maakt. Beelden kunnen daarnaast ook geschaald of gedraaid worden voor geometrische correcties of ter uitlijning van afbeeldingen. Vervolgens kunnen beelden opgedeeld worden naar gelang de kleur, textuur of randen.

2.2.4. Klassieke objectherkenningsalgoritmen

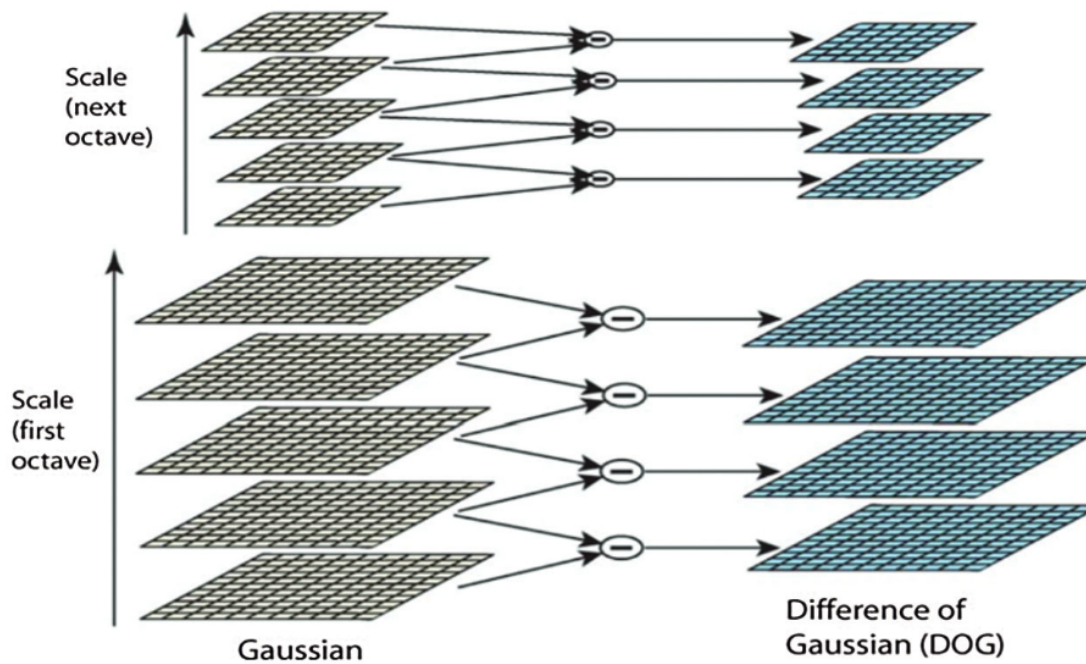
Na het verwerken van beeldfragmenten komt verdere segmentatie aan bod door middel van klassieke objectherkenningsalgoritmen, wat in vakliteratuur vaak als *feature extraction* vernoemd wordt. Met behulp van deze algoritmen kunnen we kenmerken van objecten detecteren om vervolgens te classificeren en vergelijken met andere data. Het verschil van deze algoritmen met moderne objectherkennings technieken is het gebrek aan gebruik van machine learning, wat verder aan bod komt in hoofdstuk 2.3.

Histogram van Georiënteerde Gradiënten

Histogram van Georiënteerde Gradiënten (HOG) is een simpel algoritme voor kenmerkbeschrijving van objecten voor feature extraction, met als doel om lokale vormen en textuurinformatie vast te leggen (Saher e.a., 2023). Allereerst filtert het HOG-algoritme de kleuren uit de afbeelding zodat enkel de grijswaarden overblijft. Vervolgens splitst het algoritme de beelden op in kleine overlappende cellen en berekent het de gradiënten van elke cel in zowel de horizontale als de verticale richting. Eveneens legt het de verandering in pixelintensiteit en diens orientatie vast. Op basis van deze berekeningen kan een histogram van celoriëntaties gegenereerd worden om de lokale gradiëntverdeling met informatie over randen en textuur weer te geven.

Schaalinvariante kenmerkentransformatie

Voor situaties waarin schaal- en rotatie-invariantie belangrijk zijn kan het beter zijn om gebruik te maken van SIFT. Tamara e.a. (2022) beschrijft Schaalinvariante kenmerkentransformatie (SIFT) als een complexere techniek voor het identificeren van lokale kenmerken in een afbeelding. Het eerste deel van dit algoritme bestaat uit het berekenen van de minima en de maxima uit het verschil in Gaussianen (DoG) met verschillende standaardafwijkingen, zoals gevisualiseerd op afbeelding 2.3. Daarna worden uitschieters weggefilterd door middel van een wiskundige Hessische matrix. Een normalisatie van het aantal kenmerken kan vervolgens behaald worden



Figuur 2.3: Visualisatie van het verschil in Gaussianen (Lowe, 2004)

door een minimum contrast van de verschillende standaardafwijkingen te hanteleren. Het toepassen van oriëntatietoewijzing verzekert hierbij de eigenschap van rotatie-invariantie. Uit deze stappen ontstaan enkele SIFT-beschrijvingen die vergelijking met andere afbeeldingen mogelijk maken om te bepalen of ze tot dezelfde categorie horen. Dit gehele proces kent echter aanzienlijke verwerkingstijd waardoor het gebruik in real-time applicaties beperkt blijft.

Versnelde robuuste functies

Een alternatief voor SIFT zijn Versnelde robuuste functies (SURF), een eenvoudiger algoritme met een focus op snelheid over het bereiken van de hoogste nauwkeurigheid (Wu e.a., 2013). Het concept van SURF is vergelijkbaar met dat van SIFT, echter maakt SURF gebruik van een andere methode om kenmerkbeschrijvingen te genereren. In dit algoritme wordt bewust niet gekozen voor DoG en wordt enkel gebruik gemaakt van een Hessische matrix voor de detectie van kenmerken. Daarnaast maakt het SURF-algoritme gebruik van een eenvoudigere Haar-waveletberekening met de x- en y-richtingen van elk gebied binnen een afbeelding.

2.2.5. Uitdagingen en beperkingen

Hoewel er vele mogelijkheden ontstaan door het gebruik van de voorgenoemde voorbeelden van klassieke algoritmen, kent de technologie ook enkele limitaties. De voorgenoemde algoritmen hebben vaak te maken met moeilijkheden in het detecteren van objecten in complexe situaties (Luz & Luz, 2024). Variaties in belichting, drukke achtergronden rond objecten en occlusie van objecten verminderen

de nauwkeurigheid van deze traditionele algoritmen. Machine learning kan hierop inspelen door bijkomende attributen toe te wijzen aan objecten.

2.3. Machine learning

Het vorige hoofdstuk definieerde de werking van objectherkenning en diens situering binnen de artificiële intelligentiesector samen met enkele voorbeelden van klassieke algoritmen. Deze algoritmen kennen enkele beperkingen waar deep learning een antwoord op kan bieden. De komst van machine learning (ML) kent buiten objectherkenning vele andere toepassingen zoals het ontstaan van grote taalmodellen, wat later aan bod zal komen.

2.3.1. Hoe machine learning en deep learning werkt

Net zoals objectherkenning een subset van computer visie is kan deep learning (DL) gezien worden als een subset van machine learning. Het domein artificiële intelligentie overkoepelt beide vakgebieden. Concreet maken ML-technieken gebruik van statistieken en optimalisatiemethoden om grote hoeveelheden data te verwerken (Pennone e.a., 2024). Hiermee ontstaat de mogelijkheid om uitkomstvoorspellingen te maken op basis van invoer van nieuwe data. Het algoritme groepeerde deze data in subgroepen en classificeert vervolgens elke subgroep. Daardoor ontstaat de mogelijkheid om onder andere uitkomstvoorspellingen te maken van toekomstige data. In context van deze paper gaat het daarbij concreet over het classificeren van sportgerelateerde objecten en omgevingen op basis van eerder verwerkte datasets.

2.3.2. Deep learning

Een complexere manier van dataverwerking kan gebeuren aan de hand van DL-algoritmen, een subset van machine learning dat gebruik maakt van neurale netwerken om informatie te verwerken (Božić, 2024). De algoritmen verwerken data door ze op verschillende niveaus te behandelen, in tegenstelling tot simpelere ML-technieken. In het geval van objectherkenning bestaan de lagere niveaus van een beeldfragment uit randen en vormen, terwijl hogere niveaus de kenmerken uit de lagere niveaus combineren om het object te identificeren.

Voordelen ten opzichte van machine learning

Deep learning-technieken produceren vaak betere resultaten ten opzichte van machine learning (Ahmed e.a., 2023). Bovendien krijgen ze vaak de voorkeur door volgende redenen:

Structuur van de gebruikte data In vele gevallen is de gebruikte data ongestructureerd wegens de verschillende soorten media waarin ze opgeslagen zijn. De meeste ML-algoritmen hebben moeite met deze data te verwerken waardoor de data vaak onbenut blijft voordat men ervoor kiest om deep learning toe te passen.

Nieuwe kenmerken genereren Het belangrijkste voordeel van deep learning vergeleken met andere ML-algoritmen is het vermogen om nieuwe kenmerken te genereren uit de beperkte lijst van kenmerken in de oorspronkelijke dataset. De DL-algoritmen gebruiken daarbij neurale netwerken doorheen het hele proces waardoor het steeds kan bijleren. Dit leidt tot een optimalisatie van alle relevante parameters wat uiteindelijk resulteert in een verbeterde nauwkeurigheid.

Zelfstandigheid Bovenop het genereren van nieuwe kenmerken kunnen DL-technieken op een zelfstandige basis aan resultaten bekomen. De algoritmes zoeken daarbij automatisch in de gegevens uit de trainingsdata zonder expliciete instructies te krijgen. Dit is mogelijk door kenmerken te vergelijken en combineren wat het leerproces bevordert.

Schaalbaar Het proces is zeer schaalbaar door de manier waarop het grote hoeveelheden aan data kan verwerken. Tot gevolg heeft het de mogelijkheid om steeds nauwkeurigere resultaten te produceren bij allerlei verschillende soorten data van verschillende groottes. Bij grotere datasets ontwikkelt het algoritme een beter begrip voor context en een grotere lijst van functionaliteiten.

Nadelen van deep learning

Hoewel DL-technieken vele verbeteringen aanbiedt ten opzichte van machine learning-algoritmen, kent het ook enkele nadelen waardoor het gebruik ervan beperkt wordt bij grotere organisaties:

Hogere rekenkracht DL-algoritmen maken gebruik van aanzienlijk grotere hoeveelheden data ten opzichte van ML-technieken. Dit vergt sterkere computers met meer rekenkracht en opslag.

Afhankelijkheid van data Net zoals machine learning-algoritmen kennen DL-technieken een afhankelijkheid van de gebruikte data. De resultaten van deze technieken zijn afhankelijk van zowel de kwaliteit als de kwantiteit van deze datasets. Bovendien vereist dit data die gelabeld is, wat kostelijk of in sommige domeinen zelfs onhaalbaar te verkrijgen is (Olaoye & Potter, 2024).

Complexiteit In bepaalde situaties zijn de gebruikte algoritmen complex te interpreteren, deels door de complexiteit van de gebruikte data. Het is daardoor moeilijker om de beslissingen en resultaten van deze technieken te traceren. Dit vereist het hertrainen van het algoritme om fouten recht te zetten.

2.3.3. Moderne objectherkenningsalgoritmen

Deep learning bouwt verder op klassieke objectdetectiealgoritmen om nauwkeurigheid te bevorderen en de uitdagingen zoals besproken in [Uitdagingen en beperkingen](#) te omzeilen. Door het concept van deep learning ontstaan vele nieuwe algoritmen voor het detecteren van objecten:

Convolutionele neurale netwerken)

Convolutionele neurale netwerken (CNN) hebben het vermogen om automatisch kenmerken te leren uit afbeeldingen aan de hand van convolutielagen (Nourmohammadi e.a., 2024). Ze zijn geoptimaliseerd om te werken met ruimtelijke structuren en hebben geen geheugen van voorgaande invoeren. Ondersteunende vectormachines (SVM) kunnen daarbij gebruikt worden voor classificatietaken zoals objectherkenning. Beslisbomen kunnen objecten identificeren door middel van een hiërarchische reeks beslissingen op basis van beschrijvingskenmerken. Tot slot kunnen K-nabije burens (KNN) objecten identificeren door kenmerken te vergelijken met kenmerken van gelijkaardige objecten in de trainingsdata.

Terugkerende neurale netwerken

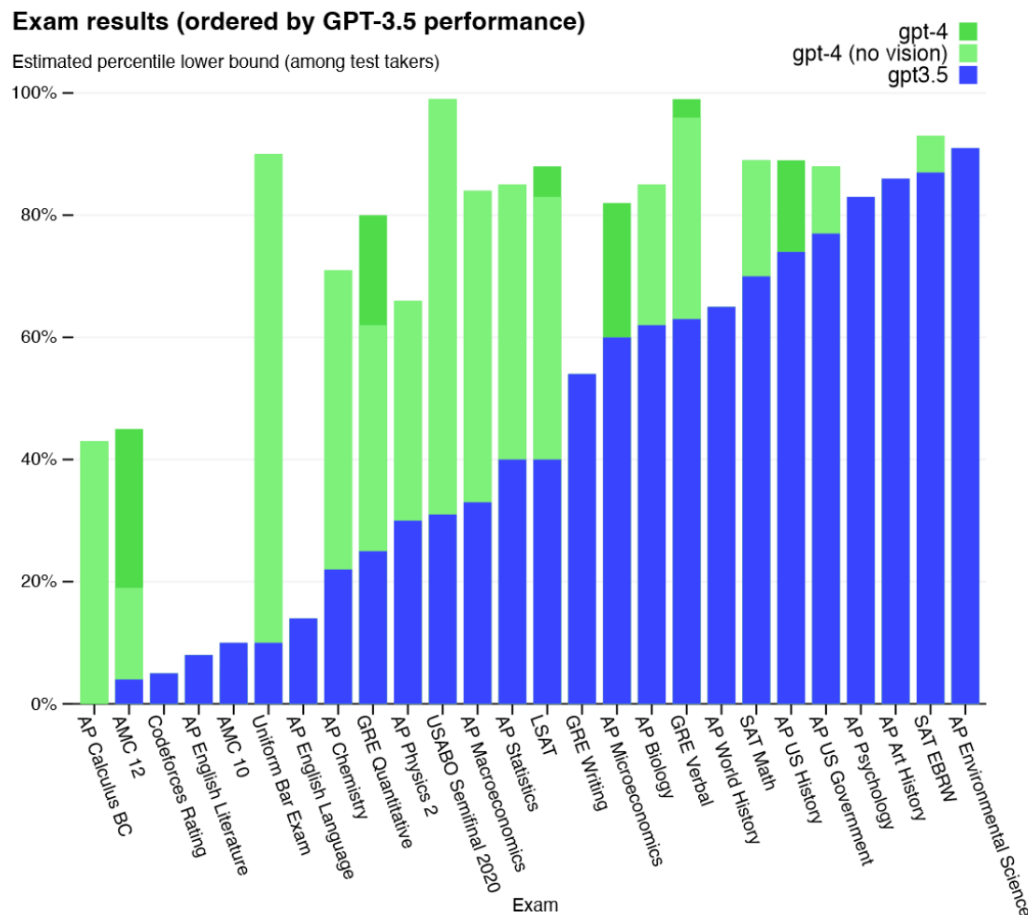
Terugkerende neurale netwerken (RNN) daarentegen worden eerder gebruikt voor doeleinden waar context van belang is. Ze zijn effectief in het onthouden van informatie doorheen verschillende stappen in een sequentie en zijn daarmee nuttig voor spraakherkenning en machinevertaling.

2.4. De werking van grote taalmodellen

Met een kennis over de manier waarop datasets gebruikt worden in machine learning kan tot slot het concept van generatieve AI verklaard worden. Generatieve AI verwijst hierbij naar de technologie van kunstmatige intelligentie om nieuwe gegevens voor te stellen op basis van trainingsdata (Gupta, 2023). Een voorbeeld hiervan is ChatGPT, een AI-chatbot die beschikbaar gesteld wordt door OpenAI. Gebruikers hebben daarbij de mogelijkheid om teksten te laten vertalen, broncode voor software te genereren of gedichten op te stellen. Dit alles gebeurt in een tekstgebaseerde interactieomgeving, waarbij gebruikers met de applicatie praten alsof ze communiceren in een chatapplicatie.

2.4.1. Vooruitgang van taalmodellen

OpenAI (2023), een onderzoeksinstituut met het doel om veilige en nuttige algemene artificiële intelligentie te ontwikkelen, beschrijft gunstige resultaten in haar technisch rapport over de nieuwere iteraties van het GPT-model met het trainen van steeds grotere datasets. De voorgenoemde generatieve AI-toepassing ChatGPT maakt daarbij gebruik van versie 3.5 van het model om verzoeken van gebruikers te vervullen. In het rapport haalt OpenAI aan dat versie 4 van haar GPT-model een score behaalt die in de top tien percent van de testnemers valt bij het afnemen van een gesimuleerde balie-examen. Versie 3.5 daarentegen scoort bij de laagste tien percent, wat een opmerkelijke evolutie is. Het rapport geeft verschillende vergelijkbare resultaten weer, een daarvan is zichtbaar op afbeelding 2.4 waarbij beide modellen enkele academische en professionele examens afleggen.



Figuur 2.4: Vergelijking tussen GPT 3.5 en GPT 4 bij het afleggen van verschillende academische en professionele examens. Bijkomende computer visie mogelijkheden helpen GPT-4 om betere resultaten te halen (OpenAI, 2023)

2.4.2. Versterkingsleren via menselijke feedback

Versterkingsleren via menselijke feedback (RLHF) heeft de mogelijkheid om de nauwkeurigheid van modellen te bevorderen na de initiële dataset verwerking. RLHF is een voorbeeld van versterkend leren waarbij ML-algoritmen geoptimaliseerd worden om nauwkeurigere resultaten te produceren die beter afgestemd zijn op specifieke verwachtingen (Balakrishnan, 2024). Dit proces vindt zich plaats na een dataset verwerkt is en er een AI-model tot stand is gekomen dat capabel is om taken uit te voeren. Met RLHF kunnen vakexperts of data ingenieurs de prestaties van het model beoordelen en corrigerende feedback geven. Deze feedback zorgt ervoor dat het model bijgestuurd wordt om specifiekere resultaten weer te geven.

2.4.3. Generatieve AI en deep learning

Generatieve AI bouwt verder op het concept van deep learning om grote taalmodellen (LLM) te ontwikkelen (Shen, 2024). LLM's hebben door middel van deep

learning de mogelijkheid om natuurlijke taal te begrijpen en suggesties te genereren. Deze AI-modellen richten zich specifiek op het verwerken en genereren van tekstuele data.

Tokens binnen een verzoek aan een taalmodel

Cope en Kalantzis (2023) beschrijven tokens in context van grote taalmodellen als volgt: “De elementaire eenheid van analyse in de LLM is het token. Soms is dit een woord, maar in het geval van woorden met samengestelde betekenissen, kan een token minder zijn dan een woord. Gereduceerd tot binaire notatie, abstraheert de machine tokens in een identieke, en voor de doeleinden van zijn analyse, betekenisloze vorm. Ze variëren alleen in het gekwantificeerde gewicht van hun nabijheid tot andere tokens(vectoren) in geschreven tekstcorpora.” Hiermee ontstaat de stelling dat tokens de maateenheid voor verzoeken aan taalmodellen vormen. Dit kan doorgetrokken worden voor multimedia zoals afbeeldingen en geluidsfragmenten, door de grotere inhoud aan data bestaan deze logischerwijs uit grotere hoeveelheden tokens.

3

Methodologie

Naast de stand van zaken vindt er zich ook een technisch aspect plaats in de vorm van een proof-of-concept. Dit hoofdstuk verduidelijkt het plan van aanpak van dit luik.

3.1. Requirements-analyse

Het doel van de requirements-analyse is om een lijst van vereisten op te stellen die de selectie van tools en de uitwerking van de proof-of-concept zullen aftoetsen. Om een beter begrip te krijgen over de wensen van personal trainers, en in het bijzonder de opdrachtgever van deze paper, vindt er zich een korte interview plaats met de co-promotor. Hieruit volgt een beter begrip voor de werking van personal training, de verwachtingen van cliënten en de workflow van coaches. Het resultaat hiervan is terug te vinden in de [Onderzoeksdoelstelling](#).

3.2. Selectie van tools

Na het bepalen van de vereisten volgt een opstelling van technologieën en platformen die gebruikt zullen worden bij het ontwikkelen van de proof-of-concept. In deze fase volgt de redenering waarom welke tools gekozen werden zoals afgetoetst met de vereisten. Hiermee kan een volgend gesprek met de opdrachtgever plaatsvinden om te verifiëren of aan de verwachtingen voldaan zal worden. Met een duidelijk begrip van welke tools gebruikt zullen worden kan de ontwikkeling van de proof-of-concept van start gaan.

3.3. Proof-of-concept

Eerst volgt een opzetting van de geselecteerde tools met documentatie om de werking te verduidelijken. Vervolgens worden de gevraagde functionaliteiten ge-

implementeerd, belangrijk hierbij is het regelmatig vragen van feedback van de co-promotor. Daarna volgt een demonstratie van het experiment om de ontdekkingen en resultaten mee te delen. Als de opdrachtgever een positieve reactie geeft komen enkele optimalisaties aan bod, waaronder het documenteren van de gebruikte hardware en het schrijven van een script om de reproduceerbaarheid van het experiment te bevorderen.

3.3.1. Visie

Dit onderdeel richt zich op het definiëren van de visie achter de uitgewerkte proof-of-concept. De aanhaling van de verwachte resultaten, diens doelpubliek en de voordelen van het gebruik ervan benadrukken de toegevoegde waarden van het technisch luik.

Beoogde resultaat

De proof-of-concept richt zich erop om moderne objectherkenningstechnieken en generatieve AI met elkaar te integreren om suggesties weer te geven op basis van ingescande sportgerelateerde objecten en omgevingen. Dit alles hoort te gebeuren op een kostefficiënte manier, ten gevolge hiervan ontstaat de keuze voor het gebruiken van een bestaand AI-model met vooraf getrainde datasets om zowel objectherkenning en het genereren van suggestieve trainingsschema's mogelijk te maken. Personal trainers horen de mogelijkheid te hebben om voorbij interacties van de gebruiker met de app in te zien en bij te sturen in het genereren van suggesties. Tot slot hoort het gebruik van de applicatie bewaard bijgehouden te worden om op elk gegeven moment de voorbij interacties en andere historische van gebruikers te consulteren. Hiermee heeft de personal trainer kostbare data van zijn cliënten zonder dit zelf administratief bij te houden of aan de cliënt te vragen dit op te schrijven en door te sturen. De proof-of-concept oogt daarmee op een mobiele applicatie dat de (administratieve) werklast van de personal trainer vermindert en bijstaat bij cliënten door middel van generatieve AI.

Architectuur

Het technisch luik bestaat uit drie lagen:

- **Een simpele Android-app** met de functionaliteit om foto's te maken en door te sturen naar een achterliggende service voor cliënten van personal trainers.
- **Een achterliggende service** met databank voor het bijhouden van de historie van voorbij interacties van gebruikers. De Android-app spreekt deze service aan om objectherkenning toe te passen op doorgestuurde foto's met als gevolg een suggestief trainingsschema terug te krijgen. Daarnaast hebben personal trainers de mogelijkheid om deze service aan te spreken om de historie van een gebruiker weer te geven en bij te sturen waar nodig.

- **Een AI-platform** dat aangeroepen wordt door de achterliggende service. Het platform gebruikt vooraf getrainde datasets om objectherkenning en generatieve AI mogelijk te maken.

Doelpubliek, innovatie en voordelen

Het gebruik van geavanceerde kunstmatige intelligentiemodellen is niet meer weg te denken en wordt steeds meer toegepast in de digitale wereld. Dit voorstel tracht deze trend door te trekken in de personal training door baanbrekende ontwikkelingen in objectherkenning en generatieve AI toe te passen. Hiermee krijgen personal trainers een zicht op het potentieel om kunstmatige intelligentietechnieken te integreren in de fitnessindustrie. Vervolgens kunnen ze ervoor kiezen om de aangetoonde functionaliteiten toe te passen in hun workflow om de werkdruk te verminderen of de ervaring van cliënten te bevorderen. De achterliggende service dient ter illustratie van een overkoepelend systeem om data van klanten bij te houden en te manipuleren. Er kan ook voor gekozen worden om het toepassen van generatieve AI te integreren in bestaande applicaties en de nieuwe functionaliteiten bloot te stellen in bestaande mobiele apps.

3.4. Conclusie en vooruitzichten

Tot slot volgt een korte conclusie met een terugblik op de resultaten ter antwoord op de gestelde onderzoeksvragen. Onduidelijkheden komen hierbij aan het bod, waarop enkele vooruitzichten en uitnodigingen voor verder onderzoek plaatsvinden.

4

Selectie van tools

Hierna volgen enkele beslissingen rond de gebruikte tools en technologieën in het technische luik van deze paper. De vereisten van de opdrachtgever staan hierbij centraal zoals besproken in de [Onderzoeksdoelstelling](#). Daarnaast komt ook een korte uitleg rond de keuze voor het te testen fitnessinstrument aan bod. Het volgende hoofdstuk beschrijft vervolgens de uitwerking van dit luik met de gekozen tools in de vorm van een proof-of-concept.

4.1. Android-app met Jetpack Compose

Een van de vereisten van deze bachelorproef is het frictieloos aanroepen van de achterliggende service. De proef is erop gericht om de gebruiker met een simpele foto alle functionaliteiten te kunnen gebruiken. Hierdoor zijn native en cross-platform applicaties meer geschikt boven progressive web apps. Native en cross-platform raamwerken zoals Jetpack Compose en Flutter bieden het gebruik van bestaande tools specifiek voor het besturingssysteem om foto's te maken en door te sturen. Gebruikers zullen daarmee de Android app kunnen gebruiken zoals ze eerder welke app zouden gebruiken om foto's op te laden, zij het Instagram of Google Photos.

4.1.1. Motivering voor Jetpack Compose

Jetpack Compose biedt een oplossing voor Android-specifieke mobiele applicaties en wordt daardoor als een native raamwerk aanschouwd. Bovendien past het beter bij de keuze voor Quarkus als achterliggend raamwerk, gezien beiden gebruik maken van het Gradle bouwsysteem en het Java ecosysteem. Een bijkomend voordeel is de betere prestaties doordat applicaties specifiek voor het Android besturingssysteem gemaakt zijn, wat de vereiste van een frictieloze gebruikerservaring garandeert.

4.2. Achterliggende service met Quarkus

Quarkus is een open source raamwerk bovenop het Java ecosysteem dat talloze functionaliteiten biedt aan softwareontwikkelaars om Java applicaties uit te werken. RedHat, een dochterondermering van IBM dat zich richt op het toegankelijk maken van open-source software, sponsort en ondersteunt dit initiatief. Bovendien dragen ontwikkelaars binnen RedHat bij aan het project met de ontwikkeling van Mandrel, een distributie van GraalVM specifiek voor de vereisten van Quarkus. Mede door deze ondersteuning kent Quarkus een penetratie in het door Java Spring-gedomineerde landschap van Java raamwerken.

4.2.1. Modern raamwerk

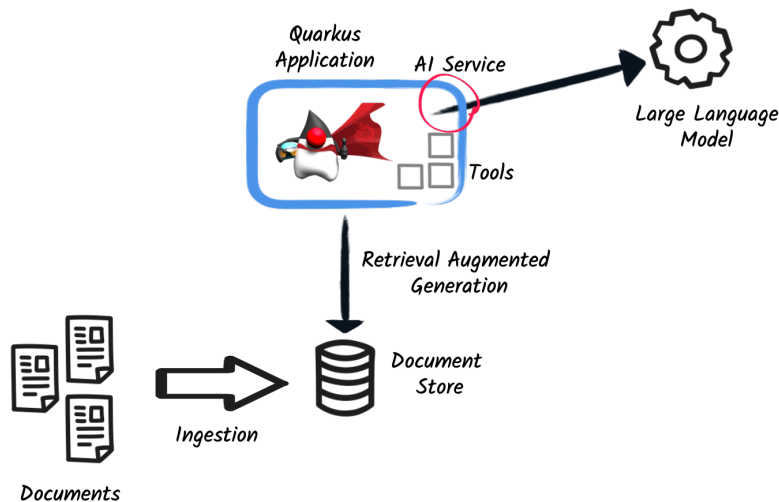
Quarkus is een modern raamwerk ontstaan in 2019 dat verder bouwt op industrie-standaarden zoals de Java Context & Dependency Injection CDI (CDI) en Persistence API (JPA) standaarden. Hierdoor is het makkelijk te interpreteren en verder op te bouwen voor bestaande Java-ontwikkelaars met expertise in back-end ontwikkeling. Bovendien heeft Quarkus het voordeel om deze standaarden te implementeren volgens hedendaagse vereisten met kennis opgedaan uit oudere raamwerken zoals Java Spring en Jakarta EE. Met een focus op Kubernetes en de bijkomende *Dev Services* wilt Quarkus zich differentiëren binnen het Java landschap.

4.2.2. Quarkus Dev Services

Aan de hand van (Quarkus, [g.d.](#)) ontstaat de mogelijkheid om op een snelle, iteratieve manier voort te bouwen zonder initiële configuratie. Zolang er geen bijkomende configuratie van afhankelijke instanties van software aanwezig is, bijvoorbeeld de login details van een databank, voorziet Quarkus van deze instanties. In het geval van de MariaDB databank, die gebruikt zal worden om de geschiedenis van gebruikers op te slaan, wordt daarbij achterliggend een Testcontainer opgestart. Testcontainers draaien op Docker, een technologie die zich richt op het draaien van software op een platform-agnostische manier, wat verder aan bod komt in subsectie [5.1.3](#).

4.2.3. Quarkus AI met Langchain4j

Langchain4j is de Java-implementatie van Langchain, een raamwerk dat het communiceren met grote taalmodellen gemakkelijk maakt. Het is best te vergelijken met Java Hibernate, de industriestandaard raamwerk om het communiceren met achterliggende databanken toegankelijker te maken (Langchain4j, [g.d.](#)). Quarkus AI bouwt hierop verder door een Langchain4j integratie te voorzien voor Quarkus applicaties om grote taalmodellen toegankelijk te maken. De situering van deze extensie is zichtbaar in afbeelding [4.1](#).



Figuur 4.1: Quarkus AI is een dienst waarmee Quarkus applicaties kunnen communiceren met grote taalmodellen. De document store stelt hierbij de databank voor met de geschiedenis van gebruikers in het opvragen van suggesties rond fitnessstoestelen. (Lowe, 2004)

4.3. Artificiële intelligentie met Vertex AI

Multimodale taalmodellen vereisen hoge rekenkracht om taken uit te voeren, hierdoor worden deze gebruikt via bestaande cloudinfrastructuur zoals Microsoft Azure en Amazon Web Services. Met de vereiste om een kostenefficiënte oplossing uit te werken komt Google Cloud naar voren als platform. Vertex AI, een dienst binnen Google Cloud, biedt daarbij bestaande taalmodellen met voorafgetrainde datasets en infrastructuur om zelf datasets te trainen.

4.3.1. Gemini 1.5 Pro

Gemini, het Google equivalent van OpenAI's GPT en Meta's Llama, biedt verschillende modellen voor gebruikers van Vertex AI. Versie 1.5 Pro is daarbij het meeste geschikt voor de proof-of-concept, gezien het zowel computer visie als generatieve AI biedt in één oplossing. Het model laat toe om te specificeren hoe betrouwbaar de resultaten horen te zijn, wat een belangrijk aspect is voor het voorstellen van suggesties aan cliënten van personal trainers. Tot slot kan het model aangeroepen worden vanuit de LangChain4j extensie.

4.4. Build tool

Moderne software projecten maken tegenwoordig gebruik van een *build tool*, wat helpt bij vele taken binnen het ontwikkelingsproces. Bovendien is dit een vereiste voor complexere applicaties waar enkele tussenstappen vereist zijn, zoals bij het bouwen van een Android-applicatie. Eveneens vergemakkelijkt het gebruik van een bouwsysteem om de proof-of-concept reproduceerbaar te houden.

4.4.1. Motivering

Kandhway en Theraja (2019) beschrijven de vijf grootste voordelen van het gebruiken van een build tool:

- **Het automatiseert het bouwproces van software.** Software moet eerst gecompileerd worden om leesbaar en uitvoerbaar te zijn door een machine. Dit is automatiseerbaar aan de hand van een build tool, wat zal helpen bij het *Reproducen van de proof-of-concept*.
- **Het vergemakkelijkt het beheren van afhankelijkheden.** De achterliggende Quarkus service en de Android app zullen gebruik maken van libraries om het ontwikkelingsproces te vereenvoudigen. Een voorbeeld hiervan is het gebruik van de Quarkus REST-extensie, wat het verwerken van appgebruikers' verzoeken versoepelt.
- **Het verzekert het correct uitvoeren van het bouwproces.** Het bouwproces bestaat uit verschillende stappen die ondergaan moeten worden. Een build tool helpt hierbij om sommige stappen voor ons te bepalen, zoals de volgorde waarin afhankelijkheden inladen.
- **Het bespaart tijd door taken in parallel uit te voeren.** De build tool splitst taken op om gelijktijdig uit te voeren. Dit versnelt het bouwproces.
- **Het is een vereiste voor het toepassen van continuous integration.** *Continuous integration* maakt het mogelijk om vooraf gedefinieerde bouwprocessen te lanceren eens er nieuwe code beschikbaar komt.

4.4.2. Gradle als buildtool

Binnen het Java-ecosysteem staan Apache Maven en Gradle centraal als build tools. Bij sommige oudere applicaties is het Ant-bouwsysteem nog in gebruik. Het Kotlin-ecosysteem kent de voorkeur naar Maven en Gradle gezien haar oorspronkelijk ontwerp om interoperabiliteit met Java te verzekeren. Google, de ontwikkelaars achter het Jetpack Compose framework en het Android besturingssysteem, hebben ervoor gekozen om uitsluitend met Gradle te werken wegens de beperktheid van Maven op vlak van bouwprocessen opstellen. Met de keuze voor Jetpack Compose zoals besproken in 4.1 zal daarmee ook gebruik gemaakt worden van Gradle voor de achterliggende Quarkus service. De reden hiervoor is om het bouwproces zo gestroomlijnd en consistent mogelijk te houden tussen de te ontwikkelende platformen, de visualisatie van dit bouwproces is te zien in afbeelding 4.2. Een bijkomend voordeel is de mogelijkheid om de Kotlin-syntax te gebruiken binnen Gradlebestanden, wat het meer consistent maakt met de syntax van de Quarkus en Jetpack Compose applicaties.



Figuur 4.2: De drie stappen in het Gradle bouwproces (Gradle, [g.d.](#))

4.5. Gekozen fitnessinstrument

Om de computer visie mogelijkheden uit testen zal gebruik gemaakt worden van afbeeldingen met dumbbells. Dumbbells zijn vrije gewichten die in de hand vastgenomen kunnen worden. Sporters gebruiken deze gewichten voornamelijk voor krachttraining en functionele training. Deze activiteiten zijn populair onder sporters volgens een opmeting van Thompson (2022). Bovendien zijn er vele ontwerpen van dumbbells in gebruik met of zonder gewichtsindicatie, wat een toegevoegde waarde biedt op het uittesten van de computer visie mogelijkheden.

5

Proof-of-concept

Na de **Selectie van tools** kan de proof-of-concept uitgewerkt worden. Allereerst worden de omgevingen opgezet om de uitwerking mogelijk te maken. Hierbij volgt telkens een kleine situering van de gebruikte architectuur en gemaakte keuzes. Daarna volgt de implementatie van de Android-app en de achterliggende service. Tot slot worden enkele details rond reproduceerbaarheid aangehaald. De uitgewerkte broncode kan teruggevonden worden op de GitHub repository van deze bachelorproef¹.

5.1. Voorvereisten

De proof-of-concept maakt gebruik van enkele bestaande tools om een werkende applicatie op te zetten. Deze dienen vooraf geïnstalleerd te worden.

5.1.1. Quarkus CLI

De Quarkus commandline-interface tool (Quarkus CLI) kan gebruikt worden om een Quarkus-applicatie te genereren. Deze tool kent vele installatiemogelijkheden, in deze bachelorproef werd ervoor gekozen om de package manager Chocolatey te gebruiken. Na het installeren van deze tool wordt **Opzetten van de Quarkus omgeving** mogelijk. Het kent de bijkomende functionaliteiten om extensies toe te voegen, bij te werken of te verwijderen. Bovendien bevordert het het updateproces van het raamwerk zelf door voorgedefinieerde acties van het Quarkus-team toe te passen tussen elke update in. Hierdoor kan deze proof-of-concept met de steeds recentste versies van Quarkus 3 en haar extensies uitgevoerd worden.

¹<https://github.com/+2324-bachelorproef-cantaertmaurice/tree/master/proof-of-concept>

5.1.2. Java Software Development Kit

Naast de Quarkus CLI maakt een Quarkus applicatie ook gebruik van de Java Software Development Kit (Java SDK). De SDK bevat een verzameling van functionaliteiten en documentatie die softwareontwikkelaars gebruiken om applicaties op te zetten. Binnen het Java ecosysteem zijn er verschillende distributies van de Java SDK, met telkens andere implementaties van dezelfde functionaliteiten. De voorkeur gaat in dit geval uit naar de Adoptium Eclipse Temurin distributie van Java 21, de huidige lange-termijnondersteuning versie. Deze distributie is open-source en vereist geen betalende licentie.

De Java Virtuele Machine

Raamwerken binnen het Java ecosysteem, zoals Quarkus en Jetpack Compose, worden in de Java programmeertaal geschreven. Ze maken daardoor gebruik van de Java SDK en de Java Virtuele Machine (JVM). Concreet schrijven softwareontwikkelaars een applicatie bovenop deze raamwerken in programmeertalen bovenop de JVM, vaak is dit Java zelf. Jetpack Compose is hier een uitzondering op waarbij softwareontwikkelaars gebruik maken van de Kotlin programmeertaal, zoals aangeraden door de ontwikkelaars achter Android. Een build tool, in dit geval de Gradle build tool, spreekt een gespecialiseerde compiler aan om de geschreven code om te vormen naar code die leesbaarder is voor computers. Het resultaat hiervan is Java bytecode die de JVM kan interpreteren en uitvoeren. Android telefoons en servers gebruiken Java Virtuele Machines die ontworpen zijn voor hun interne infrastructuur en het besturingssysteem.

5.1.3. Docker

Docker is een systeem vergelijkbaar met de Java Virtuele Machine. Het bevat alle benodigdheden om software te draaien binnen gecontaineriseerde instanties die ongeacht van de infrastructuur en het besturingssysteem op een reproduceerbare manier draaien. Hiermee ontstaat de mogelijkheid om instructies te schrijven om bepaalde diensten op te zetten op een omgeving-agnostische manier, dit in de vorm van *Docker images*. De Docker Engine voert de instructies van de Docker images uit op een manier waarin images geen zicht hebben op andere draaiende images en de rest van het systeem.

Docker Desktop

Docker Desktop is de Docker-toepassing die gebruikt zal worden voor deze proof-of-concept. Met deze applicatie kunnen Docker images opgezet worden om te draaien in Docker containers. Om dit te bereiken creëert de Docker Desktop applicatie intern een virtuele machine die Linux draait waarop deze containers vervolgens draaien.

5.2. Omgevingen opzetten en testen

De drie lagen van de **Architectuur** kennen verschillende architecturale beslissingen, deze worden hieronder toegelicht.

5.2.1. Opzetten Google Gemini omgeving

Zoals verklaard in **Selectie van tools** biedt Google Vertex AI allerlei taalmodellen ter beschikking voor Google Cloud gebruikers. Een omgeving opzetten is triviaal, een gebruiker hoeft daarbij enkel een Googleaccount en project aan te maken op Google Cloud. Daarna kiest de gebruiker ervoor om de Vertex AI-mogelijkheden en bijhorende taalmodellen aan te zetten. Voor prototypes is het mogelijk om € 280 aan krediet aan te vragen, waarmee nieuwe klanten de functionaliteiten van Gemini gratis kunnen uitproberen. Ten slotte hoort de server toegang te krijgen tot Vertex AI, dit kan door het volgen van de *Set up Application Default Credentials* handleiding voor lokale ontwikkelingsomgevingen².

5.2.2. Opzetten van de Quarkus omgeving

De Quarkus commandline-interface tool Quarkus CLI maakt het genereren en beheren van Quarkus-applicaties mogelijk. Gezien we gebruik willen maken van het Kotlin-ecosysteem om consistent te blijven met onze Android-app zal de Kotlin extensie toegevoegd worden samen met enkele andere Quarkus-extensies:

- **Hibernate ORM Panache Kotlin** zal de gegevens van de gebruikers en personal trainer opslaan in de databank op een Kotlin-gerichte wijze.
- **Java Database Connectivity (JDBC)** verzorgt de connectie met de databank zelf. JDBC haalt de inloggegevens van de databank op uit de application.properties bestand van de Quarkus-app. Als deze niet gedefinieerd zijn, dan zal Quarkus zelf een databank opstarten in een Docker-container en dit doorgeven aan JDBC.
- **Hibernate Validator** vergemakkelijkt het valideren van binnenkomende data van gebruikers. Langs deze weg verplicht de API de gebruiker om een foto mee in het verzoek te versturen.
- **LangChain4j**, tenslotte, maakt de communicatie met de Google Gemini-omgeving mogelijk.

Met de voornoemde specificaties in het achterhoofd gehouden kan deze commando uitgevoerd worden om het Quarkus-project op te zetten:

²<https://cloud.google.com/docs/authentication/provide-credentials-adc>

```
1 quarkus create app mauricecantaert:bachelorproef-ti-quarkus \
2   --extensions=kotlin,rest,rest-jackson,rest-multipart,hibernate-
   orm-panache-kotlin,hibernate-validator,jdbc-mariadb,quarkus-
   langchain4j-core
3   --gradle-kotlin-dsl
```

Codefragment 5.1: Commando om het Quarkus project met bijhorende extensies en configuratie te genereren

De werking van een Quarkus API

De achterliggende service is geschreven volgens de principes van de REST-architectuur. REST, of Representational State Transfer, is een vorm van een Application Programming Interface (API) dat het communiceren tussen apparaten via het internet standaardiseert (Doglio, 2018). De voornaamste vereiste van een REST-applicatie is het behandelen van elk verzoek als nieuwe taak die telkens reproduceerbaar is. Daaropvolgende verzoeken kennen geen geschiedenis van voorgaande verzoeken, waardoor het noodzakelijk is om steeds alle data mee te geven in één verzoek.

Het aanspreken van functionaliteiten binnen de Quarkus API

Elke bron, of type verzoek, krijgt volgens de REST-architectuur een Uniforme Bron-identificatie (URI) die gebruikt kan worden om de achterliggende dienst aan te spreken. Deze Quarkus API kent daarmee twee bronnen, één voor de gebruikers en één voor personal trainers. Gebruikers krijgen daarbij de mogelijkheid om fitnessinstrumenten te detecteren en daarvoor suggesties te laten genereren. Personal trainers krijgen zicht op de geschiedenis van gebruikers en kunnen de suggesties bijsturen waar nodig. Concreet spreekt de Android-app de achterliggende service aan met de gewenste URI's.

5.2.3. Opzetten Jetpack Compose Android-app omgeving

Clënten horen de mogelijkheid te hebben om de opgezette Quarkus applicatie op te roepen vanuit een Android applicatie. Gebruikers kiezen daarvoor om een nieuwe foto te maken of een bestaande te selecteren uit de galerij van hun telefoon. Het Android besturingssysteem verplicht applicaties om hiervoor eerst toestemming te vragen aan de gebruiker, wat geïmplementeerd zal worden door gebruik te maken van het bestaande proces dat het besturingssysteem beschikbaar stelt. Tot slot zal Retrofit, een bestaande oplossing om REST-verzoeken te versturen en ontvangen, de connectie met de Quarkus dienst behandelen. Android Studio, de geïntegreerde ontwikkelaarsomgeving voor ontwikkelaars op het Android platform, biedt een template aan voor het genereren van een Android project. Deze maakt standaard gebruik van het Gradle bouwsysteem, wat de Quarkus api tevens ook gebruikt.

5.2.4. Gemini AI testen

Vertex AI biedt een test interface vergelijkbaar aan de werking van Langchain4j. Dit is belangrijk om de eerste stap in het gebruiksverloop van klanten te testen. Als het gebruik hiervan een ongunstig resultaat zou opleveren, betekent dit dat het gebruikte taalmodel ongepast is voor deze proof-of-concept. Hierna volgen enkele testen van het Gemini 1.5 Pro platform, de resultaten worden gegoten in een vooraf gedefinieerde JSON-formaat. Het model krijgt de instructie om een schatting te maken voor onzekere eigenschappen, zoals wanneer het gewicht niet duidelijk vermeld. In elke test wordt gekozen voor een afbeelding gemaakt met een andere telefoon om verschillende gebruikers te simuleren.

Eerste test

De eerste afbeelding 5.1 bevat enkel de bovenkant van een dumbbell, de dumbbell zelf is dus niet triviaal te detecteren door het model. Het gewicht daarentegen is wel duidelijk zichtbaar, namelijk 16 pond. Gemini geeft een gunstig resultaat en detecteert het onderwerp en het gewicht:

```
1 {  
2     "name": "DUMBBELL",  
3     "type": "FREE_WEIGHT",  
4     "weight": 16,  
5     "weight_measurement": "LBS",  
6     "unknown_fields": []  
7 }
```

Codefragment 5.2: Resulterende JSON-data van de eerste computer visie test.



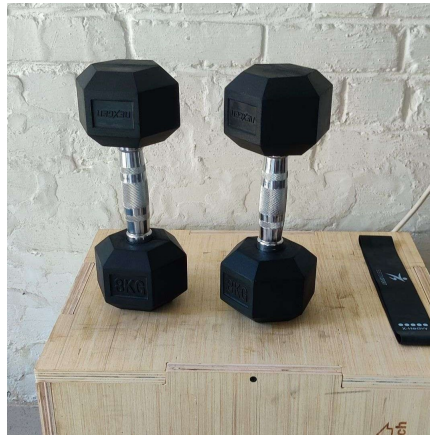
Figuur 5.1: De bovenkant van een 16 pond dumbbell, de foto werd genomen met een Apple iPhone 14.

Tweede test

Voor de tweede test bevat afbeelding 5.2 de zijkant van twee dumbbells. Echter is het gewicht, 8 kilogram, in dit geval minder duidelijk zichtbaar, Gemini geeft wederom een gunstig resultaat en detecteert zowel het onderwerp als het gewicht:

```
1 {  
2     "name": "DUMBBELL",  
3     "type": "FREE_WEIGHT",  
4     "weight": 8,  
5     "weight_measurement": "KG",  
6     "unknown_fields": []  
7 }
```

Codefragment 5.3: Resulterende JSON-data van de tweede computer visie test.



Figuur 5.2: De zijkant van twee 8 kilogram dumbbells, de foto werd genomen met een OnePlus 8T.

Laatste test

Ten slotte bevat de laatste test afbeelding 5.3 met een dumbbell met een gewicht van 5 kilogram, dit gewicht staat echter niet beschreven op het voorwerp. Gemini geeft hierbij een correcte inschatting van het gewicht, maar deelt mee dat het slechts om een inschatting gaat:

```
1 {  
2   "name": "DUMBBELL",  
3   "type": "FREE_WEIGHT",  
4   "weight": 5,  
5   "weight_measurement": "KG",  
6   "unknown_fields": [  
7     "weight"  
8   ]  
9 }
```

Codefragment 5.4: Resulterende JSON-data van de laatste computer visie test.



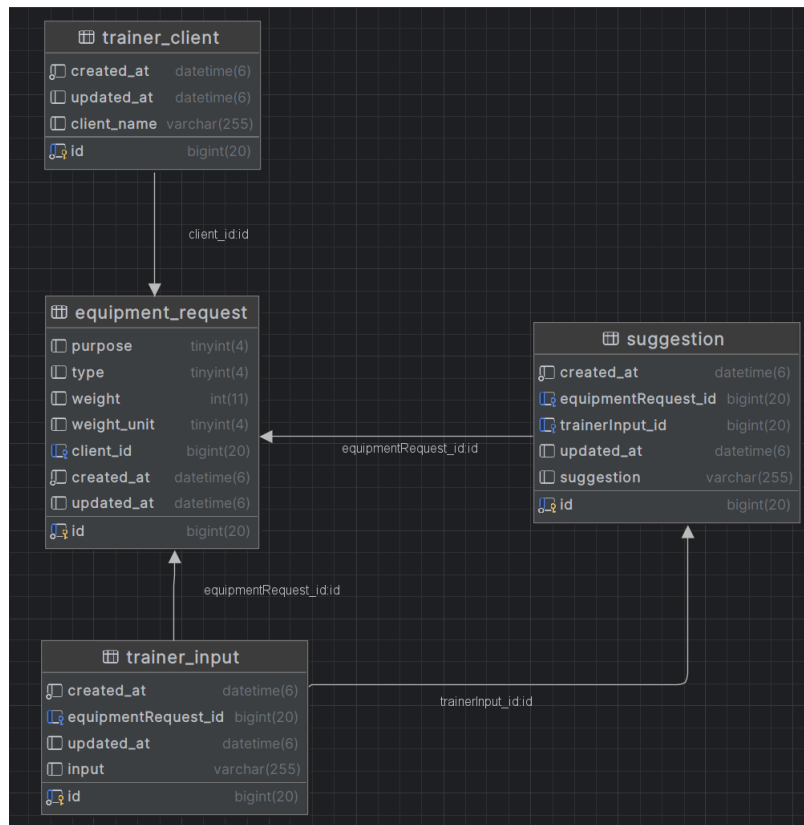
Figuur 5.3: Een zijaanzicht van een 5 kilogram dumbbell, de foto werd genomen met een Samsung S23 Ultra.

5.3. Functionaliteiten uitwerken

De resultaten uit de vorige stap bewijzen de mogelijkheden van het Gemini-platform, hiermee ontstaat de zekerheid dat de proof-of-concept hier verder op zal kunnen bouwen. Eerst volgt de opzet van het domeinmodel in de achterliggende service, dit dient om de historiek van de gebruiker bij te kunnen houden in de databank. Daarna volgt een uitwerking van de bronnen die de gebruiker en de personal trainer zal kunnen oproepen om de functionaliteiten te gebruiken. De gebruiker zal deze bronnen kunnen oproepen door middel van een kleine Android-app, wat in de laatste stap opgezet wordt. Een demo geeft het zicht op de bereikte functionaliteiten en de resultaten, wat besproken zal worden in de volgende sectie.

5.3.1. Domeinmodel opzetten

De achterliggende service hoort de historiek van de gebruiker bij te houden zodat de coach deze op elk moment hoort te kunnen raadplegen. Om dit te bereiken zullen de resultaten, zoals gezien in subsectie 5.2.4, ingedeeld worden als objecten. De Quarkus applicatie houdt de ingezonden afbeelding niet bij in de databank om de privacy van de gebruiker te bewaren, het verwerkt deze enkel zoals beschreven in [De werking van een Quarkus API](#). Vertex AI verwerkt verzoeken van de Quarkus applicatie en stuurt deze terug in het vooraf opgestelde JSON-formaat. Quarkus zal dit formaat terug omzetten naar het gepaste domein object om vervolgens te bewaren in de historiek van de gebruiker. Afbeelding 5.4 beschrijft de topologie van het uitgewerkte domeinmodel.



Figuur 5.4: Formaat om data op te slaan in de databank. Een cliënt vraagt een verzoek aan om meer informatie te krijgen over een object (equipment_request), op basis hiervan krijgt de cliënt een suggestie terug met optionele invloed van de personal trainer (trainer_input).

5.3.2. Functionaliteiten van de gebruiker

Na het opzetten van het domeinmodel kunnen de bronnen voor de gebruikers gedefinieerd en geïmplementeerd worden. Voor deze proof-of-concept is er hierbij slechts een bron blootgesteld aan de gebruiker, namelijk het doorsturen van een foto. In de achtergrond zal de applicatie de opgevraagde gegevens en de resultaten bijhouden in de databank. De bron stuurt de suggesties door naar de gebruiker zodat de persoon verder kan met zijn workout. Hierbij kijkt de applicatie naar de input van de personal trainer en past het de suggesties aan waar nodig. Belangrijk hierbij is dat de gebruiker hier geen extra input voor moet geven.

Bron definiëren

Quarkus biedt een triviale oplossing om bronnen te definiëren en bloot te stellen aan gebruikers. Elke bron voert vervolgens enkele stappen uit, in het geval van de bron voor cliënten begint dat met het verwerken van de afbeelding. De applicatie een verzoek naar Vertex AI met bijkomende gegevens zoals het vooraf gedefiniëerde JSON-formaat. Daarna slaat de applicatie het resultaat op in de databank en haalt het aansluitende instructies van de personal trainer op. Met de opgevraagde gegevens omtrent de foto en eventuele bijkomende instructies van de coach zal

Quarkus opnieuw een verzoek sturen naar Vertex AI om suggesties voor te stellen met behulp van generatieve AI. Alvorens de resulterende suggesties terug te sturen naar de gebruiker zal de applicatie deze opslaan in de databank.

Implementatie van de bron om suggesties te genereren

De bovenstaande bron kent de volgende uitwerking met behulp van enkele Lang-chain4j implementaties:

```
1 fun getSuggestions(  
2     @NotNull @RestForm("image") file: File,  
3 ): String {  
4     val user = User.byName("Maurice Canttaert") ?: throw  
        IllegalArgumentException("User can't be null")  
5  
6     // 1  
7     val equipmentDetails =  
        imageAnalyzer.getEquipmentDetails(file)  
8     val trainerInput = TrainerInput.latest(user,  
        equipmentDetails.type)  
9  
10    // 2  
11    equipmentDetails.client = user  
12    equipmentDetails.persist()  
13  
14    // 3  
15    val response =  
        suggestionGenerator.getSuggestion(equipmentDetails,  
        trainerInput)  
16    response.persist()  
17  
18    return response.suggestion  
19 }
```

Codefragment 5.5: Instructies van de bron om een afbeelding te analyseren en daarvoor suggesties te genereren

Allereerst zal het systeem de gebruiker opvragen volgens de gebruikersnaam. In de eerste stap zal daarmee de recentste input van de personal trainer voor het type fitness toestel opgevraagd worden. Ondertussen stuurt het systeem een verzoek aan de computer visie-dienst (imageAnalyzer) om het object te analyseren. Vervolgens, in de tweede stap, zal het systeem de resultaat van de fotooverwerker opslaan in de databank. Ten slotte zal het systeem suggesties laten genereren door middel

van een verzoek aan de generatieve AI-dienst (suggestionGenerator).

Implementation van de AI-gestuurde diensten

Zowel de imageAnalyzer als de suggestionGenerator uit codefragment 5.3.2 maken gebruik van een eigen implementatie van Langchain4j. Beide implementaties delen dezelfde fundamentele werking:

```

1      UserMessage.from(
2          ImageContent.from(encodedFile, mime),
3          TextContent.from(
4              """
5              A user wants to know what fitness equipment he's looking
6              at along with some details. Respond in the following JSON format.
7              If any details are unknown (such as weight or the type),
8              try to make an estimation.
9
10             Answer with a single raw JSON document, WITHOUT any
11             markdown markup such as ```json or ``` surrounding it.
12             The JSON document must contain:
13             - the type of equipment, such as a DUMBBELL or BARBELL,
14             in the `type` key as a string
15             - the purpose of equipment, such as STRENGTH or CARDIO,
16             in the `purpose` key as a string
17             - the weight of the equipment, in the `weight` key as a
18             number
19             - the weight measurement, either KG or LBS, in the
20             `weight_unit` key as a string
21             """).trimIndent()
22      )
23  )

```

Codefragment 5.6: Instructies om computer visie uit te voeren op een afbeelding om het object te analyseren en vervolgens uit te gieten in een vooraf gedefinieerd JSON-bestand

Bij het oproepen van de bron beschreven in subsectie 5.3.2 zal het systeem de functie *getEquipmentDetails()* oproepen. Allereerst zal de dienst de afbeelding verwerken zodat deze door het systeem gebruikt kan worden voor computer visie. Vervolgens wijst de dienst aan welk model toegepast zal worden, in dit geval is dat het Gemini 1.5 Pro model. Het verzoek wordt verstuurd naar het model om te verwerken, zoals beschreven in codefragment 5.3.2. Ten slotte zal de beeldverwerker het resultaat omvormen tot een gepast domeinobject dat verder gebruikt kan worden.

Implementatie van de Android applicatie

De Android-applicatie maakt gebruik van Retrofit om de bovenstaande bron op te roepen. Door een foto uit de galerij te selecteren kan de gebruiker deze dienst oproepen. Ten slotte geeft de app de gegenereerde suggesties weer in Markdown formaat.

5.3.3. Functionaliteiten van de personal trainer uitwerken

Een personal trainer krijgt toegang tot twee bronnen. Om de applicatie te gebruiken start hij met het raadplegen van de historie van de gebruiker. Met deze gegevens krijgt hij een globaal zicht op de vooruitgang van de gebruiker, bijvoorbeeld aan de hand van het gewicht van enkele gebruikte toestellen. Ook krijgt hij een zicht op de hoeveelheid toestellen die gebruikt werden, hoe vaak en over welke periode. Hiermee hoeven zowel de coach als de cliënt dit niet zelf bij te houden of te noteren. De tweede bron bestaat uit het manipuleren van toekomstige suggesties die aan de cliënt voorgesteld zullen worden, dit aan de hand van de gegevens uit de voorgenoemde bron. Met het toevoegen van enkele suggesties, zoals het focussen op een bepaalde oefening of meer repetities aan een lager gewicht, zal de gebruiker gepersonaliseerde suggesties krijgen bij het volgende gebruik van de Android app. Bovendien zal de coach deze gegevens niet zelf nog bij te houden, deze zullen telkens raadpleegbaar blijven in de databank bij volgende consultaties.

Implementatie

De implementatie van de voorgaande functionaliteit voor gebruikers haalt reeds de invoer op van de personal trainer. Langchain4j krijgt deze invoer en zorgt ervoor dat Vertex AI hier rekening mee houdt. Het definiëren van de bron en de manipulatie van de data in de databank is gelijkaardig met [Functionaliteiten van de gebruiker](#). De personal trainer kan deze bronnen oproepen door middel van Postman, een platform voor het rechtstreeks gebruiken van achterliggende API's. Het is mogelijk om een externe applicatie zoals een website of desktop applicatie te maken met een gebruiksvriendelijke interface, dit valt echter buiten de scope van deze proof-of-concept.

5.3.4. Opmerkingen

Tijdens het uitwerken van de proof-of-concept kwamen enkele uitdagingen aan bod:

- De automatische integratie van **Quarkus AI** met Langchain4j zorgt momenteel voor enkele problemen bij het uitvoeren van correcte computer visie. Een mogelijke oorzaak is het foutief verwerken van de afbeelding om door te sturen naar het Gemini taalmodel, waardoor sommige verwerkingen minder accuraat zijn. Om dit te mitigeren is dit deel van de proof-of-concept zonder Quarkus AI geschreven in pure Langchain4j instructies. Hiermee zijn de re-

sultaten zoals verwacht en overeenkomstig met de resultaten in de voorafgaande testen.

- De **vooraf gedefinieerde instructies** zoals beschreven in codefragment 5.3.2 vereisen specifieke voorwaarden om een gepast resultaat te bekomen. Hierbij moet de achterliggende service enkele zaken verduidelijken, zoals het gevraagde JSON-formaat of de context van het verzoek.

5.4. Reproducieren van de proof-of-concept

Om het reproducieren van deze proof-of-concept mogelijk te maken is er een Gradle-scripts die uitgevoerd kan worden om de applicatie op te starten. Belangrijk hierbij is dat er een **Docker** omgeving aanwezig is en draait, zodat Quarkus een test databank kan genereren. Bovendien is het belangrijk om een `.env` bestand aan te maken in de kernmap van het project met volgende waarden om de Gemini omgeving op te kunnen roepen:

```
1 GEMINI_PROJECT_ID=rare-nectar-414417
2 GEMINI_PROJECT_LOCATION=europe-west1
```

Codefragment 5.7: Omgevingswaarden met informatie over Gemini

5.5. Resultaten

De uitwerking van de proof-of-concept heeft een gunstig resultaat. Cliënten kunnen makkelijk suggesties opvragen op basis van ingezonden afbeeldingen zonder bijkomende invoer. Deze suggesties worden in Markdown formaat getoond op de telefoon van de gebruiker om een gepast overzicht te krijgen. De historiek van de gebruiker wordt bijgehouden en is ten alle tijde raadpleegbaar door de personal trainer. Bovendien kan de personal trainer volgende suggesties manipuleren door middel van eigen invoer in te geven. In subsectie 5.3.4 komen enkele opmerkingen aan bod.

6

Conclusie

De stand van zaken geeft een actueel beeld op de huidige situatie binnen de artificiële intelligentie. Hiervoor lag de focus op generatieve AI, computer visie en hoe deze gecombineerd zouden kunnen worden om een bijdrage te kunnen bieden in het werklevens. Deze bachelorproef oogde erop om een oplossing op maat te bedenken voor personal trainers in de sportsector met als doel de werkdruk te verminderen en de algehele efficiëntie te bevorderen.

6.1. Antwoord op de onderzoeksvragen

Aan de hand van de literatuurstudie kwam naar boven dat bestaande platformen zoals Google Vertex AI en OpenAI GPT gebruikt kunnen worden om aan nauwkeurige objectherkenning te kunnen doen. De eerste testen hadden daarbij een gunstig resultaat met het detecteren van dumbbells en hun gewicht. Dit alles werd gerealiseerd zonder zelf een dataset te trainen, in plaats daarvan werd gebruik gemaakt van bestaande datasets binnen het Gemini model.

Daarna werd onderzocht of het Gemini platform ook gebruikt kon worden om suggesties te genereren voor gebruikers. De verwachtingen hierbij werden overtroffen: Gemini biedt een kant-en-klare functie om suggesties te genereren in allerlei talen op een constructieve, gestructureerde wijze in het Markdown formaat. Dit formaat kan gebruikt worden om lijsten, tabellen en dergelijke visueel weer te geven aan gebruikers zonder bijkomend ontwerp van applicaties. Op deze wijze is het mogelijk om de gegenereerde suggesties van Gemini verder in te delen in allerlei Markdown formaten zonder enige aanpassing van de Android-applicatie.

Tot slot werd via de proof-of-concept het concept van een man-in-the-middle uitgewerkt. Personal trainers hebben met dit concept de mogelijkheid om suggesties

aan te passen aan de behoeften van de gebruikers. Suggesties kunnen aangepast worden in allerlei manieren, zij het de focus verleggen van bepaalde oefeningen van de schouders naar de borstkas, of zij het richten op hypertrofie in plaats van krachttraining.

Een bijkomend voordeel van dit systeem is het bijhouden van een historiek van de gebruikers. Cliënten en coaches hoeven daarbij niet zelf nog een dossier op te stellen met de gebruikte materialen, de frequentie van workouts en de beoefende activiteiten. Ook kan de vooruitgang gemeten worden door het bijhouden van de gebruikte gewichten. Het eindresultaat biedt daarmee meer dan het oorspronkelijk trachtte op te lossen.

6.2. Reflectie

De nauwkeurigheid van generatieve AI-modellen blijven een discussiepunt in de artificiële intelligentiesector, ook in deze bachelorproef is dat enigszins op te merken. Hoewel resultaten vaak nauwkeurig zijn zijn er occasioneel ook foutieve inschattingen. Hierop kan echter ingespeeld worden, wat gedemonstreerd werd bij de eerste testen rond het inschatting van de gebruikte gewichten. Het taalmodel kreeg de toestemming om onbekende waarden in te schatten, dit op voorwaarde dat er bijgehouden wordt welke waarden ingeschat werden. Opmerkelijk hierbij is dat het taalmodel vaker ging specificeren dat een waarde ingeschat werd ook al had deze het voorheen wel correct ingeschat. Dit komt wellicht door de *temperature*-instelling van het model, waarmee het model eerder kiest voor veiligere en voorspelbaardere antwoorden.

6.3. Vooruitzichten

Tijdens het opzetten van de Langchain4j implementatie kwamen de volgende onderzoeksvragen naar boven: Hoe kunnen vooraf gedefinieerde verzoeken aan AI-modellen dynamischer opgesteld worden? Kan een integratie met biometrische tracking in smartwatch en smartring wearables hier een meerwaarde in bieden om de verzoeken specifiek op het prestatievermogen van de sporter te richten? Tot slot, kan deze informatie opgeslagen en verwerkt worden door een taalmodel om de personal trainer een interpreteerbaarder en nauwkeuriger overzicht te geven op de vooruitgang van zijn klanten?



Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

Samenvatting

Vandaag de dag kent sport een prominente rol bij vele Vlamingen, dat kunnen we duidelijk aflezen uit de gepubliceerde statistieken van Statistiek Vlaanderen (2024). Hoewel sporten centraal staat voor gemiddeld één op de vijf Vlamingen, kampt de sportsector met een sterk trainerstekort. Om deze reden heeft overheidsinstantie Sport Vlaanderen (2023) reeds enkele campagnes gelanceerd om dit knelpunt op te lossen. Deze bachelorproef richt er zich op om een bijkomende oplossing te bieden voor fitnessclubs. Het uiteindelijke doel is om een proof-of-concept op zelfstandige basis te ontwikkelen waardoor leden in de startende fase minder beroep hoeven te doen op trainers. Hiermee horen sporters simpelweg hun uitrusting in te kunnen scannen door middel van een foto, en zouden ze een gepersonaliseerd trainingsschema voorgesteld krijgen. Maar hoe wordt specifieke apparatuur juist herkend op een nauwkeurige en behulpzame manier? Welke methodes kunnen gebruikt worden om het gepaste trainingsschema bij de juiste soort apparatuur voor de gebruiker voor te stellen? En hoe leggen we de link tussen het trainingsschema en de trainer om een steunmiddel aan te bieden voor zowel de sporter als de trainer? Om dit voorstel te realiseren wordt allereerst een literatuurstudie uitgevoerd om de huidige stand van zaken rond het detecteren van objecten (in dit geval fitness apparatuur) in kaart te brengen. De hoofdvraag hierbij is het onderzoeken van de gepaste manier om deze vorm van objectdetectie op een rendabele manier tot stand te krijgen. Hiermee ontstaat de mogelijkheid om een proof-of-concept uit te werken, gebruikmakend van een machine learning dataset, om gebruikserva-

ringen af te toetsen. Verwacht wordt dat deze tool een duidelijke hulp kan bieden voor software-ontwikkelaars binnen fitnessketens om functionaliteiten te ontwikkelen voor beginnende sporters.

A.1. Introductie

Vandaag de dag kent sport een prominente rol bij vele Vlamingen, dat kunnen we duidelijk aflezen uit statistieken van Statistiek Vlaanderen (2024). Hoewel sporten centraal staat voor gemiddeld één op de vijf Vlamingen, kampt de sportsector met een sterk trainerstekort. (Sport Vlaanderen, 2023) Met de toenemende groei van de artificiële intelligentiesector ontstaat de mogelijkheid voor bijkomende hulpmiddelen om de werkdruk van trainers te verlichten. Hierbij kunnen machine learning en objectdetectie bijdragen tot het personaliseren van een trainingsschema voor een gebruiker. Met de opkomst van steeds slimmere AI-tools kan de drempel om te starten bovendien verlaagd worden voor startende fitnessliefhebbers.

Door de explosie aan aandacht naar artificiële intelligentie en objectdetectie ontstaan er mogelijkheden om nauwkeuriger resultaten te bieden aan een kleinere technologische barrière. De vraag ontstaat dan ook of fitnessclubs deze technologieën kunnen toepassen om het makkelijker te maken voor startende sporters en de kans op het afhaken van trainers op de lange termijn te verlagen. Deze thesis probeert een ondersteunde oplossing te zoeken op een gebruiksvriendelijke en rendabele manier. Zo kunnen gebruikers een fitnessschema voorgesteld krijgen door het simpelweg inscannen van de gewenste fitnessapparatuur met een mobiele telefoon. Met machine learning kan deze apparatuur vervolgens gedetecteerd worden en kan de beginnende sporter beginnen aan zijn work-out. Trainers kunnen makkelijk hier op voortbouwen en het trainingsschema aanvullen en hebben van in het begin een zicht op de beschikbare apparatuur.

A.2. Literatuurstudie

A.2.1. Artificiële intelligentie

Hedendaagse artificiële algemene intelligenties komen voornamelijk voort uit vooraf getrainde taalmodellen. Aan de hand van natuurlijke taalverwerking (NLP) krijgt de intelligentie een inzicht in wat ervan verwacht en hoe het hoort te reageren op interactie van de gebruiker. De verwerkte gegevens bestaan uit een brede dekking aan informatie afkomstig van bronnen die kunnen variëren van nieuws artikelen tot taalexamen, aldus Liu e.a. (2019).

2022 kende een explosie in de waar te nemen mogelijkheden van kunstmatige intelligentie gestuurd door deze natuurlijke taalverwerkingsmodellen. OpenAI bracht haar derde generatie Generative Pre-trained Transformer (GPT) model uit en markeerde hiermee de nieuwe standaard van deep learning modellen. Het naar het model genoemde GPT-3 platform toonde belovende eerste resultaten bij het ex-

perimenteren rond het literaire aspect (Elkins & Chun, 2020), zo kon het model kwaliteitsvollere filosofische essays genereren dan bestaande geschreven essays. Ondanks deze resultaten bleek het model volgens Floridi en Chiriatti (2020) geen dieper begrip te kennen van het gegenereerde resultaat.

Slechts een jaar later, in maart 2023, kwam de volgende generatie uit van OpenAI's Generative Pre-trained Transformer (GPT) model. Het GPT-4 platform is in elk opzicht beter dan het GPT-3 platform volgens de resultaten van Katz e.a. (2023), en in een opmerkelijk opzicht, ook beter dan de gemiddelde persoon in vijf van de zeven deelgebieden in het Multistate Essay Exam (MEE), een gedeeltelijk toelatingsexamen om rechter te worden in de Verenigde Staten.

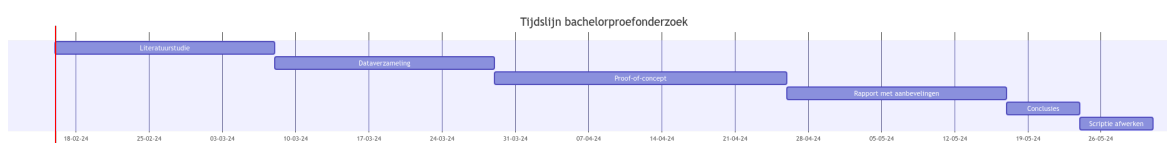
A.2.2. De technologische vooruitgang van objectdetectie in de cloud

Met de komst van het GPT-4 platform ontstond de mogelijkheid om naast tekst ook bestanden, waaronder foto's, in te voeren. Als reactie op het succes van OpenAI's (Chat)GPT kwam Google met haar eigen AI-platform, genaamd Bard (nu Gemini). Gemini kent nu ook al een integratie in Google's Cloud Vertex AI, wat de nauwkeurigheid en bruikbaarheid van object detectie in foto's sterk bevordert. (Google Cloud, 2024) Hiermee wordt het steeds interessanter om gebruik te maken van Google Cloud om gebruik te maken van machine learning.

A.2.3. Lokale machine learning

Echter zijn er ook nadelen bij clouddiensten zoals Google Cloud: niet elke organisatie is groot genoeg om het gebruik van clouddiensten te kunnen verantwoorden. Een alternatief hiervoor is het lokaal draaien van machine learning. Mogelijke opties hiervoor zijn PyTorch en TensorFlow. Hoewel deze tools ongetwijfeld sneller zullen draaien op servers in de cloud, is het ook mogelijk om hier gebruik van te maken op een lokale computer. Een moderne videokaart (hetzij de workstation Quadro-lijn of de consumenten-gerichte GeForce-lijn) van NVidia, of zelfs moderne videokaarten van AMD via het ROCm platform. (AMD, 2024) De nauwkeurigheid en snelheid zal lager liggen dan datasets gegenereerd door middel van cloudservices, maar aan een lagere kost. Het vergelijken van een cloudoplossing met een lokale machine tijdens de proof-of-concept fase van deze thesis kan hierbij interessant zijn. Echter zal de focus liggen op het gebruik van een bestaand model waarbij personal trainers niet zelf datasets hoeven te trainen.

A.3. Methodologie



A.3.1. Voorafgaande literatuurstudie

In de eerste fase vindt zich een voorafgaande literatuurstudie over het onderwerp plaats. Het praktische doel hiervan is om een verdere inzicht te krijgen op de werking van machine learning en haar rol bij object detectie op foto's. De nadruk ligt hierbij op de evolutie van deze werking en de bijkomende voordelen die deep learning technieken mogelijk bij kan brengen. Bovendien komt de werking van generatieve artificiële intelligentie aan bod, wat verder bouwt op deep learning.

A.3.2. Requirements-analyse

Vervolgens vindt er zich een requirements-analyse plaats om af te toetsen waar de focus moet liggen bij objectdetectie. Gebruikers die een app gebruikmakend van deze tool wensen te gebruiken tijdens het sporten horen eender welke gerelateerde apparatuur in te kunnen scannen. Om dit te realiseren moet gebruik gemaakt worden van een gepast multimodaal taalmodel zoals Gemini 1.5 Pro of ChatGPT 4o. Hetzelfde taalmodel hoort ook de mogelijkheid te bieden om suggesties te genereren op basis van de ingescande afbeeldingen. Bovendien horen personal trainers hier op in te kunnen spelen door deze suggesties bij te sturen. Ten slotte werden enkele test afbeeldingen verzameld om tijdens de proof-of-concept fase te testen.

A.3.3. Selectie van tools

Tijdens het opstellen van de shortlist komt het onderzoek naar de te gebruiken technologieën aan bod. Hierbij moet een beslissing komen over de te gebruiken raamwerken voor de mobiele applicatie, de achterliggende service en de te gebruiken multimodale taalmodel. Ook moet de meerwaarde van bouwsystemen zoals Gradle en Maven onderzocht worden op vlak van reproduceerbaarheid en automatisatie. Daarnaast volgt een korte analyse over het fitnessapparaat dat gebruikt zal worden in het uittesten van de proof-of-concept.

A.3.4. Proof-of-concept

Gedurende de derde fase wordt een proof-of-concept applicatie gebouwd voor Android-apparaten. Hiervoor worden er eerst datasets gegenereerd voor de objectdetectie functionaliteit. Deze datasets zullen gebruik kunnen maken van de machine learning capaciteiten van Google Cloud (Vertex AI). Hiermee kunnen we testen of het gebruik van voorafgetrainde datasets een mogelijke oplossing bieden voor het concept. De proof-of-concept toetst de mogelijke antwoorden die tijdens de requirements-analyse aan bod kwamen op de resultaten uit de proof-of-concept af.

A.4. Verwacht resultaten en conclusie

Er wordt verwacht dat de resultaten uit de proof-of-concept zal leiden tot een bruikbare tool die de drempel tot het starten van sport verlaagt en personal trainers erbij helpt de werkdruk te verminderen. Deze tool zal concreet de mogelijkheid bieden om fitness apparatuur in te scannen en detecteren door middel van bestaande deep learning technieken aangeboden door Google Gemini. Hiermee kunnen zelfstandige personal trainers snel aan de slag met door AI versterkte applicaties om bij te staan in het werklevens. De hypothese stelt dat de drempel om te starten in een fitnessclub zal verlaagd worden en dat coaches meer kunnen focussen op langetermijnschema's. Aan de hand van de resultaten ontstaat er een zicht op de huidige beperkingen en mogelijkheden alsook de gebieden waarop uitgebreid kan worden in de toekomst.

Bibliografie

- Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiaq, T., Rafa, N., Mofijur, M., Shawkat Ali, A. B. M. & Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, 56(11), 13521–13617. <https://doi.org/10.1007/s10462-023-10466-8>
- AMD. (2024, januari 11). *Installing TensorFlow for ROCm*. Verkregen 15 februari 2024, van <https://rocm.docs.amd.com/projects/install-on-linux/en/latest/how-to/3rd-party/tensorflow-install.html>
- Badoni, V. (2021, januari 27). *A Sense of Responsibility: Lidar Sensor Makers Build on NVIDIA DRIVE*. Nvidia Corporation. Verkregen 4 mei 2024, van <https://blogs.nvidia.com/blog/lidar-sensor-nvidia-drive/>
- Balakrishnan, A. (2024). Enhancing Data Engineering Efficiency with AI: Utilizing Retrieval-Augmented Generation, Reinforcement Learning from Human Feedback, and Fine-Tuning Techniques. *International Research Journal of Modernization in Engineering Technology and Science*, 6, 437–448. <https://doi.org/10.56726/IRJMET50070>
- Bert, V. V., Thomas, A. & Thomas, P. (2023, mei 6). *Syllabus IT-component Research Methods*, Hogeschool Gent.
- Blakeslee, S. (2004). The CRAAP Test. *LOEX Quarterly*, 31(3), 6–7. <https://commons.emich.edu/loexquarterly/vol31/iss3/4>
- Božić, V. (2024). Machine Learning vs Deep learning. <https://doi.org/10.13140/RG.2.2.16632.21762>
- Cope, B. & Kalantzis, M. (2023). A multimodal grammar of artificial intelligence: Measuring the gains and losses in generative AI. *Multimodality & Society*, 4(2), 123–152. <https://doi.org/10.1177/26349795231221699>
- Doglio, F. (2018). *REST API Development with Node.js: Manage and Understand the Full Capabilities of Successful REST Development* (2nd ed.) [Description based on publisher supplied metadata and other sources.]. Apress L. P.
- Elkins, K. & Chun, J. (2020). Can GPT-3 Pass a Writer's Turing Test? *Journal of Cultural Analytics*, 5. <https://doi.org/10.22148/001c.17212>
- Floridi, L. & Chiriatti, M. (2020). GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds and Machines*, 30(4), 681–694. <https://doi.org/10.1007/s11023-020-09548-1>

- Gajjar, H. & Sanyal, S. (2023). An In-depth Study of Lane Detection for Autonomous Cars using Computer Vision Techniques. *International Journal of Engineering Applied Sciences and Technology*, 8, 230–242. <https://doi.org/10.33564/IJEAST.2023.v08i02.035>
- Google Cloud. (2024, februari 16). *Innovate faster with enterprise-ready AI, enhanced by Gemini models*. Verkregen 16 februari 2024, van <https://cloud.google.com/vertex-ai?hl=en>
- Gradle. (g.d.). *Build Lifecycle*. https://docs.gradle.org/current/userguide/build_lifecycle.html
- Gupta, D. (2023). *The Potential of Generative AI* (A. Srivastava, Red.) [Description based on publisher supplied metadata and other sources.]. BPB Publications.
- Jin, X. & Hirakawa, K. (2012). Analysis and processing of pixel binning for color image sensor. *EURASIP Journal on Advances in Signal Processing*, 2012(1). <https://doi.org/10.1186/1687-6180-2012-125>
- Kandhway, H. & Theraja, P. (2019, september 6). *A general introduction to build tools*. Xebia Engineers. Verkregen 8 mei 2024, van <https://medium.com/xebia-engineering/a-general-introduction-to-build-tools-9070a47ed405>
- Katz, D. M., Bommarito, M. J., Gao, S. & Arredondo, P. (2023). GPT-4 Passes the Bar Exam. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4389233>
- Langchain4j. (g.d.). Introduction. <https://docs.langchain4j.dev/intro>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, *abs/1907.11692*. <https://doi.org/https://doi.org/10.48550/arXiv.1907.11692>
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91–110. <https://doi.org/10.1023/b:visi.0000029664.99615.94>
- Luz, H. & Luz, A. (2024). Detect and Describe Framework: Object Detection Boosting using Object Attributes.
- Moin, T. S. (2023). Overview of Computer Vision. <https://doi.org/10.13140/RG.2.2.13989.68327>
- Nourmohammadi, F., Parmar, C., Wings, E. & Comellas, J. (2024). Using Convolutional Neural Networks for Blocking Prediction in Elastic Optical Networks. *Applied Sciences*, 14(5), 2003. <https://doi.org/10.3390/app14052003>
- Olaoye, F. & Potter, K. (2024). Deep Learning Algorithms and Applications. *Dissolution Technologies*.
- Olaoye, F., Potter, K. & Doris, L. (2024). Computer Vision and Image Recognition Techniques. *Journal of Scientific Conference Proceedings*.

- OpenAI. (2023). *GPT-4 Technical Report* (tech. rap.). OpenAI. arXiv. <https://doi.org/10.48550/ARXIV.2303.08774>
- Pennone, J., Agüero, N. F., Martini, D. M., Mochizuki, L. & do Passo Suaide, A. A. (2024). Fall prediction in a quiet standing balance test via machine learning: Is it possible? (A. Mengarelli, Red.). *PLOS ONE*, 19(4), e0296355. <https://doi.org/10.1371/journal.pone.0296355>
- Quarkus. (g.d.). Dev Services Overview. <https://quarkus.io/guides/dev-services>
- Radojčić, V. & Cvetković, A. S. Advancements in Robotics, Smart Farming, and Computer vision: Revolutionizing Agriculture. In: 2023, oktober.
- Saher, M., Alsaedi, M. & Al Ibraheemi, A. (2023). Automated Grading System for Breast Cancer Histopathological Images Using Histogram of Oriented Gradients (HOG) Algorithm. *Applied Data Science and Analysis*, 78–87. <https://doi.org/10.58496/adsa/2023/006>
- Samsung. (2024, januari 31). *Galaxy S24 Specifications* (tech. rap.). Samsung Electronics Co., Ltd. https://www.samsung.com/latin_en/smartphones/galaxy-s24/specs/
- Shen, S. (2024). Application of large language models in the field of education. *Theoretical and Natural Science*, 34(1), 140–147. <https://doi.org/10.54254/2753-8818/34/20241163>
- Sport Vlaanderen. (2023, juni). *Jaarverslag 2022*. Verkregen 15 februari 2024, van <https://www.vlaanderen.be/publicaties/jaarverslag-sport-vlaanderen>
- Statista. (2019). *Worldwide market share of TV sales from 2017 to 2019, by screen resolution*. Verkregen 4 mei 2024, van <https://www.statista.com/statistics/818419/world-tv-market-share-by-type/>
- Statistiek Vlaanderen. (2024, maart 13). *Sportparticipatie*. Verkregen 5 mei 2024, van <https://www.vlaanderen.be/statistiek-vlaanderen/sport/sportparticipatie>
- Tamara, A. A.-S., Khalid, A. A., Issam, A. & Ra'ed, M. (2022). An Optimized Scale-Invariant Feature Transform Using Chamfer Distance in Image Matching. *Intelligent Automation & Soft Computing*, 31(2), 971–985. <https://doi.org/10.32604/iasc.2022.019654>
- Tasnim, S. & Qi, W. (2023). Progress in Object Detection: An In-Depth Analysis of Methods and Use Cases. *European Journal of Electrical Engineering and Computer Science*, 7(4), 39–45. <https://doi.org/10.24018/ejece.2023.7.4.537>
- Thompson, W. R. (2022). Worldwide Survey of Fitness Trends for 2022. *ACSM'S Health & Fitness Journal*, 26(1), 11–20. <https://doi.org/10.1249/fit.0000000000000732>
- Valve Corporation. (2024, maart 1). *Steam Hardware Survey* (tech. rap.). <http://web.archive.org/web/20240430110900/https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam>

- Video Electronics Standards Association. (2013, februar 8). *VESA Coordinated Video Timings (CVT) Standard – Version 1.2* (tech. rap.). Video Electronics Standards Association. <https://glenwing.github.io/docs/VESA-CVT-1.2.pdf>
- Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D. & Gong, S. (2013). A Comparative Study of SIFT and its Variants. *Measurement Science Review*, 13(3), 122–131. <https://doi.org/10.2478/msr-2013-0021>