



Faculteit Bedrijf en Organisatie

Handschriftherkenning voor doktersbriefjes m.b.v. Azure Computer Vision

Helmi Dayyeh

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

Promotor:
Irina Malfait
Co-promotor:
Wouter Baetens

Instelling: Arinti

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Handschriftherkenning voor doktersbriefjes m.b.v. Azure Computer Vision

Helmi Dayyeh

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

Promotor:
Irina Malfait
Co-promotor:
Wouter Baetens

Instelling: Arinti

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

Deze bachelorscriptie werd geschreven met het oog op het behalen van de titel: Bachelor in de Toegepaste Informatica.

Naar aanleiding van mijn stage bij De Cronos Groep, leek het de perfecte opportuniteit om in het domein van Cloud Computing en Artificiële intelligentie een onderzoek te starten. Hierbij kwam Arinti (een dochterbedrijf van De Cronos Groep) met een zeer interessant onderzoeksvoorstel. De Christelijke Mutualiteit had namelijk problemen met het lezen van doktersgeschrift op medische voorschriften. Hierbij kwam de vraag of dit proces eveneens geautomatiseerd kon worden door gebruik te maken van AI, en of er een verschil zou zijn in performantie/accuraatheid.

Aangezien het voorstel van dit onderzoek mij aangeboden werd vanuit Arinti, wil ik mijn co-promoter, Wouter Baetens bedanken om mij de kans te geven om dit zeer interessant onderzoek uit te voeren en enige vragen te beantwoorden die naar boven kwamen. Ook wil ik mijn stagebegeleider, Geert Baeke, bedanken om mij voor te stellen aan Arinti en zo aanleiding tot dit onderzoek te geven. Verder bedank ik mijn promotor, Irina Malfait, voor de snelle feedback en sturing van de bachelorproef.

Tot slot wil ik mijn ouders en vrienden bedanken voor de constante steun bij het uitvoeren van dit onderzoek.

Samenvatting

In dit onderzoek werd er gekeken naar de limitaties van handschriftherkenning. Hierbij werd de vraag gesteld of door wijze van handschriftherkenning de Christelijke Mutualiteit een meerwaarde zou zien bij het lezen van doktersvoorschriften. Met de evoluerende opkomst van Artificiële intelligentie, zijn zaken waar vroeger fysieke werkrachten voor ingezet werden, in een handgreep geautomatiseerd. Hierdoor wordt de nood aan extra kosten binnen een bedrijf geëlimineerd. Bovendien zijn mensen gekend om fouten te maken en kan dit een negatieve invloed op de workflow binnen het bedrijf veroorzaken.

De eerste taak bij dit onderzoek was de nood aan een literatuurstudie. Hierbij werd tot in detail beschreven wat de evolutie in het domein van handschriftherkenning doorheen de jaren ondergaan heeft en waar onderzoek in dit domein hedendaags toe gekomen is.

De literatuurstudie bracht ons tot enkele cruciale ondervindingen. Om mogelijks een conclusie te trekken uit deze onderzoeksvraag, zou in de eerste plaats een Machine Learning model nodig zijn, dat getraind is op de voorspelling van medische terminologie. Na enig opzoekwerk, bleek dat het verzamelen van de data nodig voor dit model, een probleem zou vormen. Doktersvoorschriften worden als een confidentiële vorm van data beschouwd binnen dit domein en worden niet zomaar vrijgegeven. Een op maat gemaakt model bleek dus geen mogelijkheid te zijn voor dit onderzoek.

Hierbij kwam het idee om gebruik te maken van Cloud Computing, men zou dan gebruik maken van een voorgetraind model dat aangeboden werd door een derde partij. Naar aanleiding van de stage, werd gekozen om gebruik te maken van de oplossingen die Microsoft te bieden heeft. Meer bepaald de Azure Computer Vision service. Met een instantie van deze service kan men eenvoudige API requests versturen die vervolgens een doeltreffend resultaat leverden door de Cloud in de achtergrond het werk te laten uitvoeren.

Om dit tenslotte op de proef te stellen, werden enkele test cases opgezet, waarbij verscheidene obstakels die zouden kunnen voorkomen, getest werden. Enerzijds werd hiervoor gebruik gemaakt van een test groep. Aan deze testgroep werden enkele doktersvoorschriften voorgesteld en werd gevraagd deze zo accuraat mogelijk te interpreteren. Vervolgens werd deze interpretatie vergeleken met de resulterende data van de API. Men kon hierdoor nagaan of artificiële intelligentie bepaalde woorden beter zou kunnen voorspellen dan mensen.

De accuraatheid hiervan werd deels ook op de proef gesteld door testen uit te voeren met een verlaagde resolutie en variërende positionering van personengegevens. Uit deze testen kon men concluderen dat doktersgeschrift wel degelijk met een relatief hoge accuraatheid en snelheid geëxtraheerd kon worden van voorschriften.

Dit proces zou op eenvoudige wijze in een productieomgeving ingezet kunnen worden, waardoor de Christelijke Mutualiteit een meerwaarde zou zien in processnelheid en accuraatheid. Aangezien dit onderzoek gebruik maakte van Engelse voorschriften, zou met verder onderzoek naar de extractie van Nederlandse voorschriften, een vollediger conclusie getrokken kunnen worden.

Inhoudsopgave

1	Inleiding	15
1.1	Probleemstelling	16
1.2	Onderzoeksvraag	16
1.3	Onderzoeksdoelstelling	16
1.4	Opzet van deze bachelorproef	17
2	Stand van zaken	19
2.1	Optical Character Recognition	19
2.2	Online vs. off-line recognition	20
2.3	Off-line recognition	21
2.3.1	Preprocessing	22
2.3.2	Letter segmentation	22
2.3.3	Feature extraction & Recognition	24

2.4	Machine Learning (ML)	25
2.4.1	Neurale netwerken	25
2.5	Cloud computing providers	26
2.6	Microsoft Azure	27
2.6.1	Software as a Service (SaaS)	27
2.6.2	Platform as a Service (PaaS)	27
2.6.3	Infrastructure as a Service (IaaS)	27
2.7	Azure Cognitive Services	28
2.8	Azure Computer Vision framework	29
2.8.1	Read API	29
2.8.2	OCR API	29
2.8.3	Recognize text API	30
2.9	Google Cloud Platform (GCP)	30
2.9.1	Cloud Vision API	32
2.10	Amazon Web Services (AWS)	32
2.10.1	Amazon Rekognition	33
2.11	Eerder uitgevoerd onderzoek	34
3	Methodologie	35
3.0.1	Gebruikte technologie	36
3.0.2	Test cases	36
3.0.3	Resultaten	36
4	Cognitive services computer vision	37
4.0.1	Opzet API	37

4.0.2	Het gebruiken van de API	41
4.0.3	Authenticatie	42
4.0.4	Opbouwen van een request	42
4.0.5	Versturen van een request	43
4.0.6	Response	44
4.0.7	Conclusie	48

5 Test cases 49

5.1 Test Case 1 50

5.1.1	Opzet	50
5.1.2	Doktersbriefje verlaagde kwaliteit resultaat	50

5.2 Test Case 2 52

5.2.1	Opzet	52
5.2.2	Doktersbriefje 1 resultaat	53
5.2.3	Doktersbriefje 2 resultaat	55
5.2.4	Doktersbriefje 3 resultaat	57
5.2.5	Doktersbriefje 4 resultaat	59

5.3 Test Case 3 61

5.3.1	Opzet	61
5.3.2	Doktersbriefje 5 resultaat	61

5.4 Resultaten 63

5.4.1	Test cases	63
5.4.2	Herkenningsgraad	64

5.5 Conclusie 65

6 Conclusie 67

A	Onderzoeksvoorstel	69
A.1	Introductie	69
A.2	State-of-the-art	70
A.3	Methodologie	70
A.4	Verwachte resultaten	70
A.5	Verwachte conclusies	70
A.6	Referenties	71
	Bibliografie	73

Lijst van figuren

2.1	x,y coördinaten worden getraceerd in functie van tijd. (Réjean Plamondon, 2000)	21
2.2	Het algemeen proces van text extractie via Off-line Recognition. (Gergely Timár en Rekeczky, 2003)	22
2.3	Woord segmentatie. (Gergely Timár en Rekeczky, 2003)	23
2.4	SaaS,IaaS,PaaS. (Wright, 2019)	28
2.5	Rotatie correctie van een woord via de OCR API. (Microsoft, 2019c)	30
2.6	Google datacenters. (Google, 2020b)	31
2.7	Text extractie request body. (Google, 2020c)	33
2.8	Text extractie request body. (Amazon, 2019)	34
4.1	Azure Dev/Test prijzen. (Microsoft, 2020a)	38
4.2	Computer Vision creatie pagina.	39
4.3	Pricing tier opties.	40
4.4	Doktersbriefje voorbeeld. Josen, 2017	45
5.1	Doktersbriefje voorbeeld statistiek 100% kwaliteit.	50
5.2	Doktersbriefje voorbeeld 60% kwaliteit.	51
5.3	Doktersbriefje voorbeeld statistiek 100% kwaliteit.	52

5.4	Doktersbriefje 1 extractie resultaat	53
5.5	Doktersbriefje 1. (Tacio, 2013)	54
5.6	Doktersbriefje 2 extractie resultaat	55
5.7	Doktersbriefje 2. (Tacio, 2013)	56
5.8	Doktersbriefje 3 extractie resultaat	57
5.9	Doktersbriefje 3. (Tacio, 2013)	58
5.10	Doktersbriefje 4 extractie resultaat	59
5.11	Doktersbriefje 4. (Tacio, 2013)	60
5.12	Doktersbriefje 5 extractie resultaat	61
5.13	Doktersbriefje 5. (Abrahams, 2008)	62
5.14	Statestiek voorspellingen.	63
5.15	Verloop van de herkenningsgraad per extractie.	64

Lijst van tabellen

1. Inleiding

Integratie van artificiële intelligentie in alledaagse workflows is een trend die meer en meer uitgeoefend wordt. Met de opkomst van artificiële intelligentie kan men eenvoudige taken die fysieke werkkrachten vereisen automatiseren. Hierdoor besparen bedrijven de tijd en het budget dat kruipt in het aanwerven en uitbetalen van personeel. Bovendien ligt het aantal transacties dat dergelijke software kan verwerken veel hoger dan hetgeen dat menselijke intuïtie kan bereiken. De taken die hierbij geautomatiseerd kunnen variëren van spraakherkenning systemen met realtime transcriptie tot handschriftherkenning op postbrieven om zo het juiste adres te achterhalen. Het herkennen van handgeschreven karakters was één van de eerste toepassingen die met de opkomst van AI verwezenlijkt kon worden. Toepassingen zoals deze kunnen vandaag de dag al volledige handgeschreven documenten omzetten naar machinale tekst met behulp van een gepast machine learning framework.

Hieruit komt de Christelijke Mutualiteit met de vraag of men deze technologie kan gebruiken om het typische doktersgeschrift dat men terugvindt op voorschriften te achterhalen. Het is een typische gezegde dat het handgeschrift van dokters praktisch onleesbaar is en dit blijkt zo uit de vraag naar een oplossing hiervoor.

Er zijn verscheidene spelers op de markt wanneer het komt om een performant AI-systeem op te bouwen. Grote spelers zoals Microsoft en Google hebben hun eigen cloud architecturen die gebruik maken van de enorme hoeveelheid data die zij verzameld hebben door de jaren heen. Aangezien de accuraatheid van een AI-systeem gepaard gaat met de hoeveelheid data die gebruikt wordt voor training spreekt het van zichzelf dat Microsoft en Google marktleiders zijn in dit veld. Er zijn echter naast deze twee oplossingen ook nog open source toepassingen die niet onderschat mogen worden.

1.1 Probleemstelling

In sommige gevallen kan het voorvallen dat wanneer men een voorschrift ingeeft bij de Christelijke Mutualiteit, het handgeschrift op dit voorschrift onleesbaar is. Dit vormt een probleem wanneer men medische kosten terug wil betalen. Het is duidelijk dat de tijd die de mensen gebruiken om dit handgeschrift te ontcijferen verloren gaat. Daarom zou een geautomatiseerd systeem dit probleem kunnen verminderen. Aangezien dit veld nog redelijk recent is, is het niet onrealistisch dat een foute voorspelling kan gebeuren. De meerderheid van deze systemen maken gebruik van recurrent neurale netwerken, wat betekent dat het systeem foute voorspellingen onthoudt en zichzelf aanpast naargelang specifieke kenmerken terugkeren. Hierdoor kan het systeem zich specialiseren in een domein waar bv. veel medische termen terugkomen.

1.2 Onderzoeksvraag

De vraag waarop de uitkomst van dit onderzoek uiteindelijk zal op antwoorden is of het wel degelijk mogelijk is om dokters handgeschrift te extraheren van doktersbriefjes. In welke zin varieert dokters handgeschrift van een ander en heeft dit invloed op de herkenning? De focus hierbij ligt op de efficiëntie van de gebruikte oplossing, in het geval van dit onderzoek zal de herkenningsfase in het praktische gedeelte gebruik maken van Azure Computer Vision. Dit omvat grotendeels de hoofdonderzoeksvraag, echter kan deze vraag nog opgedeeld worden in verschillende deelvragen.

- Welke factoren bepalen de accuraatheid?
- Worden medische termen even accuraat vertaald als normale tekst?
- Kan men deze oplossing hergebruiken voor andere doeleinden?
- In hoeverre maakt zelf ingebrachte data een verschil op het resultaat?

1.3 Onderzoeksdoelstelling

De uiteindelijke conclusie van dit onderzoek zal aantonen dat dergelijke software ontwikkeld kan worden om het probleem dat doktersgeschrift met zich meebrengt op te lossen. Wanneer de software levert wat er gevraagd wordt kan dit onderzoek als succesvol verklaard worden. Een accuraatheid van +- 74 % zou aantonen dat het systeem met een degelijke accuraatheid de extractie kan uitvoeren. In het domein van handschrift extractie wordt dit gezien als een hoog percentage. Percentages bij geprinte tekst liggen veel hoger aangezien een font eenzelfde lettertype/lettergrootte aanhoudt. Uiteraard is het ook van belang dat de CM een meerwaarde ziet in het gebruik van deze software. Hierbij zal men rechtstreeks doktersbriefjes kunnen inscannen en meteen ook een digitale versie verkrijgen van de inhoud.

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.

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

Handschriftherkenning, het woord zegt het zelf al. Het digitaliseren van handgeschreven tekst is een veelgebruikte technologie die al sinds de jaren 60 in de maak is en dagelijks talloze transacties automatiseert. Een handgeschreven adres op een brief digitaliseren of fysieke documenten converteren naar een pdf formaat elimineren de nood aan fysieke werkkrachten in verscheidene sectoren en verhogen de efficiëntie. De manier waarop een computertekst transformeert naar zijn digitale tegenligger is niet zo voor de hand liggend en gebruikt in de meeste gevallen een combinatie van technieken die over de jaren heen ontwikkeld werden.

Aangezien dit onderzoek de focus legt op tekst extractie, zullen de verscheidene technologieën die handschriftherkenning mogelijk maakt, besproken worden om zo een duidelijk overzicht te geven over wat er precies gebeurt. Na deze korte introductie zal het Azure computer Vision framework van Microsoft aangereikt worden. Dit is één van de meer modernere oplossingen die centraal staan bij het praktische gedeelte van dit onderzoek en die het herkennen van handgeschreven teksten vergemakkelijkt.

2.1 Optical Character Recognition

OCR (Optical Character Recognition) analyseert de vorm en de complexiteit van een bitsgewijs gemapt karakter en geeft deze vervolgens een waarde. Tegenwoordig kan de meeste OCR software zeer accurate resultaten weergeven als het op machinaal afgedrukte teksten neerkomt. Dit is zeker van toepassing wanneer er een duidelijke font met eenzelfde lengte tussen de karakters gebruikt wordt.

Wat echter een groter probleem is dat zich kan voordoen is variabele lengte van karakters, onbekende fonts en al zeker handgeschreven teksten. (Breithaupt, 2014)

OCR wordt vandaag de dag in een enorm aantal toepassingen gebruikt, zoals het transcriberen van brieven en in zelfrijdende auto's door het herkennen van snelheidslimieten op borden.

De technologische vordering in dit domein groeit dagelijks en zal in de toekomst enkel maar verbeteren. Echter, de accuraatheid van OCR is verre van perfect en vergt nog altijd een of andere vorm van menselijk toezicht. Op een pagina die 30 karakters telt zal zelfs een dergelijke OCR toepassing met 99 % accuraatheid, een gemiddelde van 30 fouten maken. Aangezien dit percentage van toepassing is op machinaal geschreven tekst, zal een OCR toepassing losgelaten op handgeschreven tekst een drastisch lager accuraatheidspercentage kennen. (Nagy e.a., 1999)

Technieken gebruikt bij het herkennen van handgeschreven woorden maken doorgaans gebruik van holistische of analytische strategieën voor de training en recognition fase. Holistische strategieën maken doorgaans gebruik van een top-down approach voor het herkennen van het volledige woord, hierdoor elimineert men problemen die zich voordoen bij de segmentatie fase. Bij deze strategie worden globale kenmerken van het volledige woord geëxtraheerd en worden vervolgens gebruikt bij herkenning van woorden in een kleine lexicografische omgeving. Analytische strategieën maken gebruik van een bottom-up approach. Expliciete segmentatie van woorden in karakters is nodig voor deze aanpak aangezien deze strategie geïsoleerde karakters herkent. Hierdoor is men niet gelimiteerd door een specifieke lexicografische omgeving maar kan men deze aparte karakters gebruiken in een willekeurige volgorde voor een ongelimiteerd aantal woorden. (Nafiz Arica, 2001)

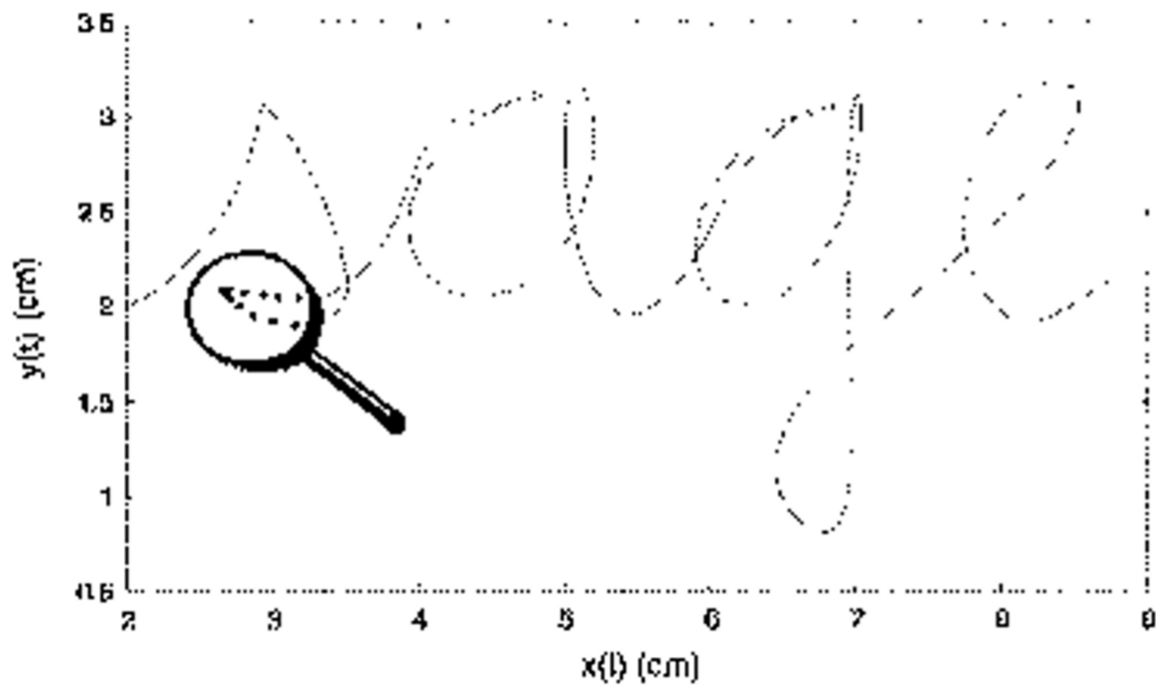
2.2 Online vs. off-line recognition

Wanneer men spreekt over het effectief herkennen van handgeschreven karakters zijn er 2 technieken die men kan onderscheiden, namelijk *Online Recognition* en *Off-line Recognition*. Online Recognition is een technologie die karakters herkent terwijl men het karakter aan het schrijven is in realtime op een digitaal oppervlak zoals bv. een tablet of een pda toestel. Input op deze toestellen gebeurt gewoonlijk door middel van een speciale pen die tijdelijke data verzamelt. (Gergely Timár en Rekeczky, 2003)

De automatische verwerking van karakters bekommt men door de data die de punt van deze pen genereert zoals bv. positie, acceleratie, snelheid. Door middel van de verwerking van deze variabelen kan men de x en y coördinaat van de pen plotten in functie van tijd en hierdoor een karakter traceren (zie Figuur 2.1). (Réjean Plamondon, 2000)

De meeste commercieel beschikbare handschriftherkenners lopen 2 tot 3 karakters achter op herkenning. Dit is in principe geen probleem aangezien de gemiddelde persoon tekst schrijft aan een **1.5-2.5** karakters per seconde. Aangezien realtime translatie van karakters

een nogal CPU intensieve taak is wordt door de huidige beschikbare technologieën de lat gelegd op een **5-10** karakters per seconde. Off-line Recognition wordt tegengesteld van Online Recognition, toegepast op reeds geschreven tekst documenten. (Tappert e.a., 1990)



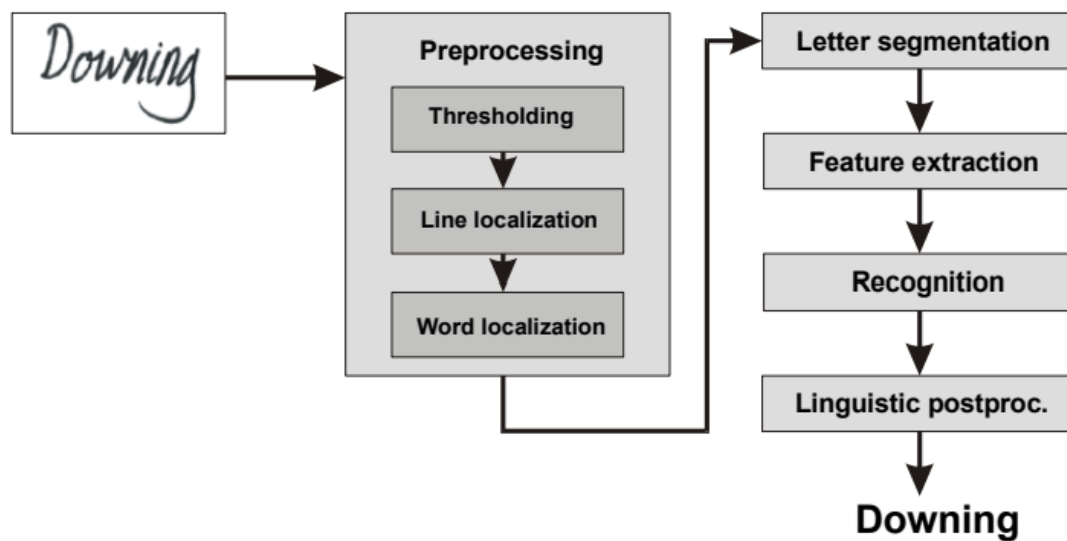
Figuur 2.1: x, y coördinaten worden getraceerd in functie van tijd. (Réjean Plamondon, 2000)

Aangezien dit onderzoek primair gebruik zal maken van Off-line Recognition zal dit verder in detail besproken worden.

2.3 Off-line recognition

Het produceren van een performant Off-line Recognition systeem is het doel van tal van onderzoekers in het domein van handschriftherkenning die dit doorheen de jaren proberen te perfectioneren. Handgeschreven teksten zijn een vorm van zelfexpressie, twee dezelfde woorden kunnen een totaal andere betekenis vormen in een verschillende context. Hierdoor kan de achterliggende betekenis van het woord drastisch veranderen. (Gergely Timár en Rekeczky, 2003)

Zoals reeds besproken maakt Off-line Recognition gebruik van een vooraf geschreven tekst om zo achteraf de tekst te extraheren. Volgens Gergely Timár en Rekeczky, 2003 kan dit proces dat doorheen verscheidene segmentatie systemen gebruikt wordt, in een generiek model gegoten worden.



Figuur 2.2: Het algemeen proces van text extractie via Off-line Recognition. (Gergely Timár en Rekeczky, 2003)

2.3.1 Preprocessing

Preprocessing omvat meerdere sub-technieken die toegepast worden vooraleer extractie plaatsvindt. Elk van deze sub-technieken speelt een belangrijke rol in het gehele proces. Zo wordt **thresholding** toegepast om de effectieve voorgrond-tekst te onderscheiden van de achtergrond. Deze techniek maakt gebruik van een grey-scale versie van het document zodat zwart duidelijk van wit onderscheiden kan worden, zo blijft de tekst geïsoleerd blijft. (Réjean Plamondon, 2000)

Ter voorbereiding van de karakter segmentatie fase wordt lijn en woordlokalisatie toegepast. Volgens Gergely Timár en Rekeczky, 2003 helpt deze lokalisatie met het analyseren van de woorden in zijn specifieke context, die op zijn beurt het herkenningpercentage zal verhogen. Het algoritme achter het lokaliseren van deze lijnen is vergelijkbaar met diegene gebruikt in Optical Character Recognition. Lijnen worden hierbij gelokaliseerd door het verwerken van de horizontale histogrammen voor het gehele document. Daardoor worden de positie en de graad waar de lijnen een lokaal minimum bereiken gekozen, om zo de plaatsen tussen de lijnen te lokaliseren.

Eens de lijnen gelokaliseerd zijn kan men vervolgens de woorden lokaliseren. Dit is in principe geen moeilijke taak meer aangezien de onderverdeling van elke lijn gekend is. Men kan door middel van het plaatsen van verticale histogrammen tussen de gekende lijnpunten, de woorden lokaliseren (zie Figuur 2.3). (Gergely Timár en Rekeczky, 2003)

2.3.2 Letter segmentation

Verdere segmentatie houdt in dat elke letter afgezonderd zal worden. Dit is echter niet zo voor de hand liggend als het lijkt. Het grootste obstakel dat men hier tegenkomt is de

And yet he is drawing on support from the anti clerical left. He is often accused of being a dictator. And yet he is today fighting a battle against militant dictatorship. His integrity. The greater part of the professional army is ranged against him. But there is no doubt that the concentrated strength of the French people is behind him because of a respect for his integrity which no French soldier or civilian has commanded in this century. Physically he is a sick man. His sight is failing him. He suffers from a cataract of both eyes. That is the principal reason why he never speaks with notes. He couldn't read them if he had them. He memorises all his speeches and when he was in England in nineteen fifty nine I congratulated him on his memory. He told me that it had always been good ever since he studied philosophy at the Jesuit college in Paris before going to St Cyr. The French Sandhurst. Spectacles could do something for his eyesight but he will not wear them because of a pardonable vanity which makes him feel that spectacles are unsuitable for a man fulfilling the role of soldier-father of the French people. As the far .

Figuur 2.3: Woord segmentatie. (Gergely Timár en Rekeczky, 2003)

variatie in afstand tussen letters. Niet iedereen schrijft letters op dezelfde manier en het kan soms voorvallen dat meerdere karakters overlappen. (Vijay Laxmi Sahu, 2012)

Om karakters op een systematische manier te isoleren wordt Internal Segmentation toegepast. Hierbij probeert men woorden te segmenteren in zijn individuele karakters zonder het gebruik van Feature Extraction. Deze eerste fase houdt geen rekening met eventuele overlappingsen en splits elke letter op in een individuele afbeelding. Vervolgens wordt elke afbeelding afzonderlijk geclassificeerd op basis van specifieke kenmerken a.d.h.v. een Hidden Markov Model (HMM). De cursieve oorsprong van handgeschrift maakt het moeilijk om woorden te segmenteren in karakter afbeeldingen. HMM's helpen om dit probleem op te lossen. (Nafiz Arica, 2001)

HMM's spelen al jaren een belangrijke rol in het domein van spraak- en gezichtsherkenning. Ondertussen worden deze modellen ook gebruikt bij patroondetectie en vormclassificatie. Deze stochastische modellen kunnen op een zeer efficiënte manier ruis en patroonvariatie in handgeschrift verwerken. Een nadeel van deze aanpak is dat men van de afbeeldingen als invoer verwacht dat deze perfect gesegmenteerd zijn van andere. Aangezien een perfecte aanpak een niet reëel voorkomen is, schat men volgens Bunke e.a., 1995 de verwachte herkenningspercentages tussen 74,5 % en 96.8 %.

2.3.3 Feature extraction & Recognition

Feature extraction is een belangrijk onderdeel van elk patroonherkenning systeem. Dit proces wordt uitgevoerd om de belangrijkste data te extraheren uit de ruwe dataset. Het uiteindelijke doel van dit proces is om een lijst van specifieke "features" van de karakters te bekomen. Met deze aanvullende data kan men vervolgens de herkenningspercentages maximaliseren. Zoals eerder aangereikt verschilt de variëteit in handgeschreven karakters drastisch en kan het extraheren van deze "features" een complexe taak zijn. (Vijay Laxmi Sahu, 2012)

Men kan karakter features onderscheiden in twee types, namelijk statistische en culturele. Een voorbeeld van statistische feature extraction is Zoning, waarbij men elke karakter afbeelding opdeelt in $N \times M$ zones. Hierdoor bekomt men bij bv. een afbeelding van 90 x 60 pixels een onderverdeling van 54 gelijke zones. Vervolgens wordt dit proces herhaald voor elke afbeelding van eenzelfde karakter. Men verkrijgt hierdoor een subset van karakter features waarvan het gemiddelde genomen kan worden. Dit gemiddelde vormt één enkele feature waarde die telkens in die zone geplaatst kan worden. (Vijay Laxmi Sahu, 2012)

Uiteindelijk worden deze features verstuurd naar de recognition engine. Deze retourneert vervolgens een lijst van woorden elk respectievelijk met zijn vertrouwensniveau. Dit niveau stelt voor hoe zeker het algoritme is van zijn voorspelling. (Gergely Timár en Rekeczky, 2003)

2.4 Machine Learning (ML)

Aangezien in de volgende hoofdstukken machinaal leren aan bod komt, zal hier een kleine inleiding op gegeven worden.

Machinaal leren (ML) is een concept waarbij men data en een algoritme samen brengt om een specifiek probleem op te lossen. Eens de data en het algoritme getraind zijn krijgt men een model als output. Dit model kan men opnieuw gebruiken met andere data. Het getrainde model geeft inzicht gebaseerd op de nieuwe data. Het bouwen van een ML systeem vergt enige kennis van data science en ML. (Microsoft, 2020h)

Volgens Salim Al-Habsib, 2019 kan men machinaal leren aanpakken op drie manieren: gesuperviseerd leren, ongesuperviseerd leren en semi-gesuperviseerd leren. Gesuperviseerd leren maakt gebruik van gelabelde trainingsdata. Het labelen van deze data neemt zowel tijd als geld in beslag, maar kan op zijn beurt weer geautomatiseerd worden door een neurale netwerk.

Semi-gesuperviseerd leren wordt volgens Zhu, 2005 beschreven als een speciale vorm van classificatie. Zoals reeds aangereikt, kan gesuperviseerd leren een ingewikkelde en dure manier zijn om een classificatie algoritme op te bouwen. Het labelen van data heeft in de meeste gevallen een vorm van menselijke controle nodig om een acceptabel resultaat te bekomen. Semi-gesuperviseerd leren lost dit probleem op door gebruik te maken van een grote hoeveelheid ongelabelde data, gepaard gaande met een deel gelabelde data. Een vergelijkbare toepassing hiervan, kan men terugvinden bij de Custom Vision service van Azure Cognitive Services. Ongesuperviseerd leren maakt gebruik van ongelabelde data. Het classificeren gebeurt hier door het clusteren van data met gelijkaardige patronen.

2.4.1 Neurale netwerken

Sanjay Kumar, 2016 beschrijft een artificieel neurale netwerk, als een krachtige modelleer tool, die complexe in- en uitgaande relaties kan weergeven en opvangen. De motivatie voor de ontwikkeling van neurale netwerken, stemt af van de nood aan een systeem dat complexe en intelligente taken kan uitvoeren, net zoals de taken uitgevoerd door het menselijk brein. Neurale netwerken komen op twee manieren overeen met het menselijk brein. Ze vergaren leerstof door het leren van vorige transacties en de vergaarde leerstof wordt opgeslagen in de vorm van synaptische gewichten.

Volgens Microsoft, 2020c worden de volgende types ANN als populairst beschouwd:

Feedforward neurale netwerk (FNN)

Een FNN is één van de meest eenvoudige soort artificiële neurale netwerken. In een FNN mag data maar één richting vloeien, namelijk van de input laag naar de output laag. Data wordt getransformeerd door het gebruik van meerdere lagen. Elke laag bestaat uit een aantal neuronen en is volledig geconnecteerd met al de neuronen in de voorgaande laag.

De laatste volledig geconnecteerde laag stelt de output voor en eveneens de voorspelde labels. (Microsoft, 2020c)

Recurrent neuraal netwerk (RNN)

Recursieve neurale netwerken worden in een aantal specifieke domeinen gebruikt. Dit type netwerk slaagt de output van een laag op en geeft deze vervolgens terug aan de input laag. Hierdoor bekomt men een vorm van recursie en kan men de uitkomst van een laag op voorhand voorspellen. Deze netwerken worden voornamelijk gebruikt voor complexe taken zoals handschriftherkenning en taalherkenning. (Microsoft, 2020c)

Convolutioneel neuraal netwerk (CNN)

Convolutionele neurale netwerken zijn één van de meest doeltreffende types van netwerken, gepaard met een unieke architectuur. Lagen worden in 3 dimensies opgebouwd: breedte, hoogte en diepte.

Neuronen in de ene laag connecteren partieel met al de neuronnen in de volgende laag. De finale output laag wordt gereduceerd tot een enkelvoudige vector van waarden die het vertrouwensniveau van de voorspelling weergeeft. Convolutionele neurale netwerken worden voornamelijk gebruikt bij video en afbeelding herkenning. (Microsoft, 2020c)

2.5 Cloud computing providers

Wanneer men dus uiteindelijk handschriftherkenning in een project wenst te integreren zijn er meerdere mogelijkheden. Men kan gebruik maken van een Cloud computing platform dat services zoals tekstherkenning en extractie aanbiedt, maar er is natuurlijk ook de optie om een open source framework zoals Tensorflow of Theano te gebruiken. Bij deze open source optie wordt er verwacht dat men het neurale netwerk/model volledig zelfstandig opbouwt en een eigen inbreng van trainingsdata voorziet. In sommige gevallen kan dit een voordeel zijn, wanneer er bv. een zeer specifiek model nodig is. Cloud gebaseerde oplossingen hebben hierbij het voordeel dat ze een model gebruiken dat getraind is met data verzameld door de provider. Verscheidene providers laten het toe om zelf nog extra trainingsdata in te brengen in het model, zoals bv. de Custom Vision service van Microsoft Azure.

In het domein van Cloud Computing zijn er dus enkele marktleiders die een forse voor-sprong hebben door middel van de hoeveelheid data die zij door de jaren heen hebben verzameld. Dit onderzoek maakt voornamelijk gebruik van de services die Microsoft Azure te bieden heeft, maar er zullen in de volgende hoofdstukken nog enkele andere alternatieven besproken worden die een gelijkaardig pakket aanbieden.

2.6 Microsoft Azure

Microsoft Azure is een uitgebreide set van cloud services die helpt bij het organiseren en het aanpakken van uitdagingen die zich voordoen op grootschalige niveaus. Deze uitdagingen worden typisch gerealiseerd door middel van cloud computing. Wanneer iemand niet beschikt over de nodige server of opslagcapaciteiten die zich kunnen voordoen bij sommige uitdagingen, dan kan men opteren voor cloud computing. Hierbij kan men bij providers zoals Azure, gebruik maken van de cloud om data op te slaan. Cloud computing platformen zoals Azure worden gerealiseerd door meerdere datacentrales op een globaal niveau. Hierdoor kan men wereldwijd op een betrouwbare manier toegang bieden tot verscheidene services die beschikbaarheid en betrouwbaarheid van data realiseren. (Microsoft, 2020d)

Volgens Vaibhav A Ghandi, 1999 kan men cloud computing opdelen in verschillende service modellen.

2.6.1 Software as a Service (SaaS)

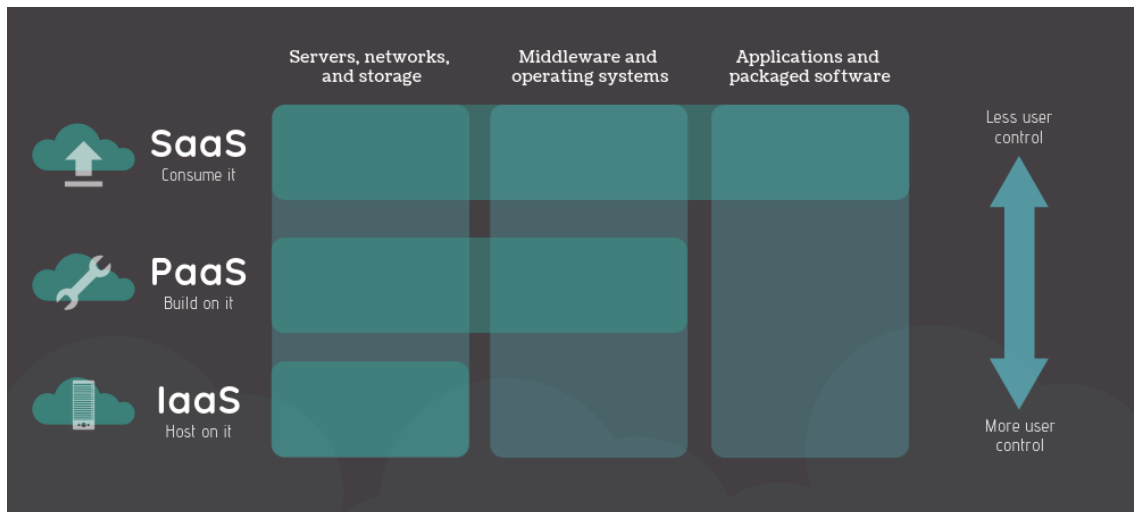
Software as a Service is een software gedistribueerd model waarin applicaties door een service provider gehost worden. De betalingsmethode voor SaaS volgt een subscriptie gebaseerd model. Hierdoor kunnen organisaties de focus leggen op hun core business zonder zich bezig te houden met infrastructuur management. (Manish Godse, 2009)

2.6.2 Platform as a Service (PaaS)

Platform as a Service biedt bedrijven een digitaal platform voor ontwikkeling en inzet van hun eigen applicaties en services. Hierdoor elimineert men de nood aan het onderhouden van servers, het programmeren van software en het opstellen van interne veiligheid protocollen. (Wright, 2019)

2.6.3 Infrastructure as a Service (IaaS)

Infrastructure as a service is de bundeling van hardware en geassocieerde software geleverd als een service. Deze hardware kan bestaan uit servers, opslag en of netwerk toepassingen. Meegeleverde software kan gaan van besturingssystemen, virtualisatie technologieën of file systemen. Bij IaaS is de provider enkel verantwoordelijk voor het operationeel houden van de data centers. Klanten zijn zelf verantwoordelijk voor het opstellen en het onderhouden van hun software services. (Bhardwaj e.a., 2010)



Figuur 2.4: SaaS,IaaS,PaaS. (Wright, 2019)

2.7 Azure Cognitive Services

Azure cognitive services geeft ontwikkelaars de kans om op een eenvoudige wijze artificiële intelligentie toe te voegen aan hun project zonder enige voorkennis te hebben van machinaal leren en/of data science. Toepassingen van cognitive services gaan van het analyseren en herkennen van objecten in afbeeldingen tot het herkennen van emoties op gezichten in realtime. Een cognitive service levert deels of de volledige componenten die nodig zijn in een toepassing die gebruik maakt van machinaal leren. Deze componenten kunnen bestaan uit bv. data, het algoritme of een vooraf getraind model. Een service zoals deze verwacht dat de gebruiker generieke info over de dataset meelevert. (Microsoft, 2020g)

Azure cognitive services kan geclassificeerd worden als een Software as a Service (SaaS), omvat een groep van services die onmiddellijk beschikbaar zijn voor deployment en kunnen opgedeeld worden in de volgende categorieën:

- **Decision:** Bouwt apps die aanbevelingen geven voor het maken van efficiënte en geïnformeerde beslissingen.
- **Language:** Zorgt ervoor dat men taal kan verwerken met voorgebouwde scripts, die emoties evalueren en leren wat een gebruiker juist wenst.
- **Search:** Voegt Bing Search API's toe aan apps en geven de mogelijkheid om miljarden webpagina's, afbeeldingen en video's te doorzoeken met één enkele API.
- **Speech:** Transformeert gesproken tekst naar digitale tekst en vertaalt één taal naar een ander samen met spraak verificatie en herkenning.
- **Vision:** Herkent, identificeert, indexeert en onderhoudt foto's, video's en digitale inkt toepassingen.

Sommige van deze services maken het mogelijk om zelf training data mee te geven om

vervolgens een model te trainen. Hierdoor kan men het model uitbreiden door gebruik te maken van de data die de service levert samen met de data die de gebruiker meelevert. Wanneer men zelf data meelevert, kan het voorvallen dat deze data nog op een specifieke manier gelabeld moet worden conform met de service. Door zelf data mee te geven kan je de service vormen naar de noden van de gebruiker. (Microsoft, 2020g)

2.8 Azure Computer Vision framework

Computer vision biedt een aantal services die afgedrukte of handgeschreven tekst detecteren en extraheren. Dit kan in verschillende situaties toegepast worden, zoals het maken van notities, inscannen van medische dossiers of bij beveiliging en bankieren. Voor het herkennen van tekst levert het Computer Vision Framework drie verschillende API's elk geoptimaliseerd voor een specifieke toepassing. (Microsoft, 2020b)

2.8.1 Read API

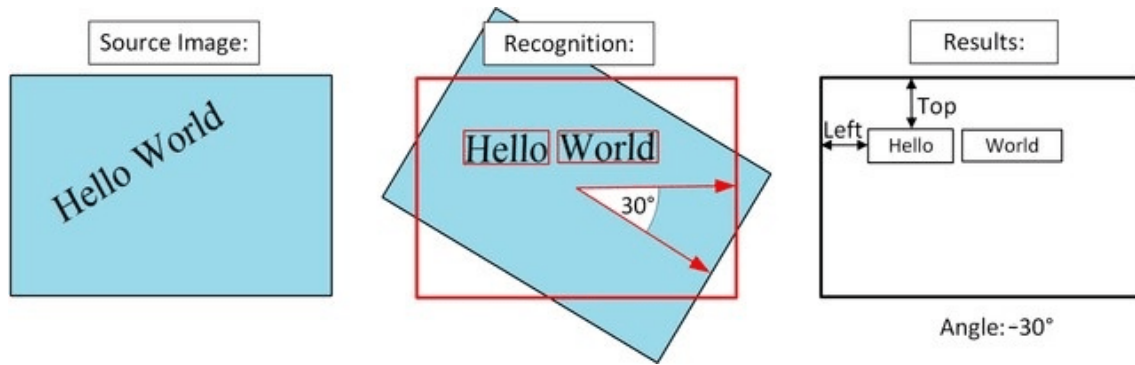
De Read API kan tekst in een afbeelding detecteren met behulp van de nieuwste herkeningsmodellen van computer vision, en zet geïdentificeerde tekst om in een karakterstroom. Deze API is voornamelijk geoptimaliseerd voor afbeeldingen die veel tekst hebben of met een bepaald aantal visuele ruis. Voor elke lijn tekst wordt eveneens beslist welk herkenningsmodel gebruikt zal worden. Hierdoor ondersteunt de API geprinte en handgeschreven tekst. De Read API voert de herkenning algoritmen asynchroon uit omdat men bij data intensieve documenten een langere tijd nodig heeft tot men een resultaat verkrijgt. (Microsoft, 2019c)

De leesoperatie houdt de originele lijn groeperingen van herkende woorden bij in zijn output. Elke lijn heeft zijn eigen coördinaten. Respectievelijk heeft elk woord in deze lijn ook zijn specifieke coördinaten. Wanneer een herkend woord een lage vertrouwens niveau heeft wordt dit percentage per woord meegeleverd. (Microsoft, 2019c)

2.8.2 OCR API

Men kan in grote lijnen de OCR API gelijkstellen aan de Read API, echter voert deze API zijn operaties synchroon uit en zijn deze niet geoptimaliseerd voor data intensieve documenten. Hiervoor maakt de OCR API gebruik van een ouder herkenningsmodel maar is deze ook compatibel met meerdere talen.

Wanneer nodig kan OCR de rotatie van specifiek herkende tekst aanpassen door middel van de afbeelding te draaien op zijn horizontale as (ziet figuur 2.5).



Figuur 2.5: Rotatie correctie van een woord via de OCR API.
(Microsoft, 2019c)

2.8.3 Recognize text API

De Recognize Tekst API is in zekere zin gelijkaardig aan dat van OCR, maar verwerkt zijn operaties asynchroon en maakt gebruik van geüpdatete herkenningmodellen. Deze API heeft echter enkele limitaties. De accuraatheid van deze operaties is grotendeels afhankelijk van de kwaliteit van de gebruikte afbeeldingen. De volgende factoren kunnen de accuraatheid beïnvloeden:

- Wazige afbeeldingen.
- Handgeschreven of cursieve tekst.
- Artistieke font keuzes.
- Kleine tekstgrote.
- Complexe achtergronden, schaduwen over tekst.

(Microsoft, 2019c)

2.9 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) biedt net zoals Microsoft Azure services voor opslag, analyse, big data en machine learning op een globaal niveau. Google realiseert dit door hun grootschalig netwerk van datacenters verspreid over 22 regio's. Door de spreiding van deze locaties kan men efficiënte en betrouwbare connectie met de cloud verzekeren.

Men kan de services die Google levert opdelen in enkele omvattende categorieën:

- **Compute engine:** De compute engine levert flexibele virtuele machines die draaien op de Google datacenters.
- **Storage & Databases:** Voorziet globale toegang tot verschillende opslag architecturen op enterprise niveau.
- **Machine learning:** Maakt het mogelijk voor developers om machine learning in een project te integreren en eveneens over te gaan naar een productieomgeving.

- **Big data:** Opslag en analyse mogelijkheden van grote hoeveelheden data, gepaard met data warehouse integratie en stream analytics.
- **Developer tools:** Voorziet pipelines en automatisatie van deployment in een productie omgeving.



Figuur 2.6: Google datacenters. (Google, 2020b)

Vergelijkbaar met Azure Cognitive Services heeft GCP een gelijkaardig machine learning platform, namelijk AI hub. De Google Cloud AI Hub is een gehoste opslagplaats waarbij men kant-en-klare AI toepassingen kan implementeren, en end-to-end AI pipelines kan opzetten. AI Hub biedt ook professionele deelmogelijkheden waarmee men AI toepassingen privé kan hosten. (Google, 2020e)

Naar aanleiding van dit onderzoek worden meerdere toepassingen besproken waarmee handschriftherkenning gerealiseerd kan worden. De oplossing die Google hier namelijk voor voorziet is Google Vision AI, meer specifiek de Cloud Vision API.

Google Vision AI biedt meerdere opties aan om computer visie modellen te integreren in applicaties of in websites. Men kan deze opties verder onderverdelen in drie categorieën:

- Auto ML Vision
- Cloud Vision API
- Cloud Vision product search

(Google, 2020a)

2.9.1 Cloud Vision API

De Google Cloud Vision API biedt een verzameling van technologieën die men kan gebruiken om de inhoud van afbeeldingen te analyseren. Hiervoor kan men gebruik maken van een zelf getraind ML model of één van de vele vooraf getrainde modellen die de API aanbiedt. Door gebruik te maken van AutoML Vision kan men naast de vooraf gedefinieerde labels, zelf specifieke labels toevoegen. Hierdoor kan men een domein specifiek model opbouwen. Toepassingen die gebruik maken van deze vooraf getrainde modellen omvatten, maar zijn niet gelimiteerd tot: afbeelding classificatie, object detectie, geprinte en handgeschreven tekstherkenning, gezichtsherkenning, product logo identificatie. (Google, 2020d)

De Vision API kan tekst in afbeeldingen detecteren en extraheren. Net zoals het Computer Vision framework van Microsoft, maakt Google ook gebruik van de OCR technologie om de achterliggende logica van karakter herkenning te realiseren. Er zijn namelijk 2 annotatie technieken die de Vision API ondersteunt:

- **TEXT_DETECTION:** Deze annotatie techniek wordt voornamelijk gebruikt voor het extraheren van geprinte tekst. Hiermee kan men bv. de tekst van een verkeersbord herkennen.
- **DOCUMENT_TEXT_DETECTION:** Wordt ook gebruikt voor tekst extractie maar is geoptimaliseerd voor documenten en handgeschreven teksten.

Beide technieken worden getriggerd bij aanvang van een REST request. De request body bestaat uit een JSON string. In deze string wordt een base64 gecodeerde afbeelding meegegeven samen met de gewenste annotatie techniek. Wanneer de request als succesvol wordt beschouwd door de server, dan retourneert deze een HTTP status code 200 OK.

De status code 200 geeft aan dat het verzoek geslaagd is, gepaard met deze status code wordt ook een response body geretourneerd in JSON formaat. In deze response wordt de gedetecteerde zin meegegeven, de geschatte taal en de coördinaten van waar de zin zich precies bevindt in de afbeelding. De Vision API voorziet ook de optie om rechtstreeks een afbeelding van Google Cloud of het web mee te geven bij de request. (Google, 2020c)

Op figuur 2.7 ziet men een voorbeeld van hoe een JSON request body opgebouwd wordt.

Beide types ondersteunen de integratie van één of meerdere taal hints. Deze hints specificeren de taal van de tekst die voorkomt in de meegegeven afbeelding. Dit kan in de sommige gevallen de snelheid van extractie verhogen. Meestal wanneer men dit niet specificeert bekomt men betere resultaten, aangezien de API zelf automatisch de taal detecteert. Het kan zelfs voorvallen dat een foutieve taal hint de herkenning kan hinderen.

2.10 Amazon Web Services (AWS)

Amazon Web Service (AWS) is de derde grote tegenhanger in het domein van Cloud Computing, naast Microsoft Azure en Google Cloud.

```
{
  "requests": [
    {
      "image": {
        "source": {
          "imageUri": "cloud-storage-image-uri"
        }
      },
      "features": [
        {
          "type": "DOCUMENT_TEXT_DETECTION"
        }
      ]
    }
  ]
}
```

Figuur 2.7: Text extractie request body. (Google, 2020c)

AWS maakt hun services wereldwijd beschikbaar op meerdere locaties. Deze locaties bestaan uit verschillende individuele regio's en zones. Elke regio is verder opgedeeld in geïsoleerde locaties gekend als "beschikbaarheid zones". AWS laat het toe om meerdere resources zoals instanties van AWS services of data, te deployen op meerdere locaties. Deze resources worden niet over verschillende regio's gedupliceerd, tenzij men dit expliciet specificeert. Elke regio is volledig zelfstandig en isoleert zich volledig van andere regio's. Hierdoor bekomt men een hoge graad van fout tolerantie en stabiliteit over het netwerk. (Amazon, 2020a)

AWS levert net zoals de concurrentie, tal van Cloud Computing services. Elk van deze service kan gebundeld worden in een cluster van services:

- **Analytics**
- **Application Integration**
- **Compute**
- **Database**
- **Internet of Things**
- **Machine Learning**
- **Containers**

2.10.1 Amazon Rekognition

Amazon Rekognition valt onder de AWS Machine Learning cluster maakt het mogelijk om op een eenvoudige wijze video en afbeelding analyse toe te voegen aan een applicatie. Wanneer men een afbeelding of een video meegeeft in een extractie request naar de Rekognition API, dan kan de API op een accurate manier objecten, mensen, tekst en activiteiten detecteren. Er is ook de mogelijkheid om net zoals bij Google Cloud Vision, custom labels mee te geven, om een model te maken dat gespecificeerd is aan de noden van de klant. (Amazon, 2020b)

De Rekognition API kan tekst in afbeeldingen en video's detecteren, en deze vervolgens converteren naar machinale tekst. Men kan een afbeelding aan een request meegeven in de vorm van een base64 gecodeerde string of een Amazon S3 object (zie figuur 2.8). Amazon S3 is een opslag service aangeboden door AWS, waar men key-value paren van objecten in kan opslagen. Deze objecten kan men rechtstreeks meegeven aan een API request en is niet gelimiteerd tot enkel afbeeldingen. (Amazon, 2019)

```
{
  "Image": {
    "S3Object": {
      "Bucket": "bucket",
      "Name": "inputtext.jpg"
    }
  }
}
```

Figuur 2.8: Text extractie request body. (Amazon, 2019)

Vervolgens retourneert de API een JSON response waarin de herkende woorden samen met hun vertrouwensniveau en positie weergegeven worden.

2.11 Eerder uitgevoerd onderzoek

Zoals aangereikt in de inleiding is dit onderzoek gebaseerd op een recent onderzoek uitgevoerd door Vectr. Consulting in samenwerking met de Christelijke Mutualiteit. Hierbij ontwikkelde zij een bot die automatisch doktersgeschrift op Belgische doodsaktes met een accuraatheid van 47 % correct voorspelde. Hiervoor maakten zij gebruik van het machine learning framework Tenserflow van Google, gepaard met een convolutioneel neuraal netwerk ontwikkeld door H. Scheidl. De data gebruikt in het trainen van dit neuraal netwerk werd verzameld gedurende 6 jaar. Finale resultaten van dit onderzoek tonen aan dat 47% van de testset correct gelabeld werd door het neuraal netwerk. Dit percentage is zeer gevoelig aan de kwaliteit van de preprocessing op de data en de segmentatie van woorden in karakters. Uit dit onderzoek kon men uiteindelijk concluderen dat het herkennen van doktersgeschrift zeker haalbaar is met een degelijk getraind neuraal netwerk.

3. Methodologie

Na een adequate hoeveelheid informatie vergaard te hebben bij het literatuuronderzoek, was het vervolgens mogelijk om het praktische deel van dit onderzoek uit te voeren. Uit de literatuurstudie kon vastgesteld worden dat handschriftherkenning op meerdere manieren gerealiseerd kon worden. Cloud computing voorziet een eenvoudige implementatie met behulp van vooraf getrainde modellen, maar komt met additionele transactiekosten. Een open source machine learning framework zoals Tensorflow kan in deze situatie een zeer effectieve oplossing bieden.

Men kan zoals gezien bij het onderzoek uitgevoerd door Vectr. Consulting, een zeer specifiek model opbouwen met zelf vergaarde data. Dit vormde echter een probleem bij dit onderzoek aangezien handgeschreven doktersbriefjes als een nogal confidentiële vorm van documenten worden gezien. Verder ligt de hoeveelheid data nodig om een toelaatbaar resultaat te bekomen hoog, dit zou enige tijd vergen om deze te bekomen.

Een reële aanpak was dus om gebruik te maken van één van deze Cloud computing oplossingen, alsook te observeren hoe een model getraind met een generieke dataset reageert op doktersgeschrift.

3.0.1 Gebruikte technologie

Aangezien ik momenteel een stage volg waar het gebruik van de Microsoft Azure Cloud centraal staat, leek het een interessante keuze om bij dit onderzoek gebruik te maken van de Computer Vision API van Azure Cognitive Services, meer bepaald de Read API. Hierbij kan zoals besproken in hoofdstuk 2.8 van het literatuuronderzoek, de API gebruikt worden om handgeschreven teksten te achterhalen van afbeeldingen. Daarom werd in een volgend hoofdstuk de onderdelen van deze API tot in detail besproken, samen met de software implementatie die hiermee communiceert.

3.0.2 Test cases

Om aan te tonen hoe de ontwikkelde software presteert, werden enkele test cases opgesteld. Bij elke test case werd een specifiek obstakel, dat zou kunnen voorvallen bij extractie, getest. Zo wordt bij de eerste test een afbeelding van een doktersvoorschrift gebruikt dat een lager dan normale resolutie heeft. Hierdoor worden de limitaties van de API op proef gesteld aangezien deze geen minimum resolutie definieert.

Een tweede meer fundamenteel gerichte test werd uitgevoerd om de gemiddelde accuraatheid van de API te testen. Hierbij werd er een extractie request afgevuurd op een aantal doktersbriefjes van hetzelfde type om zo inzicht te krijgen in hoe de API reageert op een normale workload. De teruggekregen data van deze request wordt achteraf samengebracht en weergegeven bij de resultaten.

Een derde en laatste test werd uitgevoerd om te testen of het uiterlijk van een voorschrift invloed heeft op het extractie resultaat. Verschillende voorschriften kunnen een vorm van hulplijnen hebben om de schrijver te helpen. Dit brengt mogelijks voor- en nadelen met zich mee en kan het beoogde resultaat beïnvloeden.

3.0.3 Resultaten

Uiteindelijk werd de bekomen data resulterend uit de test cases in een grafiek gegoten, waarbij de accuraatheid en payload van elke extractie request vergeleken werd. De uitkomst van deze resultaten bepaalde eveneens de bekomen conclusie van dit onderzoek en geeft wel degelijk aan of het mogelijk is om doktersgeschrift te extraheren van een voorschrift. Eveneens geeft dit ook een reëel inzicht over de meerwaarde die deze oplossing kan bieden op lange termijn.

4. Cognitive services computer vision

Vooraleer men de Computer Vision API kan gebruiken moet men in het bezit zijn van een lopende Microsoft Azure subscriptie. Deze subscriptie komt in een aantal verschillende opties en varieert drastisch in prijs en beschikbare noden. In figuur 4.1 worden de Azure subscriptie mogelijkheden duidelijk weergegeven met hun specifieke karakteristieken. Elke subscriptie type heeft zijn plaats en voordeel binnen de Azure cluster.

Aangezien studenten recht hebben op een gratis 30 dagen proefabonnement met een kredietlimiet van € 130, volstaat een Visual Studio subscriber abonnement voor dit onderzoek. Zoals weergegeven op figuur 4.1, zijn deze individuele abonnementen ook beschikbaar met een kredietscore van € 45 en € 85. Deze credits zijn gedurende de loop van een volledige maand beschikbaar. (Microsoft, 2020a)

Vervolgens is er de mogelijkheid om één van de vele beschikbare services aan te vragen die Microsoft Azure te bieden heeft. Credits worden automatisch afgenomen van het gebruikersaccount, in functie van de hoeveelheid calls uitgevoerd per service.

4.0.1 Opzet API

In de volgende sectie wordt de configuratie en opzet van een appservice, meer bepaald de Computer Vision API, in chronologische volgorde doorlopen en besproken. Hierbij wordt er inzicht gegeven in de mogelijke limitaties van de API en eveneens de verbonden transactiekosten. Na de opzet van deze API is het mogelijk om deze aan te roepen vanuit een software implementatie die in de volgende sectie besproken wordt.

	FOR INDIVIDUALS	FOR TEAMS (ENTERPRISE AGREEMENT CUSTOMERS)	FOR TEAMS (ALL OTHER CUSTOMERS)
Dev/Test pricing options	Monthly Azure credits for Visual Studio subscribers	Enterprise Dev/Test	Pay-As-You-Go Dev/Test
Number of instances allowed	One per Visual Studio subscriber	Unlimited	Unlimited
Monthly credit included*	€45, €85, or €130	None	None
Billing options	No credit card required Credit card billing for overages (invoicing available)	Always billed to the Enterprise Agreement Consumes prepaid Azure Monetary Commitment funds, if available	Credit card billing (invoicing available)
Billed usage	Overages are billed, after the monthly credit is consumed	All usage is billed	All usage is billed
Allowed to run Windows 10 Virtual Machines	✓	✓	✓
Service Level Agreement	Only for Azure DevOps and Application Insights	Only for Azure DevOps and Application Insights	Only for Azure DevOps and Application Insights
Licensing terms	Restricted to dev/test use only For use by active Visual Studio subscribers, and by end users providing feedback and performing acceptance tests	Restricted to dev/test use only For use by active Visual Studio subscribers, and by end users providing feedback and performing acceptance tests	Restricted to dev/test use only For use by active Visual Studio subscribers, and by end users providing feedback and performing acceptance tests

Figuur 4.1: Azure Dev/Test prijzen. (Microsoft, 2020a)

Een service kan men aanvragen via de Azure marketplace, hier wordt er een overzicht weergegeven van de verschillende categorieën van services. De Computer Vision API valt onder de categorie: ‘AI + Machine Learning’ en wordt beschouwd als een Cognitieve Service. (Microsoft, 2020b)

De eerste stap is het navigeren naar de marketplace om vervolgens Computer Vision te selecteren. Na deze selectie krijgt men de optie aangeboden om een instantie van deze service aan te maken, gepaard gaande met de configuratie mogelijkheden zichtbaar op figuur 4.2

The screenshot shows the 'Create' page for the Computer Vision service in the Microsoft Azure portal. The breadcrumb navigation at the top indicates the path: Home > Marketplace > Computer Vision > Create. The form contains the following fields:

- Name ***: Handschriftherkenning (with a green checkmark)
- Subscription ***: [Selected subscription]
- Location ***: (Europe) West Europe
- Pricing tier (View full pricing details) ***: S1 (10 Calls per second)
- Resource group ***: BP_2020 (with a 'Create new' link below it)

At the bottom of the form, there is a blue 'Create' button and a link to 'Automation options'.

Figuur 4.2: Computer Vision creatie pagina.

De creatie van een Computer Vision instantie verwacht 5 parameters:

- **Name**
- **Subscription**
- **Location**
- **Pricing tier**
- **Resource group**

De naam van een instantie mag enkel bestaan uit alfanumerieke karakters en dient nog niet te bestaan binnen de geselecteerde resource groep. Verder dient de lopende subscriptie van de ingelogde gebruiker geselecteerd te worden. Indien parameters zoals “pricing tier” buiten de scope van de subscriptie gaan krijgt de gebruiker een melding dat de gekozen subscriptie niet volstaat voor de gekozen noden en dat een upgrade nodig is. De gekozen locatie speelt een belangrijke rol als het aankomt op latentie en response time van de API. Hierdoor ligt de beste keuze voor dit onderzoek bij de West-Europese endpoint.

De pricing tier komt in twee verschillende opties:

- **F0 (20 Calls per minuut)**
- **F1 (10 Calls per seconde)**

Beide pricing tiers bieden standaard interactie met de API aan en kan men enkel onderscheiden in het aantal toegelaten transacties per tijdseenheid. De F0 tier volgt een vereenvoudigt transactieplan aangezien dit gratis verkregen kan worden. Bij dit plan wordt er een maandelijkse limiet ingesteld van 5000 transacties samen met een call limiet van 20/minuut. Voor de meeste gebruikers die een niet commerciële actie verrichten, volstaat dit plan. Echter biedt de service ook de S1 tier aan die meer gericht is naar gebruik op Enterprise niveau. Er wordt hierbij voor het Read onderdeel van Computer Vision, een vaste prijs opgesteld van €2.109 per 1,000 transacties. (Microsoft, 2020e)

F0 Free		S1 Standard	
20	Calls per minute	10	Calls per second
5K	Calls per month		
	AnalyzeImage		AnalyzeImage
	TagImage		TagImage
	DescribeImage		DescribeImage
	OCR		OCR
	GetThumbnail		GetThumbnail
	0.00		0.84
	USD PER CALL (ESTIMATED)		STARTING EUR/1000 CALLS (ESTIMATED)

Figuur 4.3: Pricing tier opties.

Omdat dit een observatief onderzoek is, volstaat de F0 tier voor het uitvoeren van de test cases en het verzamelen van genoeg data om een conclusie te kunnen trekken. Verdere implementatie van de ontwikkelde software in een productie omgeving kan een hogere tier verwachten. Wanneer men opmerkt dat de F0 tier niet volstaat voor de implementatie, is

het mogelijk deze achteraf te upgraden in het API-overzicht (zie fig 4.3).

Na creatie van de service krijgt men een bevestiging mail toegestuurd met de verwachte uitrol tijd. Dit zou niet langer mogen duren dan 5 minuten. Eens beschikbaar, kan men de onderhoudspagina van de service terugvinden in het Azure dashboard. Hier zijn enkele belangrijke parameters van belang om de nodige communicatie te voorzien met software die calls naar de API verstuurd. Dit zijn namelijk de volgende parameters:

- **API Key**
- **API Endpoint**

Elke request uitgevoerd naar de API verwacht een geldige API key en zorgt ervoor dat deze request geautoriseerd wordt (Microsoft, 2019a). Door deze key weet de API ook wat de limieten zijn van het gekozen plan. De API Endpoint is dan de uiteindelijke URL waar de request naar verstuurd wordt. De endpoint bevindt zich dan ook eveneens op de locatie gekozen bij configuratie van de service. De verdere implementatie en het gebruik van deze parameters wordt verder besproken in de volgende sectie.

Het is echter ook nog op te merken dat de huidige versie van de API (Version 3) enkel de extractie van Engels en Spaanse handgeschreven tekst ondersteunt. Er is in dit onderzoek bewust gekozen om de extractie uit te voeren op Engels geschreven doktersbriefjes. Zoals besproken in hoofdstuk 3, is het verzamelen van handgeschreven teksten in het Nederlands een zeer moeilijke taak. Dit vergt in de meeste gevallen de samenwerking met een tweede partij die willend is om deze data vrij te geven. Daarom is er voor dit onderzoek zoals besproken in hoofdstuk 3, gekozen om gebruik te maken van een Cloud computing model dat getraind is met een generieke dataset. Dit is in zekere zin ook een interessante omgeving waar testen uitgevoerd kunnen worden. Deze testen kunnen aantonen of een generieke dataset even goed presteert bij de extractie van medische termen. In toekomstige versies en naarmate de evolutie van de API, zullen vermoedelijk meerdere talen ondersteunt worden voor extractie, waaronder Nederlands.

De OCR API ondersteund momenteel wel al de extractie van geprinte tekst in het Nederlands samen met 25 andere talen (Microsoft, 2019b).

4.0.2 Het gebruiken van de API

De volgende sample code van deze implementatie is beschikbaar op github, en werd grotendeels verkregen door Microsoft, 2020f mits enkele aanpassingen voor string parsing. Vervolgens zullen de belangrijkste onderdelen van deze code besproken worden samen met een test request + resultaat. De code werd opgebouwd in C# .NET en maakt gebruik van het Newtonsoft.Json framework voor eenvoudige interactie en manipulatie van Json bestanden.

4.0.3 Authenticatie

Zoals besproken in de voorgaande sectie, zal elke request verstuurd naar de API, geautoriseerd moeten worden. Dit houdt in dat de API Key meegegeven wordt in de hoofding van de request. Echter is het van enorm belang dat men beide Api Key en Endpoint nooit in code mag terugvinden en dit al zeker niet in een github of productie omgeving. Gebruikers die geen eigenaar zijn van de API kunnen met deze key ongezien requests versturen en data ontvangen. Men kan deze key op enkele manieren verbergen in het project, ook de mogelijkheid is er om gebruik te maken van Azure Keyvault. Hierbij wordt een gezamenlijk overzicht weergegeven van al de beschikbare keys van het gebruikers account. Deze zijn achteraf raadpleegbaar met een request call naar de keyvault.

In deze toepassing werd er gekozen om de parameters *subscriptionKey* en *endpoint*, temporeel op te slagen in de omgevingsvariabelen van het project. Hierdoor worden deze enkel geïnjecteerd bij opstart en zijn ze deels verborgen.

```
static string subscriptionKey =  
    Environment.GetEnvironmentVariable  
        ("COMPUTER_VISION_SUBSCRIPTION_KEY");  
  
static string endpoint =  
    Environment.GetEnvironmentVariable  
        ("COMPUTER_VISION_ENDPOINT");  
  
static string uriBase =  
    endpoint + "/vision/v3.0-preview/read/analyze";
```

Fragment 1.1: Parameters.

De *uriBase* is een samengestelde parameter bestaande uit de gegenereerde API endpoint van de service, samen met het type request dat men wenst uit te voeren. In dit geval zal de *read* operatie uitgevoerd worden.

4.0.4 Opbouwen van een request

Een read request wordt vervolgens opgebouwd door het initialiseren van een *HttpClient*. Deze client voegt nodige parameters toe aan de request body en header, vooraleer men deze afvuurt. In fragment 1.2 wordt de API key toegevoegd aan de header. Dit is zoals besproken in sectie 4.0.3 nodig om de request te autoriseren. Verder is het van belang dat de taal van de meegegeven afbeelding gespecificeerd wordt. Momenteel is dit Engels in het formaat "en_US", maar in latere stages van de API zal dit evolueren naar Nederlands.

```
HttpClient client = new HttpClient();

// Request headers.
client.DefaultRequestHeaders.Add(
    "Ocp-Apim-Subscription-Key", subscriptionKey);

var builder = new UriBuilder(uriBase);
builder.Port = -1;
var query = HttpUtility.ParseQueryString(builder.Query);
query["language"] = language;
builder.Query = query.ToString();
string url = builder.ToString();
```

Fragment 1.2: HttpClient.

Uiteraard is het van belang dat een afbeelding meegegeven wordt aan de request. Men kan echter niet zomaar een “JPEG” of “PNG” meegeven aan de client. Deze dient eerst omgezet te worden naar een binaire voorstelling van data, in dit geval een array van bytes. Vervolgens wordt deze array van bytes meegegeven aan een instantie van “ByteArrayContent”. Deze instantie combineert de binaire data samen met het “MediaType” van de data. Hierdoor weet de API het type van de meegeleverde data in de body.

4.0.5 Versturen van een request

Na opzet van de request header en body, kan men de “client” gebruiken om een post request te versturen m.b.v. de methode *PostAsync()* (zie fragment 1.3). Deze methode ontvangt twee parameters, namelijk de “url” en de samengestelde “ByteArrayContent”. Merk op dat deze methode van het type *Async* is. Dit betekent dat men nooit zeker is wanneer men het beoogde resultaat terugkrijgt van de API en zal dit aldus asynchroon toekomen.

Om onnodige fouten te vermijden wordt een “await” operator vóór de functie toegevoegd. Deze operator zorgt er voor dat verdere code afhankelijk is van het resultaat van deze response en blokkeert hierbij dus andere threads tót een resultaat verkregen is.

Het geretourneerde resultaat bevat echter niet de geëxtraheerde tekst, maar één van twee status codes. Bij aanvang van een POST request, zal de API een read operatie uitvoeren op de meegeleverde afbeelding en vervolgens het bekomen resultaat bijhouden. Dit resultaat dient achteraf opgehaald te worden door wijze van een GET request naar de API. In fragment 1.3 wordt een check uitgevoerd op de geretourneerde statuscode. Indien deze van het type “succes” is, bevindt de URI locatie van de geëxtraheerde tekst zich in de header van de response.

```

byte[] byteData = GetImageAsByteArray(imageFilePath);

using (ByteArrayContent content = new ByteArrayContent(byteData)) {
    content.Headers.ContentType =
        new MediaTypeHeaderValue("application/octet-stream");
    response = await client.PostAsync(url, content);
}
if (response.IsSuccessStatusCode)
    operationLocation =
        response.Headers.GetValues("Operation-Location")
            .FirstOrDefault();
else {
    string errorString = await response.Content.ReadAsStringAsync();
    Console.WriteLine("\n\nResponse:\n{0}\n",
        JToken.Parse(errorString).ToString());
    return;
}

```

Fragment 1.3: Post request.

Eens de URI gekend is, wordt deze gebruikt om een GET request op te bouwen. Hierin bevindt zich het uiteindelijke resultaat van de tekst-extractie in JSON formaat. Zoals beschreven in de samplecode van Microsoft, 2020f is de tijd van teruggave afhankelijk van de hoeveelheid te verwerken tekst. Daarom wordt aangetoond in fragment 1.4, dat om de 1000 ms een check gedaan wordt op de beschikbaarheid van de data.

```

byte[] byteData = GetImageAsByteArray(imageFilePath);

using (ByteArrayContent content = new ByteArrayContent(byteData)) {
    content.Headers.ContentType =
        new MediaTypeHeaderValue("application/octet-stream");
    response = await client.PostAsync(url, content);
}
if (response.IsSuccessStatusCode)
    operationLocation =
        response.Headers.GetValues("Operation-Location")
            .FirstOrDefault();
else {
    string errorString = await response.Content.ReadAsStringAsync();
    Console.WriteLine("\n\nResponse:\n{0}\n",
        JToken.Parse(errorString).ToString());
    return;
}

```

Fragment 1.4: Get request.

4.0.6 Response

Dit concludeert in grote lijnen de werking van de API, in de volgende sectie wordt een eenvoudige test request opgezet met een beter inzicht op wat voor data de API precies retourneert en hoe deze gebruikt kan worden in een probleem dat zich kan voordoen in de echte wereld. In eerste instantie zal dit een JSON-string zijn met een aantal zeer specifieke eigenschappen van elk gedetecteerd woord.

Om dusdanig de software te gaan testen, wordt op figuur 4.4 gebruik gemaakt van een redelijk leesbaar doktersbriefje. Merk op dat deze voorschriften drastisch kunnen variëren en dat deze verschillende vormen kunnen aannemen.

Reynaldo O. Joson, MD, MHA, MHPed, MSc Surg

MANILA DOCTORS HOSPITAL

Suite 301, DSMT MAC, U.N. Avenue Ermita, Manila

522-47-13; 558-08-88 Loc 4100 ; 0918.804.0304

rojasonmedclinic@gmail.com ; rjoson2001@yahoo.com

<https://sites.google.com/site/rojasonmedicalclinic>

LINA CRUZ

Date: 1-24 20 17

Rx

^{tr}
Tramadol + Paracetamol tab

— #20

Sig: 1 tab every 6 hours

PTC: 49049095



, M.D.

Lic. # 44609

Figuur 4.4: Doktersbriefje voorbeeld. Joson, 2017

Een eerste uitdaging is dus om na te gaan of medische termen, meer bepaald medicatie zoals “Tramadol” en “Paracetamol” gezien in figuur 4.4, gedetecteerd worden. Dit zou enerzijds al een eerste stap zijn om een conclusie te trekken bij dit onderzoek. De volledige handgeschreven tekst die als persoon waargenomen wordt zou dus het volgende zijn:

- LINA CRUZ
- PX
- Tramadol + Paracetamol tab
- #20
- Seg. 1 tab every 6 hours
- PTR : 49049095

Vervolgens laten we de software los op figuur 4.4 en bekijken de uitkomst. Het ruwe JSON-resultaat is terug te vinden op github maar zal voor leesbaarheid geparsed worden naar een duidelijker resultaat. Microsoft, 2017 beschrijft de parameters in deze geretourneerde JSON als volgt:

- **Status:** De teruggekeerde status van de request, mogelijks “succeeded”, “notStarted”, “failed” en “running”. Vanaf deze status op “succeeded” komt te staan, zal het JSON-bestand de herkende tekst bevatten.
- **Lines:** Een lijst van herkende zinnen. Het maximum aantal lijnen dat op één pagina gedetecteerd kan worden is 300. Hierbij worden deze lijnen gesorteerd op de volgorde waarin deze gedetecteerd werden (van linksboven naar rechtsonder). De lijnen stellen dan in de meeste gevallen ook een volledige zin voor.
- **Text:** Een volledig herkende zin in één lijn, deze wordt verder opgesplitst in een lijst van woorden.
- **Words:** Een verzameling van woorden in een herkende zin.
- **BoundingBox:** Een rechthoek dat de positie bevat van de herkende zin of het herkende woord, gespecificeerd als een lijst bestaande uit 8 nummers. Deze nummers stellen de coördinaten van het herkende woord voor, relatief aan de linkerbovenhoek van de afbeelding. Dit zijn dus 4 paren van x,y coördinaten waardoor een rechthoek gevormd wordt (zie fragment 1.5).
- **Confidence:** Het vertrouwensniveau van elk gedetecteerd woord. Zoals beschreven in hoofdstuk 2, zou dit voor handgeschreven teksten lager moeten liggen dan getypte tekst.

Om in het volgende hoofdstuk collectief de resultaten op een eenvoudige wijze te vergelijken en samen te vatten, wordt dus gebruik gemaakt van een parsing methode. Deze methode genereert een overzicht dat enkel de zin, de woorden en het vertrouwensniveau weergeeft, wat uiteraard ook de belangrijkste attributen zijn.

In fragment 1.5 wordt een deel van het ruwe geretourneerde resultaat weergegeven. Merk op dat voor de leesbaarheid, de “boundingBox” parameter van elk specifiek woord weggelaten is, aangezien men een algemene “boundingBox” van de volledige zin ter beschikking heeft.

```
{
  "boundingBox": [
    65,
    312,
    529,
    287,
    532,
    332,
    67,
    353
  ],
  "text": "tramadol+_Paracetamol_tab",
  "words": [
    {
      "text": "tramadol",
      "confidence": 0.328
    },
    {
      "text": "+",
      "confidence": 0.822
    },
    {
      "text": "Paracetamol",
      "confidence": 0.463
    },
    {
      "text": "tab",
      "confidence": 0.655
    }
  ]
}
```

Fragment 1.5: Ruw resultaat.

```

LINA CRUZ

'LINA' detected with a confidence level of [0.909]
'CRUZ' detected with a confidence level of [0.762]

tramadol + Paracetamol tab

'tramadol' detected with a confidence level of [0.252]
'+' detected with a confidence level of [0.82]
'Paracetamol' detected with a confidence level of [0.482]
'tab' detected with a confidence level of [0.721]

# 20

'#' detected with a confidence level of [0.851]
'20' detected with a confidence level of [0.958]

Sig. I fab every 6 hours

'Sig.' detected with a confidence level of [0.487]
'I' detected with a confidence level of [0.487]
'fab' detected with a confidence level of [0.57]
'every' detected with a confidence level of [0.252]
'6' detected with a confidence level of [0.551]
'hours' detected with a confidence level of [0.42]

PTR : 49049095

'PTR' detected with a confidence level of [0.391]
': ' detected with a confidence level of [0.671]
'49049095' detected with a confidence level of [0.481]

```

Fragment 1.6: Parsed resultaat.

Men kan duidelijk zien dat mits de lage vertrouwenswaarde, de zin “Tramadol + Paracetamol tab” volledig juist geëxtraheerd is. Dit resultaat stelt wel degelijk voor dat medische termen opgenomen zijn bij het trainen van het extractie model. Verder worden de rest van de handgeschreven zinnen vlekkeloos overgenomen. Dit toont enerzijds aan hoe accuraat dit model is mits dit nog een zeer recente technologie is.

Als de gemiddelde accuraatheid genomen wordt van elk woord, denkt de API de tekst met een accuraatheid van 59% voorspeld te hebben. In principe ligt dit percentage relatief laag aangezien men kan vaststellen dat de meerderheid van de woorden volledig juist voorspeld werden. Verder, was de responsetijd van de API was in dit geval +- 8.673 seconden.

4.0.7 Conclusie

Uit dit hoofdstuk kunnen we concluderen dat het wel degelijk mogelijk is om handschrift te extraheren van een afbeelding. Dit enerzijds rekening houdend met medische termen en opschriften. Wat nu momenteel echter in vraag gesteld wordt, is of dit resultaat aanhoudend is. Hoe reageert de API tegenover een ander type doktersbriefje, schrijfstijl of kleur? Het is onmogelijk om duizenden testen op te zetten om elke specifieke karakteristiek dat in een doktersbriefje kan voorkomen, te testen.

In het volgende hoofdstuk worden enkele van deze algemeen gerichte vragen in een test gegoten.

5. Test cases

Uit het vorige hoofdstuk kon men vaststellen dat de Computer Vision API wel degelijk een bruikbaar resultaat levert. Dit resultaat kan verder gemanipuleerd worden voor verscheidene use cases die zich kunnen voordoen in eender welk domein. In figuur 3.4 van hoofdstuk 4 werd een semi leesbaar doktersbriefje gebruikt. In principe was dit een perfect voorbeeld om enkel de functionaliteit van de API te testen. Echter is het van belang om na te gaan waar de API de lijn trekt.

Om die redenen werden er in het volgende hoofdstuk drie test cases opgesteld. Elke van deze test cases bestaat uit één of meerdere doktersbriefjes. Hierbij werden deze voorgesteld aan aantal personen om na te gaan hoe zij deze tekst interpreteerden. Achteraf werd van elk geïnterpreteerd woord de populairste interpretatie genomen en vergeleken met het resultaat van de API.

Hierdoor is het mogelijk om een rechtstreekse vergelijking uit te voeren van hoe mensen een onleesbare tekst interpreteren vergeleken met een AI-toepassing.

Eveneens wordt er per extractie een tabel opgesteld met het geëxtraheerde woord, de interpretatie en de herkenninggraad.

5.1 Test Case 1

5.1.1 Opzet

De eerste test case zal nagaan wat het effect van een verlaagde resolutie is op de extractie. Om dit effectief te gaan testen zal men gebruik maken van hetzelfde doktersbriefje in figuur 4.4

Echter werd een kwaliteitscompressie op deze afbeelding uitgevoerd waarbij deze nog maar 60 % van de originele kwaliteit bevat (zie fig 5.2). Vooraleer een vergelijking kan plaatsvinden, werd een statistische tabel van de originele extractie request opgesteld in figuur 5.1. Men kan duidelijk vaststellen dat de meerderheid van de woorden correct voorspeld werden.

Effectief woord	Voorspelling	Herkenningsgraad	Voorspelling
LINA	LINA	0.909	Juist
CRUZ	CRUZ	0.762	Juist
1	1	0.7	Juist
24	24	0.945	Juist
17	17	0.252	Juist
tramadol	tramadol	0.328	Juist
+	+	0.822	Juist
Paracetamol	Paracetamol	0.463	Juist
tab	tab	0.655	Juist
#20	#20	0.818	Juist
Sig.	Sig.	0.487	Juist
1	l	0.487	Juist
tab	fab	0.57	Fout
every	every	0.252	Fout
6	6	0.57	Juist
hours	hours	0.324	Juist
PTR	PTR	0.391	Juist
:	:	0.671	Juist
49049095	49049095	0.481	Juist

Figuur 5.1: Doktersbriefje voorbeeld statistiek 100% kwaliteit.

5.1.2 Doktersbriefje verlaagde kwaliteit resultaat

Wanneer naar figuur 5.2 gekeken wordt, ziet men op het eerste zicht visueel geen grote verschillen in kwaliteit, echter kan dit wel een invloed hebben op de extractie kwaliteit. Meer bepaald de virtuele “ruis”. Deze ruis kan karakters vervormen en in extreme gevallen deze zelfs onherkenbaar maken. Vervolgens werd er een extractie request afgevuurd op figuur 5.2 en werd de verkregen data hiervan in tabel 5.3 weergegeven.

Reynaldo O. Joson, MD, MHA, MHPed, MSc Surg
MANILA DOCTORS HOSPITAL
Suite 301, DSMT MAC, U.N. Avenue Ermita, Manila
522-47-13; 558-08-88 Loc 4100 ; 0918.804.0304
rojasonmedclinic@gmail.com ; rjoson2001@yahoo.com
<https://sites.google.com/site/rojasonmedicalclinic>


LINA CRUZ Date: 1-24 20 17

Rx

^{tr}
Tramadol + Paracetamol tab
— #20

Sig. 1 tab every 6 hours

PTC: 49049095


_____, M.D.
Lic. # 44609

Figuur 5.2: Doktersbriefje voorbeeld 60% kwaliteit.

Effectief woord	Voorspelling	Herkenningsgraad	Voorspelling
LINA	LINA	0.939	Juist
CRUZ	CRUZ	0.762	Juist
1	1	0.694	Juist
24	24	0.959	Juist
17	17	0.252	Juist
tramadol	tramadol	0.286	Juist
+	+	0.836	Juist
Paracetamol	Paracetamol	0.471	Juist
tab	tab	0.625	Juist
#20	#20	0.799	Juist
Sig.	Sig.	0.276	Juist
1	l	0.41	Juist
tab	fab	0.57	Fout
every	every	0.252	Fout
6	6	0.436	Juist
hours	hours	0.415	Juist
PTR	PTK	0.603	Fout
:	:	0.697	Juist
49049095	49049095	0.442	Juist

Figuur 5.3: Doktersbriefje voorbeeld statistiek 100% kwaliteit.

Wanneer men beide tabellen vergelijkt zijn er enkele verschillen die meteen naar voor komen. Zoals verwacht is de algemene herkenninggraad gedaald. De meerderheid van de woorden werden nog steeds juist voorspeld echter met een lagere herkenninggraad. Er werd één woord vergeleken met figuur 4.4 verkeerd voorspeld, namelijk "PTR". Dit stelt voor dat een lagere resolutie wel degelijk invloed kan hebben op de extractie. Narmate de verlaging in kwaliteit zullen meer en meer woorden verkeerd voorspeld worden.

5.2 Test Case 2

5.2.1 Opzet

De tweede test case is een meer generiek gerichte test, hierbij wordt gekeken naar de algemene accuraatheid van de teruggegeven data en het aantal juiste voorspellingen. Hier werd een andere aanpak genomen tegenover test case 1. De juistheid wordt hierbij bepaald door de voorspelling te vergelijken met een interpretatie. Hierbij werden vier doktersbriefjes voorgesteld aan een groep mensen. Elke persoon in deze groep probeerde de handgeschreven tekst in deze doktersbriefjes zo accuraat mogelijk te interpreteren. Achteraf werd de gemiddeld populairste interpretatie van elk woord genomen en in een tabel gegoten. Door een vergelijking uit te voeren van het voorspelde met de interpretatie, kan men nagaan of de API handgeschreven woorden op dezelfde manier interpreteert als mensen.

5.2.2 Doktersbriefje 1 resultaat

Zoals men kan zien op figuur 5.4 zijn er in totaal 23 woorden/karakters geëxtraheerd. Hierbij zijn 14 van deze 25 correct voorspeld naargelang de interpretatie. Merk op dat men de interpretatie niet als volledig betrouwbaar mag beschouwen. Een persoon kan zonder medische achtergrond of kennis een woord op een volledig andere manier interpreteren dan iemand die ervaring heeft in dit domein. Een voorbeeld hiervan zijn de woorden “amoxicillin” en “Himox”, de populairste interpretatie hiervan werd gezien als “omoxirillin” en “Hinlox”.

Echter kan men deze voorspelling niet zoals in andere gevallen fout beschouwen, want “amoxicillin” en “Himox” zijn medisch correcte termen. Men kan dus wel degelijk vaststellen dat medische termen in hun geheel correct voorspeld kunnen worden, echter is hier wel nog een manuele controle voor nodig. In conclusie kan men een gemiddelde herkenningsgraad van 0.45087 per woord vinden, met 60% van deze woorden correct voorspeld.

Interpretatie	Voorspelling	Herkenningsgraad	Voorspelling
armando	armando	0.285	Juist
coqiq	Coqua	0.252	Fout
west	Went	0.396	Fout
rimbo, makati	Rumtomakati	0.42	Fout
city	City	0.727	Juist
29	19	0.272	Fout
M	M	0.846	Juist
12	12	0.762	Juist
03	-13	0.255	Fout
10	-90	0.284	Fout
Hiniox	Himox	0.291	Juist
omoxirillin	amoxicillin	0.264	Juist
500mg	Joong	0.252	Fout
cap#	Cap#	0.372	Juist
21	21	0.762	Juist
Sig:	Sig:	0.28	Juist
1	1	0.586	Juist
cap	cap	0.414	Juist
3x	ox	0.719	Fout
A day	aday	0.855	Juist
for	for	0.252	Juist
seven	sueen	0.487	Fout
days	days.	0.337	Juist

Figuur 5.4: Doktersbriefje 1 extractie resultaat

Name: Armando Coquia
Address: West Rimbo, Makati City
Age: 29 Sex: M Date: 12-03-90

R_x

(Tminox)

Amoxicillin 500mg Cap #21

Sig: 1 cap 3x a day for
seven days.

Physician's Sig. [Signature]
Lic. No. 123457
PTR No. 1234567
S2 No. _____

Figuur 5.5: Doktersbriefje 1. (Tacio, 2013)

5.2.3 Doktersbriefje 2 resultaat

Bij de resultaten van de extractie op figuur 5.7 zijn er in totaal 22 woorden geëxtraheerd, waarvan 2 onbepaald. Eveneens zijn bij de interpretatie 3 woorden als onbepaald verklaard. Dit fenomeen doet zich voor wanneer bepaalde woorden onherkenbaar zijn voor de API, waardoor deze overgeslagen worden. Dit was dus het geval bij de interpretatie van “biovik 100 mg tab”.

Een mogelijke oorzaak kan zijn dat 2 woorden te dicht bij elkaar staan of dat de gebruikte inkt mogelijk te licht is. Verder werden bepaalde woorden simpelweg oninterpreteerbaar verklaard door de testgroep, waardoor deze niet zullen bijdragen tot het gehele resultaat. De gemiddelde herkenningsgraad mits onbepaalde woorden is 0.525182, met 58 % van de voorspelde en geïnterpreteerde woorden correct voorspeld.

Interpretatie	Voorspelling	Herkenningsgraad	Voorspelling
Agatha	Agatha	0.727	Juist
peralta	Peralta	0.443	Juist
B4C	Ople	0.292	Fout
40	to	0.997	Fout
waray	manray	0.542	Fout
Onbepaald	s	0.572	Onbepaald
MC	Me	0.593	Fout
14	14	0.931	Juist
F	Onbepaald	/	Onbepaald
2-	2-	0.691	Juist
4-11	4-11	0.689	Juist
biovik 100 mgtab	Onbepaald	/	Onbepaald
Paracetamol	Paracetamol	0.278	Juist
#	#	0.862	Juist
12	12	0.944	Juist
Sig.	Gig.	0.261	Fout
take	take	0.563	Juist
1	1	0.29	Juist
tab	fat	0.262	Fout
2	g	0.303	Fout
4h	th	0.252	Fout
or	or	0.749	Juist
Onbepaald	an	0.313	Onbepaald
Onbepaald	haded	0.42	Onbepaald

Figuur 5.6: Doktersbriefje 2 extractie resultaat

Name: Agatha Peralta
Address: Blk 40 Manuway St. Mc
Age: 14 Sex: F Date: 2-4-11

R_x

Prionic 500mg tab
(Paracetamol) #12

qig: take 1 tab q 4h or
as needed

Physician's Sig. _____
Lic. No. 43452
PTR No. 1345900
S2 No. _____

Figuur 5.7: Doktersbriefje 2. (Tacio, 2013)

5.2.4 Doktersbriefje 3 resultaat

Volgens de resultaten van de extractie op figuur 5.9 zijn er in totaal 21 woorden geëxtraheerd, waarvan 1 onbepaald. Tegenover extractieresultaat 5.6 zijn hier veel minder woorden als onbepaald verklaard. Dit lijkt in zeker zin ook logisch als men kijkt naar de algemene slordigheid van figuur 5.7 vergeleken met figuur 5.9 Een terugkerende trend die men kan waarnemen bij deze resultaten, is de voorspelling van de medicatie dosering.

Deze is in geen enkel van de gevallen tot nu toe juist voorspeld. De gemiddelde herkenningsgraad mits 1 onbepaald woord is 0.507048 met 57 % van de voorspelde en geïnterpreteerde woorden correct voorspeld.

Interpretatie	Voorspelling	Herkenningsgraad	Voorspelling
Londorion	Lonpulation	0.364	Fout
parcual	Pascual	0.447	Fout
Boni	for	0.227	Fout
ave	ave,	0.383	Juist
MC	Mc	0.433	Juist
65	45	0.252	Fout
F	Onbepaald	/	Onbepaald
5	5	0.883	Juist
-	-	0.694	Juist
12-12	12-12	0.632	Juist
captlomil	Captopull	0.254	Fout
25mg	lsmy	0.117	Fout
tab	lab	0.4	Fout
captogen	Captogen	0.446	Juist
#	#	0.799	Juist
60	60	0.445	Juist
sig:	gig:	0.57	Fout
1	l	0.396	Fout
tot	tot	0.694	Juist
2x	2x	0.703	Juist
a	a	0.877	Juist
day	day	0.632	Juist

Figuur 5.8: Doktersbriefje 3 extractie resultaat

Name: Corporacion Pascual
Address: Boni Ave. MC
Age: 65 Sex: F Date: 5-12-12

Rx

Captopril 25mg tab
Captopren #60
Sig: 1 tab 2x a day

Physician's Sig. [Signature]
Lic. No. 43452
PTR No. 1345900
S2 No. _____

Figuur 5.9: Doktersbriefje 3. (Tacio, 2013)

5.2.5 Doktersbriefje 4 resultaat

Bij de laatste extractie request ziet men positieve resultaten. Hierbij zijn 27 woorden geëxtraheerd, zonder onbepaalde voorspellingen. Merk op dat de meerderheid van de foutief gerekende voorspellingen, maar 1 karakter verschillen van de interpretatie.

Een toekomstige oplossing voor dit probleem zou een fuzzy matching NLP model kunnen zijn. Hierbij worden woorden die enkele karakters van het originele afwijken, vergeleken met een generieke databank. Men kan op deze manier woorden die niet volledig juist verklaard zijn toch nog goedkeuren door wijze van vergelijking. Echter zou dit uiteraard de kost en tijd van extractie enkel maar vergroten. Kijkend naar het resultaat, werd er een gemiddelde herkenninggraad van 0.564259 per woord bekomen, waarbij 64% van de interpretatie correct voorspeld.

Interpretatie	Voorspelling	Herkenninggraad	Voorspelling
Lolita	Lolita	0.287	Juist
aluarey	aluasen	0.445	Fout
Bageng	Baging	0.46	Fout
Log.	blog!	0.252	Fout
Pazig	Pasig	0.252	Fout
City	City	0.694	Juist
39	39	0.762	Juist
F	F	0.613	Juist
2-5-13	12-5-13	0.387	Fout
FeSop	Fesoy	0.378	Fout
tab	tat	0.567	Fout
#	#	0.727	Juist
30	30	0.849	Juist
Sig	Sig	0.762	Juist
:	:	0.661	Juist
A.	A.	0.692	Juist
D.	D.	0.799	Juist
Ascorbic	Ascorbic	0.264	Juist
acid	acid	0.271	Juist
#	#	0.727	Juist
30	30	0.849	Juist
100 mg	toomg	0.363	Fout
tab	tab	0.799	Juist
Sig:	gig:	0.273	Fout
one	ance	0.289	Fout
a	a	0.873	Juist
day	day	0.94	Juist

Figuur 5.10: Doktersbriefje 4 extractie resultaat

Name: Lolita Alvarez
Address: Bagongilog, Pasig City
Age: 39 Sex: F Date: 2-5-13

Rx

FeSO₄ tab # 30

Sig: A.D.

Ascorbic Acid # 30
100mg tab

Sig: Once a day

Physician's Sig. J. Delacruz
Lic. No. 12345
PTR No. 1234567
S2 No. _____

Figuur 5.11: Doktersbriefje 4. (Tacio, 2013)

5.3 Test Case 3

5.3.1 Opzet

De derde en laatste test case zal nagaan of het uiterlijk van een doktersbriefje invloed heeft op het gemiddelde resultaat. Voorgaande doktersbriefjes gebruikt in test case 2, maakten gebruik van éénzelfde template. Men bekomt hierdoor een gestroomlijnd resultaat waarbij men de uitkomst enkel kan baseren op het handgeschrift. Echter is het wel nog interessant om een extractie uit te voeren op een doktersbriefje dat gebruik maakt van een ander template. Hierbij wordt de positionering en volgorde van extractie mogelijks anders opgenomen door de API. Bij deze test werd gebruik gemaakt van het doktersbriefje weergegeven in figuur 5.13. Merk op dat meerdere karakteristieken van figuur 5.13 duiden op een suboptimale extractie. Hierbij ziet men net zoals in test case 1, een lage resolutie gepaard met een slordig handschrift. In essentie stelt deze test een extreem geval voor dat zelden zal voorkomen bij een echte implementatie.

5.3.2 Doktersbriefje 5 resultaat

Kijkend naar de resulterende data zichtbaar of figuur 5.12, kan men vaststellen dat de gemiddelde herkenningsgraad 0.496833 bedraagt. In totaal werd 50 % van de interpretaties correct voorspeld door de API. Dit wijst m.a.w. op een daling van +- 8 % tegenover de voorgaande resultaten. De meerderheid van de medische termen zijn hierbij wel correct voorspeld, met de tot nu toe enige correcte voorspelling van de medische dosering.

Interpretatie	Voorspelling	Herkenningsgraad	Voorspelling
MB,	MC,	0.489	Fout
Onbepaald).	0.397	Onbepaald
Pair	Pang	0.401	Fout
126	125	0.441	Fout
Creek	Crack	0.252	Fout
Stivushrol	Stilleshot	0.28	Fout
25	25	0.841	Juist
mg	mg	0.816	Juist
milistial	milestial	0.637	Juist
Sig:	Sig:	0.469	Juist
199	194n	0.393	Fout
for	for	0.57	Juist
8	8	0.855	Juist
days	days	0.431	Juist
Min 1	then	0.762	Fout
Onbepaald	US	0.359	Onbepaald
Bhmin	Bhumi	0.252	Fout
#15926	#15926	0.298	Juist

Figuur 5.12: Doktersbriefje 5 extractie resultaat

MIDTOWN PARK PHARMACY

Phone: DAB-4907 or DAB-4908 FARMVIEW, TEXAS
915 MAIN STREET

For Mrs. J. Paris

Address 126 Creek

R Stilbestrol 25mg
(Miletral)

Sig: 1 qid for 8 days
then 1 qid

Drug No. B Humin #5926 M11

Date 7/17/57 Address _____

REPT BY DIST 1 2 3 4 TMRD P R N ☐ NON-RES ☐

Figuur 5.13: Doktersbriefje 5. (Abrahams, 2008)

5.4 Resultaten

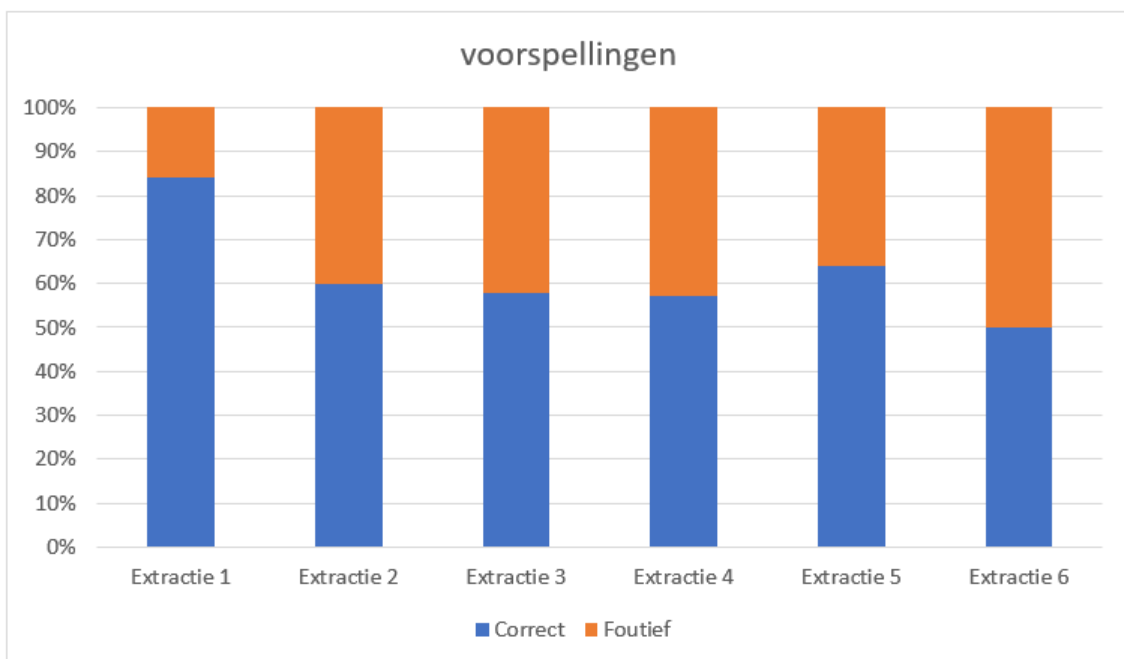
Na het uitvoeren van de tests, kunnen we collectief de resultaten verzamelen en vergelijken.

5.4.1 Test cases

Op figuur 5.14 ziet men een weergave van de bekomen extractie percentages. Kijkend naar test 1, werd 84 % van de geëxtraheerde woorden correct voorspeld. Vergeleken met de percentages bekomen in test case 2 en 3, ligt dit relatief hoog. Een mogelijke verklaring voor dit hoge percentage zou kunnen liggen bij de ruimere spreiding van de woorden op figuur 5.2.

Extracties 2-5 uitgevoerd in test case 2, tonen een relatief gelijkaardig resultaat. Bij deze test case werden 4 doktersbriefjes met éénzelfde template getest. Gemiddeld werd 59.75 % van de woorden correct voorspeld. De meerderheid van de foutieve voorspellingen lag bij de naam of het adres van de persoon in kwestie. In principe was dit te verwachten aangezien de focus bij de dataset minder ligt op unieke naamgevingen. Wat men echter wel kan vaststellen is dat de meerderheid van de medische termen correct voorspeld werden, wat bijdrage levert aan een positief resultaat.

Test case 3 toonde, vergelijkend met test case 2, relatief weinig tot geen verschil aan in resultaat. Dit geeft enerzijds ook aan dat het uiterlijk van een doktersbriefje het resultaat niet ongewild beïnvloedt. Wat echter wel een tegenargument voor deze bewering kan zijn is dat de spreiding van de woorden, gezien bij doktersbriefje 1, bekomen werd door een groter schrijfvlak. Wat op zijn beurt weer een beter resultaat zou geven.

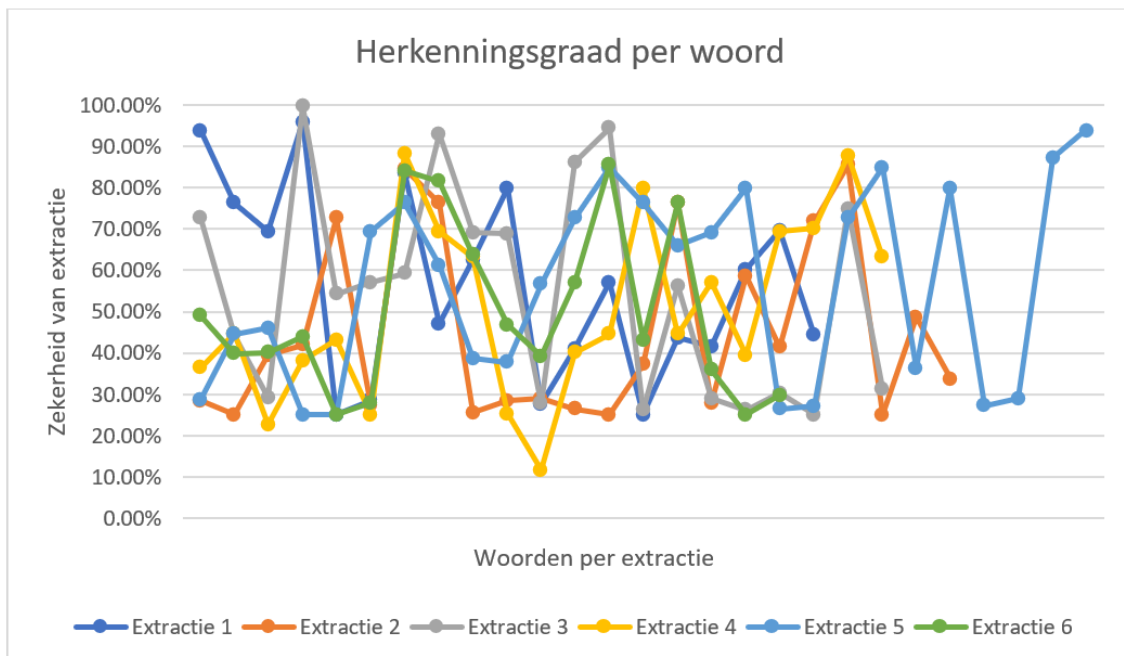


Figuur 5.14: Statistiek voorspellingen.

5.4.2 Herkenningsgraad

Verder, werd in figuur 5.15 het verloop van de herkenningsgraad per geëxtraheerd woord weergegeven. Elke weergegeven lijn stelt de volgorde voor waarin elk woord voorspeld werd, waarbij de y-as de zekerheid van extractie in % voorstelt. Zoals eerder beschreven, ligt de meerderheid van foutieve voorspellingen bij de extractie van naam of adres. De oorzaak hiervan zou enerzijds liggen bij gebrek aan opname van unieke naamgevingen bij de training van het model.

Merk op dat men dit fenomeen duidelijk kan waarnemen in figuur 5.15. De naam en adres van de patiënt worden in de meeste gevallen en dus ook bij de gebruikte doktersbriefjes bovenaan weergegeven. Aangezien in hoofdstuk 4 (sectie 4.0.6) beschreven werd dat de API, een lijn extractie van linksboven naar rechtsonder uitvoert, zullen deze dus eerst geëxtraheerd worden. Men ziet ook wel degelijk dat de eerste 4-5 woorden bij elke extractie zelden een herkenningsgraad boven de 50 % hebben.



Figuur 5.15: Verloop van de herkenningsgraad per extractie.

Collectief was de API bij het uitvoeren van de 3 test cases, per woord, gemiddeld 52 % zeker van zijn voorspelling. Dit getal zal uiteraard evolueren, naargelang het aantal testen dat men uitvoert, vergroot en de variatie van het type doktersbriefjes toeneemt.

5.5 Conclusie

Uit de resultaten kon men vaststellen dat de Computer Vision API wel degelijk bruikbaar is voor het extraheren van doktersgeschrift. Hierbij werd de meerderheid van de woorden correct voorspeld, gepaard met de juiste medische termen. De interpretaties verzameld door de testgroep werden grotendeels ook door de API voorspeld. Het is echter wel op te merken dat bij deze tests maar een beperkt aantal doktersbriefjes gebruikt werd. Dit komt grotendeels door de beperkte beschikbaarheid en vrijstelling van dit type document. Uiteraard zijn de meeste mensen niet bereid om hun naam en gegevens open te stellen aan een breder publiek. Aangezien het bekomen van test cases al een moeilijke zaak was zou dit al zeker een hindernis vormen bij het trainen van een volledig extractie model.

De Computer Vision service die bij dit onderzoek gebruikt werd, kan maximaal 5000 requests per maand ontvangen. Deze oplossing is ruim schaalbaar door middel van de Azure Cloud en kan mogelijks met enige configuratie op een eenvoudige wijze in een productie omgeving tewerkgesteld worden, mits hier een bruikbaar plan voor gekozen is.

Collectief kon men vaststellen dat de API gemiddeld 62% van een doktersbriefje volledig juist kan voorspellen. Wanneer de ontwikkelde software gebruikt zou worden in een productie omgeving waar eenzelfde type briefje voorkomt, dan zou men een meer gestroomlijnd resultaat bekomen.

Uiteraard is het onmogelijk om enkel uit deze uitgevoerde testen een volledige conclusie te trekken, maar het wijst wel degelijk richting een positief resultaat.

6. Conclusie

Bij aanvang van deze studie werden de limitaties van handschriftherkenning op de proef gesteld. Hiermee kwam de CM initieel met de vraag of dergelijke software instaat was om handgeschreven teksten van een doktersbriefje te extraheren. Uit de voorgaande literatuurstudie bleek dat Cloud Computing de laatste nieuwe trend is in de wereld van AI en Machine Learning. Door het gebruik van de Cloud, werd de nood aan confidentiële trainingsdata en opzet van een specifiek ML model, geëlimineerd.

Het is ook echter nog op te merken dat de extracties en test cases uitgevoerd werden op engelse teksten. Hier werd bewust voor gekozen aangezien de Computer Vision API enkel Engel en Spaans ondersteund voor handschriftherkenning. Dit zou naargelang de evolutie van Azure Cognitive Services veranderen. De mogelijkheid voor handgeschrift in het Nederlands bestaat echter wel zoals gezien bij het onderzoek van Vectr. Consulting. Hierbij verkregen zij een vooraf gelabelde dataset door hun samenwerking met het Vlaams Agentschap Zorg & Gezondheid.

Hierbij werd de Computer Vision API van Microsoft Azure tewerkgesteld om een “proof of concept” applicatie op te zetten. Uit de resulterende extracties bleek dat een doktersbriefje voor gemiddeld 62% gelijk gesteld werd aan de menselijke interpretaties. Daarnaast bleek de API weinig tot geen problemen te hebben bij het herkennen van medische terminologie. Deze ondervinding stelt wel degelijk voor dat rekening gehouden werd met een medisch doelpubliek naargelang training van het model.

Naast een generieke extractie uit te voeren om vast te stellen dat de API wel degelijk capabel is voor productiedoeleinden, kwamen enkele deelvragen naar boven die mogelijks invloed konden hebben op het gehele resultaat. Hierbij werden mogelijke limitaties op proef gesteld, waaronder de invloed van kwaliteit en uiterlijk op het extractie resultaat. Een

doktersbriefje met een verlaagde resolutie toonde wel degelijk een verschil aan in resultaat. Om de verdere evolutie van dit fenomeen vast te stellen, zou een apart onderzoek naar de invloed van resolutie op afbeeldingen binnen Computer Vision, gestart moeten worden.

Verder werd eveneens de impact van het fysieke uiterlijk van het voorschrift en de positionering van de woorden getest. Bij het gebruik van drie verschillende types voorschriften, bleek een voorschrift met een ruimere spreiding van de woorden, een beter resultaat aan te tonen. Het is vanzelfsprekend dat men dokters niet kan forceren omrekening te houden met deze spreiding, maar het is wel op te merken dat het helpt bij extractie.

Uiteindelijk kan men uit dit onderzoek concluderen dat handschriftherkenning wel degelijk een verschil kan maken bij het efficiënt en performant extraheren van doktersgeschrift. De CM zou hierbij een meerwaarde bekomen door de automatisering van dit proces. De conclusie van dit onderzoek werd enerzijds deels aangereikt in het onderzoek uitgevoerd door Vectr. Consulting. Het is wel degelijk mogelijk om dit proces doormiddel van AI te automatiseren, maar men heeft hier mogelijks de samenwerking met een derde partij voor nodig, aangezien data niet zomaar weg gegeven wordt.

Een generiek ML model zoals aangeboden door de Computer Vision API is dus wel degelijk in staat om een gelijkaardig resultaat te leveren als dat van een op maat gemaakt model.

A. 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.

A.1 Introductie

Probleemstelling: Handgeschrift op doktersbriefjes is in sommige gevallen onleesbaar voor de christelijke Mutualiteit. Het is nu ook niet wetenschappelijk bewezen dat dokters het slechtste handgeschrift hebben, maar het wordt wel sociaal vastgesteld.

Motivatie: Het gebied van Artificiële intelligentie en machine learning spreekt toekomst. Bijna ieder toestel maakt al op een manier gebruik van AI. Door de opkomst van IoT genereren miljoenen gebruikers dagelijks data die gebruikt wordt om de accuraatheid van AI geïntegreerde systemen te vergroten.

We kunnen hier dan ook uit vaststellen dat in de nabije toekomst het gebruik van AI alleen maar effectiever zal worden. Het is dus belangrijk dat we hier zo snel mogelijk op inspelen en een domein onderzoeken waar we dit dan ook kunnen gebruiken.

Doelstelling: De werkelijke onderzoeksvraag is dus of de workflow efficiëntie van de christelijke Mutualiteit verbeterd kan worden door middel van het integreren van handschriftherkenning. Dit zal natuurlijk in meerdere stappen uitgevoerd worden. Door het verzamelen van voorgaande data en het trainen van het neurale netwerk wordt verwacht dat dit model het handgeschrift op doktersbriefjes kan herkennen/ parsen met een efficiëntie van +- 47 %. Dit zal uiteraard naarmate de grootte van de dataset verbeteren.

A.2 State-of-the-art

Zoals vooraf al aan bod kwam is deze onderzoeksvraag gebaseerd op een recent onderzoek dat uitgevoerd werd door Vectr.Consulting in samenwerking met de Christelijke Mutualiteit. In dit onderzoek werd een model opgezet en getraind om doktershandgeschrijf te herkennen op doodsakten. Verder, baseerde het praktische gedeelte van dit uitgevoerde onderzoek zich op het open source Tensorflow framework. Dit onderzoek zal vooral gebaseerd zijn rond de Microsoft Azure stack, meer bepaald de afdeling Cognitive Services. Microsoft hanteert hiervoor hun krachtig cloud networking solution om op basis van vooraf getrainde modellen realtime data te bezorgen aan de consument. Dit framework is vooral nog in de early development stages en zal naarmate het gebruik alleen maar verbeteren.

A.3 Methodologie

In zeker zin zal er een observatieonderzoek plaats vinden. De dataset zal bestaan uit vooraf geschreven doktersbriefjes waarbij het computer vision gedeelte van azure cognitieve services de data uit elke image extraheert en analyseert. Door deze methodologie zal de eindgebruiker de mogelijkheid hebben de geëxtraheerde data een rating te geven op basis van accuraatheid van foto naar tekst extractie. Het model zal hierdoor alleen maar bijleren en zichzelf aanpassen naar toekomstige data.

A.4 Verwachte resultaten

Na het trainen en modelleren van de data zal het handgeschrijf op doktersattesten met een nauwkeurigheid van omtrent 47 % voorspeld kunnen worden. Data zal gestructureerd worden in functie van de kwaliteit en hoe zeker het model is van zijn voorspelling. Om dit percentage te bereiken zal het model enkel getraind worden met data die aan een bepaalde kwaliteitsstandaard voldoet en zal na elke voorspelling de eindgebruiker geprompt worden om de kwaliteit van extractie te raten. Medische termen zullen hierbij een zekere rol spelen, aangezien deze niet altijd de makkelijkste zijn om te herkennen.

A.5 Verwachte conclusies

Wanneer de bevindingen concorderen met hetgeen verwacht wordt, wordt er gesteld dat de Christelijke Mutualiteit een meerwaarde in workflow en resources zal bekomen, in de hoop dat meerdere taken door artificiële intelligentie geautomatiseerd zullen worden. Dit is natuurlijk een oplossing die op lange termijn zichzelf zal verbeteren tot op het punt dat de accuraatheid van de voorspelde tekst naar een 80-90 % convergeert.

A.6 Referenties

<https://vectr.consulting/news/deciphering-handwriting>

<https://vectr.consulting/news/smart-scanner/artificial-intelligence>

<https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>

<https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5>

Bibliografie

- Abrahams, M. (2008). *Problems reading your prescription?* <https://www.improbable.com/2008/04/18/guardian-column-109/>
- Amazon. (2019). *Detecing Text in an Image*. <https://docs.aws.amazon.com/rekognition/latest/dg/text-detecting-text-procedure.html>
- Amazon. (2020a). *Amazon Rekognition*. https://aws.amazon.com/rekognition/?nc2=h_ql_prod_ml_rek
- Amazon. (2020b). *What is Amazon Rekognition?* <https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html>
- Bhardwaj, S., Jain, L. & Jain, S. (2010). CLOUD COMPUTING: A STUDY OF INFRA-STRUCTURE AS A SERVICE (IAAS).
- Breithaupt, M. (2014). Improving OCR and ICR Accuracy Through Expert Voting.
- Bunke, H., Roth, M. & Schukat-Talamazzini, E. G. (1995). Off-line Cursive Handwriting Recognition using Hidden Markov Models.
- Gergely Timár, K. K. & Rekeczky, C. (2003). Analogic Preprocessing and Segmentation Algorithms for Off-line Handwriting Recognition.
- Google. (2020a). *AI and machine learning products*. <https://cloud.google.com/products/ai/>
- Google. (2020b). *Cloud locations*. <https://cloud.google.com/about/locations>
- Google. (2020c). *Detect handwriting in images*. <https://cloud.google.com/vision/docs/handwriting>
- Google. (2020d). *Vision AI*. <https://cloud.google.com/vision>
- Google. (2020e). *Why Google cloud?* <https://cloud.google.com/why-google-cloud>
- Joson, R. O. (2017). *Legible Physician Handwriting - An Advocacy - A Legacy*. <https://sites.google.com/site/rojoson70memoir/legible-physician-handwriting---an-advocacy---a-legacy>
- Manish Godse, S. M. (2009). An Approach for Selecting Software-as-a-Service (SaaS) Product.

- Microsoft. (2017). *Computer Vision API - v3.0 Preview*. <https://westcentralus.dev.cognitive.microsoft.com/docs/services/5d98695995feb7853f67d6a6/operations/5d9869604be85dee480c8750>
- Microsoft. (2019a). *Call the Computer Vision API*. <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/vision-api-how-to-topics/howtocallvisionapi>
- Microsoft. (2019b). *Language support for Computer Vision*. <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/language-support#text-recognition>
- Microsoft. (2019c). *Recognize printed and handwritten text*. <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-recognizing-text>
- Microsoft. (2020a). *Azure Dev/Test Pricing*. <https://azure.microsoft.com/en-us/pricing/dev-test/>
- Microsoft. (2020b). *Computer Vision*. <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>
- Microsoft. (2020c). *Deep learning vs. machine learning*. <https://docs.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning#deep-learning-machine-learning-and-ai>
- Microsoft. (2020d). *Introduction to cloud computing*. <https://azure.microsoft.com/en-us/overview/>
- Microsoft. (2020e). *Prijsinformatie Cognitive Services - Computer Vision-API*. <https://azure.microsoft.com/nl-nl/pricing/details/cognitive-services/computer-vision/>
- Microsoft. (2020f). *Quickstart: Extract printed and handwritten text using the Computer Vision 2.0 and 2.1 REST API and Csharp*. <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/quickstarts/csharp-hand-text?tabs=version-3>
- Microsoft. (2020g). *What is cognitive services?* <https://azure.microsoft.com/en-us/services/cognitive-services/>
- Microsoft. (2020h). *What is machine learning?* <https://azure.microsoft.com/en-us/overview/what-is-machine-learning-platform/>
- Nafiz Arica, F. T. Y.-V. (2001). An Overview of Character Recognition Focused on Off-Line Handwriting.
- Nagy, G., Nartker, T. A. & Rice, S. V. (1999). Optical Character Recognition: An illustrated guide to the frontier.
- Réjean Plamondon, S. N. S. (2000). On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey.
- Salim Al-Habsib, A. I. K. (2019). Machine Learning in Computer Vision.
- Sanjay Kumar, N. S. (2016). Offline Handwriting Character Recognition (for use of medical purpose) Using Neural Network.
- Tacio. (2013). <http://rxistsource.blogspot.com/2013/07/samples-of-erroneous-violative-and.html>
- Tappert, C. C., Suen, C. Y. & Wakahara, T. (1990). The State of the Art in On-Line Handwriting Recognition.
- Vaibhav A Ghandi, D. C. K. K. (1999). Comparative study of Amazon EC2 and Microsoft Azure cloud architecture.
- Vijay Laxmi Sahu, B. K. (2012). Offline Handwritten Character Recognition Techniques using Neural Network: A Review.

-
- Wright, N. (2019). *IaaS vs SaaS vs PaaS: A guide to Azure cloud service types*. <https://www.nigelfrank.com/blog/iaas-vs-saas-vs-paas-a-guide-to-azure-cloud-service-types/>
- Zhu, X. (2005). Semi-Supervised Learning Literature Survey.