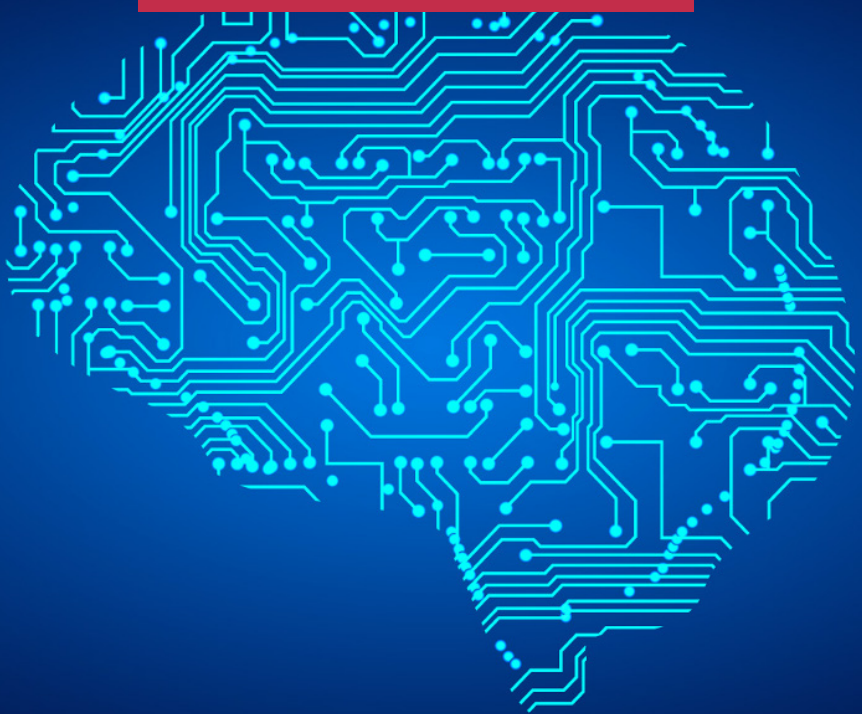
Scriptie

Productclassificatie aan de hand van productafbeeldingen

Björn de Mul

7 juni 2018

Quintor



AFSTUDEERSTAGE IN DIENST VAN QUINTOR

In opdracht van:

HZ UNIVERSITY OF APPLIED SCIENCES

Scriptie

2.0

7 juni 2018

Student:

Björn de Mul (69654)
mul0004@hz.nl

beoordelaars:

Andries Nieuwenhuize (Hz)
Daan de Waard (Hz)

Begeleider:

Ben Ooms (Quintor)

Wijzigingen

Herziening	Datum	Auteur(s)	Beschrijving
0.1	1 februari 2018	B. de Mul	Opzetten sjabloon onderzoeksrapport
0.2	5 februari 2018	B. de Mul	Toevoeging inleiding
0.3	10 februari 2018	B. de Mul	Toevoeging theoretisch kader
0.4	13 februari 2018	B. de Mul	Toevoeging methode
0.5	2 maart 2018	B. de Mul	Spelling- en grammaticacontrole
0.6	14 maart 2018	B. de Mul	Onderzoeksvraag en deelvragen bijgewerkt
0.7	19 maart 2018	B. de Mul	Taaltechnisch gecontroleerd
0.8	20 maart 2018	B. de Mul	Onderzoek aangepast aan de hand van de eerste feedback
0.9	2 april 2018	B. de Mul	Toevoeging resultaten deelvraag 1
1.0	12 april 2018	B. de Mul	Toevoeging resultaten deelvraag 2
1.1	01 mei 2018	B. de Mul	Aanpassingen aan de hand van feedback
1.2	07 mei 2018	B. de Mul	Toevoeging resultaten deelvraag 3
1.3	12 mei 2018	B. de Mul	Toevoeging conclusie
1.4	13 mei 2018	B. de Mul	Toevoeging discussie
1.5	15 mei 2018	B. de Mul	Toevoeging aanbeveling
1.6	16 mei 2018	B. de Mul	Toevoeging voorwoord
1.7	16 mei 2018	B. de Mul	Toevoeging samenvatting
1.8	25 mei 2018	B. de Mul	Feedback verwerken
1.9	30 mei 2018	B. de Mul	Feedback verwerken
2.0	7 juni 2018	B. de Mul	Uiteindelijke versie

Voorwoord

Terugkijkend op het afstuderen ben ik een tevreden persoon, ik heb iets moois en innoverends neergezet. Ik heb een innovatief programma ontwikkeld, dat producten classificeert door slechts enkele afbeeldingen te uploaden. Bij aanvang van dit afstudeeronderzoek/afstudeerperiode had ik nooit verwacht dat dit het eindresultaat zou zijn. Door activiteiten uiteenlopend van uitzoek- en programmeerwerk tot reflecteren of evalueren. Buiten de grote hoeveelheid werk was het ook een half jaar van veel kilometers, maar ik me reuze vermaakt en was het een geweldige ervaring.

Quintor heeft mij de mogelijkheid geboden om een innoverend product te ontwikkelen, die direct in de praktijk kan worden toegepast. Door deze ervaring heb ik veel nieuwe kennis en inzichten opgedaan, waarvoor ik Quintor dan ook hartelijk wil bedanken. Natuurlijk wil ik alle betrokken personen bij Quintor bedanken, maar in het bijzonder Ben Ooms en Lissanne van Dam voor hun inzet en enthousiaste ondersteuning.

Vrijwel dagelijks reed ik de afgelopen periode heen en weer naar Den Haag, wat ongeveer 314 kilometer was. Het was dan ook ontzettend fijn dat ik altijd de mogelijkheid had om te overnachten bij Stef Jansen. Stef, bedankt voor de gezelligheid en het feit dat ik altijd bij je terecht kon!

Eugenie de Mul, Jordy de Mul, Marjolein Oude Luttikhuis, Stef Jansen en Thijs van de Eerden, ook jullie wil ik bedanken voor het controleren van mijn scriptie. Het taaltechnische aspect van mijn scriptie vond ik het lastigst, maar dankzij jullie hulp is mij dit gelukt.

Als laatste wil ik iedereen bedanken die mij heeft gesteund tijdens deze periode en ervoor gezorgd heeft dat ik zonder al te veel zorgen door deze periode heen ben gekomen.

Björn de Mul
7 juni 2018. Den Haag

Samenvatting

Dit onderzoek heeft als doel om Quintor te adviseren op welke wijze machine learning de huidige manier van classificeren bij ToolMax kan verbeteren. Hierbij hoort de volgende onderzoeksvraag **"Op welke wijze kan het toepassen van een convolutional neural network de huidige manier van classificeren bij Toolmax verbeteren, om automatische productclassificatie mogelijk te maken?"** De bijbehorende deelvragen zijn:

1. aan welke eisen moet het nieuwe systeem minimaal voldoen,wil het automatische productclassificatie voor ToolMax mogelijk maken?
2. Op welke manier kan een zelflerende machine worden ingericht voor het classificeren van producten, welke voldoet aan de gespecificeerde eisen?
3. Op welke manier kan de zelflerende machine ingezet worden, om de beoogde resultaten in de praktijk te behalen?

Het onderzoek was kwalitatief van aard. Om tot de juiste resultaten te komen is er gekeken naar de verschillende eisen die aan het systeem wordt gesteld, welke modellen hiervoor nodig waren en hoe deze toegepast konden worden in de praktijk. Uit de resultaten kan geconcludeerd worden dat, door het inzetten van een neuraal netwerk, het handmatig classificeren met minstens 46% gereduceerd kan worden. Het geautomatiseerde classificatieproces kan eenvoudig ingeschakeld worden door een interface.

Vanwege een beperkt tijdsbestek zijn de modellen in het onderzoek slechts tot een minimaal niveau getraind. Op basis hiervan wordt er geadviseerd om de verschillende modellen langer te trainen, zodat de resultaten nauwkeuriger zijn. Daarnaast wordt er aanbevolen om dit onderwerp verder te onderzoeken, waardoor dit mogelijk nog meer efficiëntie kan opleveren voor ToolMax.

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Afkortingen

ANN Artificial Neural Network. iv, 3, 4, 5, 6

API Application Programming Interface. 18, 20, 24

CLI Command Line Interface. 16

CNN Convolutional Neural Network. iv, vi, 4, 5, 6, 7, 8, 9, 14, 15, 16, 22, 24, 25, 26

ILSVRC ImageNet Large-Scale Visual Recognition Challenge. 8, 14, 15

OTAP Ontwikkeling, Test, Acceptatie en Productie. 9

PoC Proof of Concept. 9, 10, 24, 26

ReLu Rectified Linear Unit. 4, 5, 14

SMART Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdgebonden. 12

UI User Interface. iv, 18, 20, 24

1 Inleiding

ToolMax is een bedrijf dat in 2005 is opgericht. Het bedrijf heeft zich gespecialiseerd in de levering van gereedschap via internet (ToolMax, 2018). Door de jaren heen is ToolMax sterk gegroeid, zij bieden zowel aan bedrijven als particulieren meer dan 90.000 producten (ToolMax, 2018). Dit onderzoek is geschreven naar aanleiding van de vraag van ToolMax om het proces van productclassificatie te automatiseren door middel van “machine learning”. Bij dit proces worden producten geclassificeerd aan de hand van productafbeeldingen. Quintor heeft dit onderzoek op zich genomen en vervolgens uitbesteed aan afstudeerstudenten van verschillende hogescholen. Quintor geeft aan dat zij zelf te weinig kennis hebben met betrekking tot “machine learning”. Door middel van dit onderzoek, hopen zij dan ook hun interne kennis te verbreden.

Quintor is een toonaangevend bedrijf op het gebied van: Agile software development, enterprise Java, .Net technologie en Mobile development. Binnen Quintor werken 123 mensen, verspreid over vestigingen in Groningen, Amersfoort en Den Haag (Quintor, z. j.). Daarnaast heeft Quintor samenwerkingsverbanden met enkele hogescholen, waaronder de Haagse hogeschool en de hogeschool van Utrecht.

1.1 Aanleiding

E-commerce sites zoals ToolMax hebben een flink aantal producten van verschillende categorieën, waarbij correct classificeren van groot belang is. Doormiddel van classificeren wordt de vindbaarheid van het product namelijk vergroot (Granada Research, 2001). Tevens helpt classificeren bij de verkoop en de distributie van producten, doordat dit automatisch productinformatie naar de klanten en/of search engines brengt (Granada Research, 2001).

1.1.1 Knelpunten ToolMax

Voor ToolMax is het een arbeids- en tijdsintensieve klus om alle producten te classificeren, zo stelt de eigenaar van ToolMax in het interview[1]. Elke dag zijn er twee medewerkers bezig met het classificeren van producten. ToolMax biedt 90.000 producten aan, onderverdeeld in ongeveer 200 categorieën. Er zijn ongeveer 130 leveranciers die deze producten leveren. Nieuwe productoverzichten komen dagelijks meerdere malen binnen. Gemiddeld duren de classificaties een uur, maar dit kan echter langer duren als een overzicht veel nieuwe producten of subcategoriën bevat.

De eigenaar van ToolMax geeft aan dat de kwaliteit van ToolMax.nl daalt, doordat er geneigd is om producten te classificeren onder generieke categoriën. Dit wordt gedaan vanwege tijdsbesparing, maar bezoekers van de site moeten hierdoor juist langer zoeken naar het gewenste product. Op deze manier loopt ToolMax klanten mis, wat Granada Research (2001) ook stelt.

1.1.2 Het proces

In “B: Classificatieproces ToolMax” is in een flowdiagram weergegeven hoe het huidige proces van productclassificatie bij ToolMax is ingericht. Het diagram toont aan dat een productclassificatieproces soms meer dan een uur kan duren. Indien het proces niet goed verloopt, kan dit zelfs nog langer duren, omdat de gemaakte stappen continu herhaald moeten worden.

Door bovenstaande complicaties is bij ToolMax de vraag ontstaan om het proces te automatiseren. Automatisering is mogelijk door gebruik te maken van productclassificatie aan de hand van bijbehorende afbeeldingen.

[1] Het interview kan terug gevonden worden in A: Interview antwoorden - vooronderzoek

1.1.3 Relevantie

Bedrijfsrelevantie. ToolMax biedt 90.000 producten aan, welke onderverdeeld zijn in ongeveer 200 categorieën. Door dit grote aanbod is het classificeren van producten voor ToolMax een tijdrovende en kostbare klus. Dit onderzoek tracht, door middel van automatisering, de benodigde arbeidsuren en daarbijhorende kosten te reduceren. De eigenaar van ToolMax geeft ook aan dat het nieuwe systeem mogelijk toegepast kan worden om de verschillende producten, die al gecategoriseerd zijn, te valideren. Met andere woorden classificeert het systeem de verschillende producten op de site, waarna gekeken kan worden of de classificatie overeenkomt met de classificatie die het product heeft. Op deze manier kan de kwaliteit van ToolMax verhoogd worden. Bovenstaande toepassingen maken beide gebruik van de verschillende productafbeeldingen die bij een product horen.

Maatschappelijke relevantie. Het onderzoek richt zich slechts op het classificatieproces van ToolMax, waardoor het weinig tot geen invloed heeft op het maatschappelijk belang. De maatschappij, of specifiek de klanten, zullen de veranderingen niet opmerken. Enkel de methode achter het categoriseren van producten zal veranderen, de uitkomst blijft hetzelfde.

Kennisgebied en theoretische relevantie. Dit onderzoek valt binnen de volgende kennisgebieden: afbeeldingverwerking, beeldverwerking en patroonherkenning. Het analyseren van afbeeldingen, voornamelijk complexere afbeeldingen met verschillende kenmerken, blijft een ingewikkeld proces waarin nog geregeld fouten voorkomen. In dit onderzoek wordt onderzocht welke eigenschappen afbeeldingen gemeen hebben en hoe deze gekoppeld kunnen worden aan verschillende producten. Daarnaast wordt de kennis van “machine learning” bij Quintor vergroot.

1.2 Probleemstelling

1.2.1 Doelstelling

Het doel van dit onderzoek is advies geven over de wijze waarop machine learning de huidige manier van classificeren bij ToolMax kan verbeteren. Hierbij wordt er gekeken waar de machine aan moet voldoen en hoe deze zowel ingericht als ingezet kan worden. Er zal specifiek gekeken worden naar de convultional neural networks, omdat deze zeer goede resultaten behalen bij verschillende standaardafbeelding classificatiecompetities (Benenson, 2016).

1.2.2 Vraagstelling

De onderzoeksvraag luidt als volgt: “Op welke wijze kan het toepassen van een convolutional neural network de huidige manier van classificeren bij ToolMax verbeteren, om automatische productclassificatie mogelijk te maken?”

Aansluitend hierop horen de volgende deelvragen:

1. Aan welke eisen moet het nieuwe systeem minimaal voldoen, wil het automatische productclassificatie voor ToolMax mogelijk maken?
2. Op welke manier kan een zelflerende machine worden ingericht voor het classificeren van producten, welke voldoet aan de gespecificeerde eisen?
3. Op welke manier kan de zelflerend machine ingezet worden, om de beoogde resultaten in de praktijk te behalen?

2 Theoretisch kader

Om producten te classificeren aan de hand van desbetreffende afbeeldingen, moeten de verschillende productafbeeldingen geanalyseerd worden. Het analyseren van deze producten valt binnen de vakgebieden: afbeeldingverwerking, beeldverwerking en patroonherkenning. Het classificeren van productafbeeldingen, in de verschillende vakgebieden, is reeds eerder onderzocht. In dit hoofdstuk zullen de meest relevante onderzoeksuitslagen worden beschreven, welke van invloed zijn op het onderzoek in opdracht van Quintor.

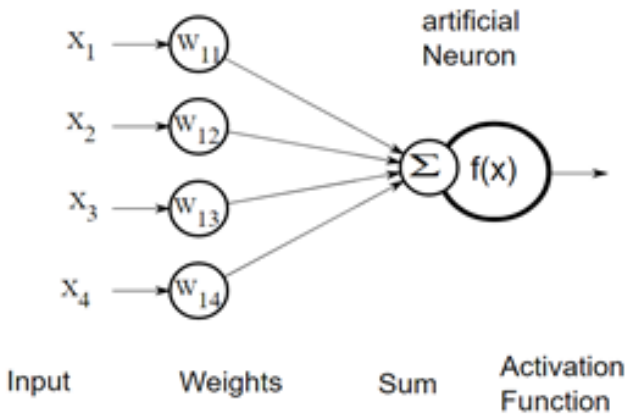
2.1 Productclassificatie

E-Commerce is in de laatste jaren explosief gegroeid. Grote E-commerce sites zoals Amazon, Ebay en Alibaba hebben miljoenen producten op hun sites welke verkocht worden door duizenden verkopers. Ebay biedt een miljard producten aan, welke onderverdeeld worden in duizenden klassen (Ebay Inc., 2018). Deze klassen worden gebruikt om ervoor te zorgen dat gebruikers eenvoudig kunnen navigeren en daarbij producten gemakkelijker vinden. Hierdoor is productclassificatie essentieel voor elk E-commerce bedrijf, aangezien dit de handel faciliteert tussen de kopers en verkopers (Granada Research, 2001). Uit het bovenstaande kan geconcludeerd worden dat dankzij productclassificatie verschillende producten onder één generieke zoekterm vallen, waardoor ze gemakkelijker te vinden zijn.

2.2 Artificial Neural Network (ANN)

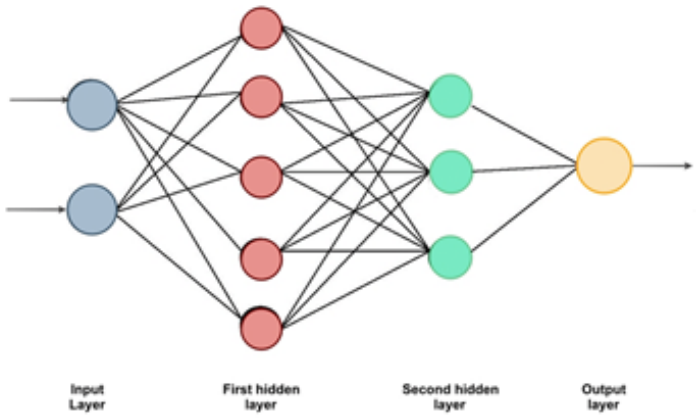
Het menselijk brein is een grote processor met 100 miljard basisblokken die ervoor zorgen dat de mens een geheugen heeft (Hagan, Demuth & Beale, 2014). Er kan hierbij gedacht worden aan voorbijgaande ervaringen die worden toegepast in nieuwe situaties. Deze basisblokken staan bekend als zenuwcellen. Elke zenuwcel kan tot wel 200.000 andere zenuwcellen met elkaar verbinden. In de praktijk verbindt een zenuwcel echter tussen de 1.000 en 10.000 zenuwcellen met elkaar. De kracht van het menselijk brein komt van de gigantische hoeveelheid zenuwcellen en de grote hoeveelheid connecties die deze onderling hebben. ANN proberen de meest eenvoudige elementen van het menselijk brein na te bootsen (de zenuwcellen) en zodoende een systeem te creëren dat beslissingen kan nemen aan de hand van eerdere ervaringen (Hagan et al., 2014).

Een ANN werkt door middel van nodes/units, welke vergeleken kunnen worden met de zenuwcellen in het menselijk brein. Dit zijn de basiselementen van een ANN (Hagan et al., 2014). Een node ontvangt data/input van andere nodes of externe bronnen en berekent hieruit een uitvoerwaarde. Elke invoer bevat een gewicht en is gebaseerd op de prioriteit van de daarbij behorende input.



Figuur 2.1: Bovenstaande illustratie (Hloewe, z. j.) illustreert een node

De node voert een functie “f” uit op de som van de input en de daarbij behorende gewichten. De functie “f” wordt ook wel de activatie functie genoemd, waarbij de uitvoer van de node gelijk is aan $y = f(w_{11} * X_1 + w_{12} * X_2 + ... + w_{1n} * X_n + b)$. Een neuraal netwerk is een clustering van verschillende nodes. Deze clustering wordt gerealiseerd door verschillende lagen die verbonden zijn met andere lagen (Hagan et al., 2014)



Figuur 2.2: Bovenstaande illustratie (Harrach, z. j.) illustreert een kunstmatig neurale netwerk

In feite hebben alle neurale netwerken dezelfde opbouw. Er zijn twee lagen die connecties hebben met de buitenwereld: de zogeheten input- en outputlaag. De inputlaag ontvangt de inputdata van de buitenwereld en de outputlaag zendt data naar de buitenwereld (Hagan et al., 2014). Daarnaast zijn er eventueel meerdere verborgen lagen, waarbij de nodes verschillende acties uitvoeren op de inputs die ze ontvangen. De outputs worden vervolgens doorgestuurd naar alle nodes in de daaropvolgende laag. Op deze manier ontstaat er een zogenaamde “feedforward” pad naar de output (Hagan et al., 2014).

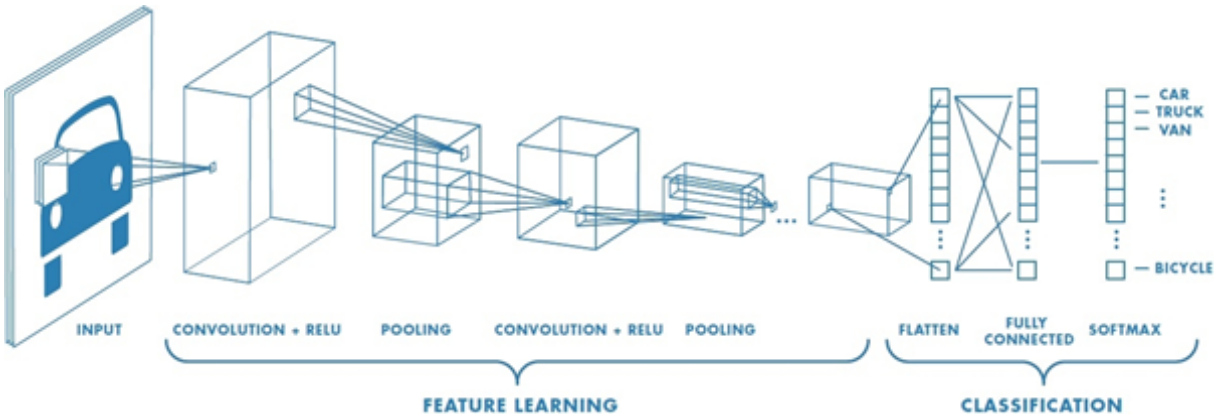
Zolang een ANN niet getraind is, zal deze geen antwoord kunnen geven op vragen die gesteld worden. Vandaar dat een ANN altijd van tevoren getraind dient te worden, omdat deze anders geen beslissingen kan nemen. Dit komt doordat een ANN beslissingen neemt die gebaseerd zijn op eerdere ervaringen. Een ANN kan aan de hand van een supervised en unsupervised getraind worden (Hagan et al., 2014).

Bij supervised training krijgt de ANN een input met een bijbehorende output. Het netwerk verwerkt de input en vergelijkt de berekende uitkomst met de gewenste en/of gegeven uitkomst. Fouten worden in dit proces terug gepropageerd naar de inputlaag, waarbij tevens de verschillende gewichten worden aangepast. Gedurende de training zal het bovenstaand proces een voorgedefinieerd aantal keer herhaald worden, waardoor de gewichten worden gefinetuned en de uitkomsten steeds dichter bij het gewenste resultaat komen (Hagan et al., 2014).

Bij een unsupervised training wordt het systeem enkel input aangeboden. Het systeem moet zelf beslissen op basis van welke features de input data moet worden gegroepeerd. Dit proces wordt ook wel een zelforganiserend of zelf aanpassend systeem genoemd (Hagan et al., 2014)

2.3 Convolutional Neural Network (CNN)

Een CNN is een alternatieve vorm van een ANN. Een CNN bestaat uit nodes, welke beslissingen maken aan de hand van hetgeen wat ze geleerd hebben uit voorgaande ervaringen. Het verschil tussen een ANN en CNN, is dat een CNN ervan uit gaat dat het enkel afbeeldingen als input krijgt. Op deze manier kan er in de architectuur specifieke eigenschappen van afbeeldingen worden toegepast, waardoor de “feedforward” efficiënter geïmplementeerd kan worden en er minder parameters nodig zijn in de verschillende lagen (Karpathey, 2017). Met andere woorden, een CNN verwerkt enkel afbeeldingen en eenANN kan alles verwerken. In het algemeen heeft een CNN meerdere verborgen lagen: De convolutional, de Rectified Linear Unit (ReLu), de pooling en de fully connected laag (Karpathey, 2017).



Figuur 2.3: Schematische overzicht van een CNN

De convolutional laag bestaat uit een set van zelflerende filters en berekent het dot product[1] van deze filters met de sectie van de afbeelding, dus de filter wordt over de afbeeldingsmatrix heen geschoven (beter nog, convolve) en er wordt een nieuwe matrix gecreëerd als output. Deze output gaat verder naar de ReLu laag. De filter zorgt er eigenlijk voor dat de afbeelding onder verschillende situaties bekeken wordt, wat "mapping van features" heet (Karpathey, 2017).

De ReLu voert een activatiefunctie uit waarbij alle waardes met $x < 0$ naar nul worden geplot en alle waardes met $x > 1$ naar een lineaire functie worden geplot met een steilheid van 1[2]. Convolution is een lineaire operatie die de input omzet naar lineare data. De input van het CNN is echter vaak niet lineair (Karn, 2016). De ReLu laag zorgt dan ook dat de niet-lineariteit terug aan het systeem wordt toegevoegd.

De pooling laag, ook wel subsampling of downsampling (Karpathey, 2017), zoomt in op verschillende onderdelen van een afbeelding. Het verkleint de feature map met behoud van de meest belangrijke informatie, onbelangrijke data wordt weggefilterd. Dit kan bijvoorbeeld door een "spatial oppervlakte" te pakken (bijvoorbeeld een 2 x 2 matrix) en dan daaruit steeds het grootste element te nemen (Karpathey, 2017).

De convolutional, ReLu en pooling laag kunnen meerdere keren herhaald worden. Hoe vaker deze "lagen" herhaald worden, hoe nauwkeuriger het resultaat. Deze drie lagen vormen samen het lerende gedeelte van het CNN.

De laatste lagen zijn een of meerdere "fully connect lagen". Hierin heeft een node connectie met alle andere nodes in de vorige laag, wat tevens ook het enige verschil is met bijvoorbeeld de convolutional laag, waarin de daadwerkelijke classificatie gebeurt (Karpathey, 2017).

2.4 Multi-Instance Learning

Zoals eerder beschreven in "2.2 Artificial Neural Network (ANN)" is de meest gebruikte trainingsmethode van ANNs de supervised training, waarbij de verwachte uitkomst wordt meegegeven aan elke input in het systeem. Dit kan vergeleken worden met een docent die bij een voorbeeld het antwoord al geeft. Er zijn echter momenten waarop deze manier faalt, voornamelijk als objecten en/of instanties alleen collectief te labelen zijn en niet individueel.

Maron en Ratan (1998) beschrijven een voorbeeld van een afbeelding die een waterval toont. Welk onderdeel zorgt ervoor dat de afbeelding geclassificeerd wordt als een waterval? Verschil-

[1] $a * b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + ... + a_n b_n$ waarbij $a = [a_1, a_2, ..., a_n]$ en $b = [b_1, b_2, ..., b_n]$

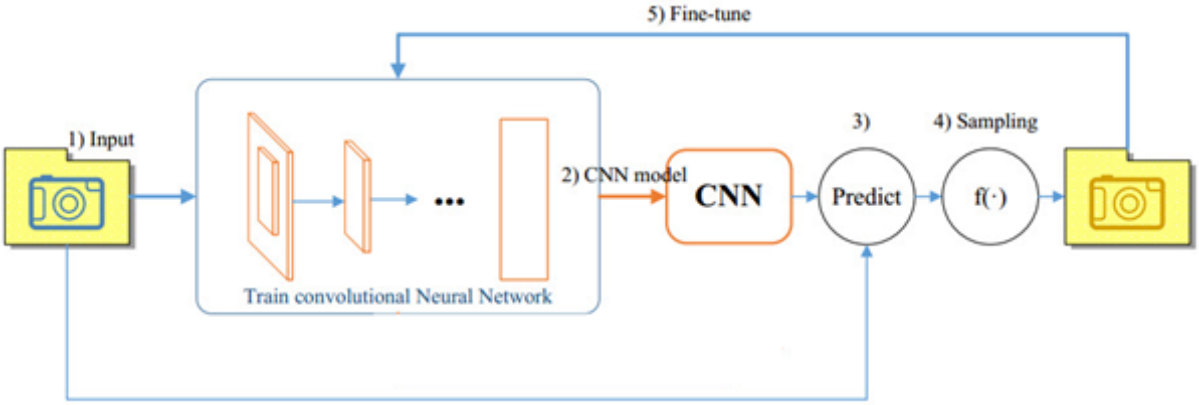
[2] Dit is gelijk aan de functie $f(x) = max(0, x)$

lende onderdelen van de afbeelding kunnen hiervoor verantwoordelijk zijn. Zij benoemen als voorbeelden: de bloemen die in bloei zijn, de vogels die rondvliegen of de witte stroom van het water. Er kan dus geprobeerd worden om overeenkomsten te vinden tussen de verschillende afbeeldingen die gelabeld worden als waterval en die afwezig zijn in afbeeldingen die niet gelabeld worden als waterval (Maron & Ratan, 1998).

Multi-Instance Learning is een manier om het systeem te leren welk object/instantie de collectie onderscheidt en dus categoriseert/labelt (Maron & Ratan, 1998). Bij Multi-Instance learning wordt er een set van "bags" (bijvoorbeeld afbeeldingen) als input gegeven. Bij elke "bag" wordt er aangegeven of deze positief of negatief is. Elke "bag" bevat meerder "instances" (objecten in de afbeelding, zoals bloemen, gras, water, etc.). Een "bag" is negatief als alle "instances" negatief zijn en een bag is positief als één of meerdere "instances" positief zijn. Uit een set van gelabelde "bags" zal het systeem proberen om overeenkomsten te vinden die ervoor zorgen dat onbekende "bags" gelabeld kunnen worden (Maron & Ratan, 1998).

Het vinden van verschillende objecten/instanties is moeilijk, omdat de ratio tussen positieve- en negatieve "instances" in een positief gelabelde "bags" (het ruis level) oneindig groot kan zijn. Zoals reeds vermeld, dient slechts één "instance" in de "bags" positief te zijn om als positief gelabeld te worden. Echter kunnen er dan nog steeds oneindig veel negatieve "instances" voorkomen in een "bag". Welke "instance" is nu verantwoordelijk voor de positieve label? Is het ook mogelijk dat de label in verband wordt gebracht met juist een negatieve "instance"? Bijvoorbeeld, omdat alle foto's dezelfde achtergrondstructuur hebben?

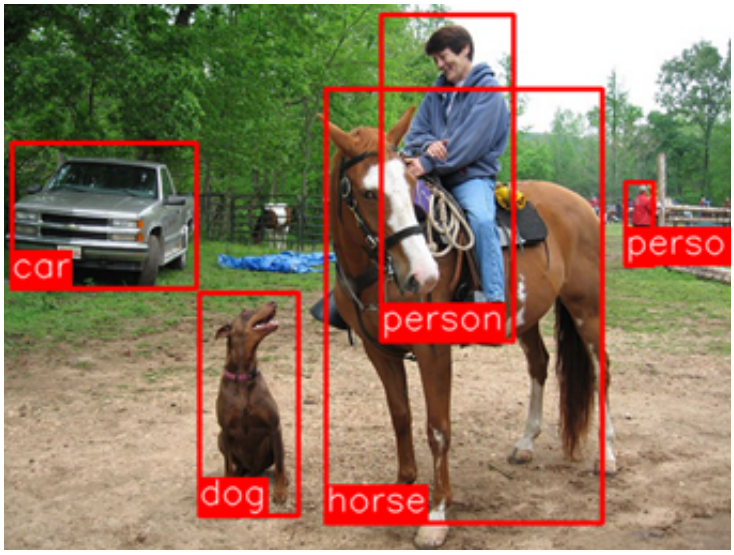
2.5 Belangrijkste concepten in samenhang



Figuur 2.4: De wijze waarop een CNN wordt getraind

ANNs, en dan specifiek de alternatieve vorm CNNs, zijn uitermate geschikt om afbeeldingen te classificeren (Gupta & Aggerwal, 2010). Een ANN, daardoor ook een CNN, dient getraind te worden. De bovenstaande afbeelding geeft goed weer hoe een CNN getraind wordt. Een afbeelding gaat het model in en het systeem zal voorspellen in welke categorie de afbeelding behoort. Indien de categorie niet overeen komt met de label dat meegegeven is, zal het systeem zichzelf finetunen zodat de voorspelling dichter in de buurt komt van de gegeven klasse. Dit is de wijze waarop "supervised learning" werkt, welke gebruikt zal worden in dit onderzoek. Als tijdens de training genoeg verschillende afbeeldingen aan bod gekomen zijn, is het CNN in staat om een klasse te voorspellen bij een onbekende afbeelding.

Door het trainen leert een CNN patronen te herkennen in een afbeelding, behorend bij een bepaalde categorie. Het wordt echter lastig als een afbeelding meerdere categorieën bevat. Het is voor een CNN namelijk niet mogelijk om meerdere categorieën aan een afbeelding te koppelen.



Figuur 2.5: Objectdetectie

Om toch meerdere categorieën aan een afbeelding te koppelen kan "multi-instance learning" gebruikt worden. Door gebruik te maken van "multi-instance learning" in combinatie met een CNN, kunnen de verschillende objecten in een afbeelding aan de juiste categorie gekoppeld worden. Dit wordt ook wel "objectdetectie" genoemd. In figuur 2.5 is duidelijk weergegeven hoe "objectdetectie" werkt.

3 Methoden

Dit hoofdstuk beschrijft de aanpak van het onderzoek en de daarbij gebruikte methodes.. Daarnaast zal er beschreven worden welke data er verwacht wordt en hoe deze geanalyseerd kan worden. Het onderzoek zal, gedurende de periode van 29 januari 2018 tot en met 29 juni 2018, uitgevoerd worden bij Quintor, locatie Den Haag. Het onderzoek wordt gedaan namens Quintor in opdracht van ToolMax.

3.1 Deelvraag 1

Om het huidige systeem te verbeteren, zal de volgende vraag beantwoord moeten worden: "Aan welke eisen moet het nieuwe systeem minimaal voldoen, wil het automatische productclassificatie voor ToolMax mogelijk maken". Deze vraag wordt beantwoord door middel van kwalitatieve methodes in de vorm van een semigestructureerd interview. Met behulp van een semigestructureerd interview met de eigenaar van ToolMax, zullen de minimumeisen bekend worden waaraan het systeem moet voldoen, wil het automatische productclassificatie mogelijk maken. Er is gekozen voor een semigestructureerd interview zodat de mogelijkheid bestaat om door te vragen indien antwoorden onduidelijk of onvolledig zijn. Nadat het interview afgenomen is zal deze getranscribeerd en gecodeerd worden. Door het coderen van de gegeven antwoorden kunnen er onderligende concepten of thema's in de data en de onderliggende relaties ontdekt worden (Gorden, 1998). De interviewvragen staan weergegeven in: "C Interviewvragen - requirements"

Er zullen verschillende acceptatiecriteria uit het interview volgen, waaraan het uiteindelijke systeem moet voldoen om het huidige systeem te verbeteren. Naast de verschillende acceptatiecriteria waaraan het systeem moet voldoen, moet het systeem ook betrouwbaar zijn. Dit om automatische productclassificatie mogelijk te maken. Tijdens het semigestructureerd interview zal dan ook getracht worden te ontdekken wanneer een systeem betrouwbaar genoeg is om deze te gebruiken voor automatische productclassificatie.

Via exploratief onderzoek dat uitgevoerd wordt in de vorm van deskresearch, zal er gekeken worden welke eisen andere E-commerce bedrijven stellen aan productclassificatie. Hierbij is gekozen voor deskresearch, omwille van het feit dat dit zich uitstekend leent om globaal beeld te krijgen van de verschillende eisen die andere E-commerce bedrijven stellen. Op deze manier kunnen de acceptatiecriteria aangevuld worden met industriestandaarden en de mogelijke problemen die volgen vanuit het flowdiagram, welke is opgesteld tijdens het vooronderzoek[1].

De acceptatiecriteria zullen duidelijke eisen/requirements stellen aan de inrichting van de zelflerende machine. Tevens vormen zij een belangrijke basis voor de latere requirements van de zelflerende machine en het systeem in de praktijk. De acceptatiecriteria zullen tot slot gevalideerd worden door eisen/requirements te overleggen met de opdrachtgever. Daarnaast worden de verschillende eisen/requirements samen met de opdrachtgever geprioriteerd. Hiermee wordt zeker gesteld dat de verschillende eisen/requirements voldoen aan de verwachtingen van de opdrachtgever.

3.2 Deelvraag 2

Om te onderzoeken hoe een zelflerende machine ingericht kan worden zodat het producten kan classificeren, moet er gekeken worden naar de verschillende inrichtingen die mogelijk zijn voor een zelflerende machine. Hiervoor wordt exploratief onderzoek uitgevoerd in de vorm van deskresearch.

[1] zie "B Classificatieproces ToolMax"

Er is gekozen voor deskresearch omdat dat zich uitstekend leent om een globaal beeld te krijgen van de verschillende bestaande modellen, die mogelijk te gebruiken zijn om producten te classificeren. Daarbij wordt er gekeken naar modellen die het ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)[2] gewonnen hebben, omdat deze modellen bewezen hebben dat ze werken. De ILSVRC bestaat al sinds 2010 (Russakovsky et al., 2015). Doordat er een grote hoeveelheid modellen zijn die meegedaan hebben met de ILSVRC zullen slechts deze gekozen worden die innovaties toevoegen aan de theorie achter CNN's. Verder zal er gekeken worden naar recente modellen die problemen oplossen in de modellen uit de ILSVRC.

Om de gevonden modellen met elkaar te kunnen vergelijken, is er voor gekozen om een verschillende experimenten, in de vorm van laboratoriumonderzoek, toe te passen. Er is gekozen voor een laboratoriumonderzoek, omdat hierdoor zoveel mogelijk externe factoren, welke invloed kunnen hebben op de onderzoeksresultaten, uitgesloten worden. De verschillende resultaten zijn daardoor niet afhankelijk van externe factoren en kunnen objectief met elkaar vergeleken worden.

Voor het laboratoriumonderzoek zal er een testframework ontwikkeld worden met behulp van de library genaamd Tensorflow (TensorFlow, z. j.)[3]. De modellen worden in dit testframework getraind en met elkaar vergeleken. Het testframework zorgt ervoor dat al deze acties voor elk model hetzelfde zijn. De dataset die gebruikt wordt voor het trainen zal van de site: www.ToolMax.nl onttrokken worden, door middel van een zelf ontwikkelde web scraper. Op deze manier is de gebruikte dataset representatief voor de classificaties die het systeem uiteindelijk zal uitvoeren.

De resultaten van de verschillende experimenten worden gebruikt om de zelflerende machine zo optimaal mogelijk te ontwikkelen. Dit kan ontwikkeld worden door gebruik te maken van zowel een enkel model als een combinatie van verschillende modellen. De inrichting moet voldoen aan de acceptatiecriteria die zijn opgesteld in "3.2 Deelvraag 2".

Uiteindelijk zal de inrichting gevisualiseerd worden, zodat er in een oogopslag te zien is hoe de zelflerende machine functioneert. Hierbij is ook een beschrijving van de zelflerende machine toegevoegd, zodat deze gemakkelijk toe te passen is in een daadwerkelijk systeem dat de producten gaat classificeren.

3.3 Deelvraag 3

Door middel van toegepast onderzoek zal getracht worden om de zelflerende machine, waarvan de inrichting bepaald is in de vorige deelvraag, toe te passen in de praktijk werkzame CNN. Dit wil zeggen dat er gekeken wordt of het mogelijk is om een interface te maken, die ervoor zorgt dat de zelflerende machine op gebruikersvriendelijke manier gebruikt kan worden door de medewerkers van ToolMax, die verantwoordelijk zijn voor de classificatie. Hierbij wordt er dan ook gekeken of de Proof of Concept (PoC) voldoet aan de gestelde eisen. Dit zal wordt gedaan aan de hand van toegepast onderzoek, in combinatie met veldonderzoek.

De PoC wordt ontwikkeld aan de hand van de Ontwikkeling, Test, Acceptatie en Productie (OTAP)-methodiek (Capgemini, 2014). De OTAP-methodiek leent zich om te kijken of de PoC de beoogde resultaten behaalt. Aangezien de PoC tijdens ontwikkeling, testing, acceptatie en productie getest wordt, waarbij de acceptatie- en productietests een realistische praktijkomgeving nabootsen. De acceptatie- en productietests zullen dus de verschillende criteria uit "3.2

[2]Volgens Russakovsky et al. (2015) is de ILSVRC ontworpen om: "alogritme voor objectdetectie en beeldsclassificatie te evalueren op grote schaal. Daarnaast stelt het onderzoekers in staat om de voortgang van afbeelding classificatie over een breder scala aan objecten te vergelijken"

[3]Uit een pakketselectie (zie: "D Pakketselectie") is gebleken dat deze library het meest geschikt is voor de doeleinden die horen bij dit onderzoek.

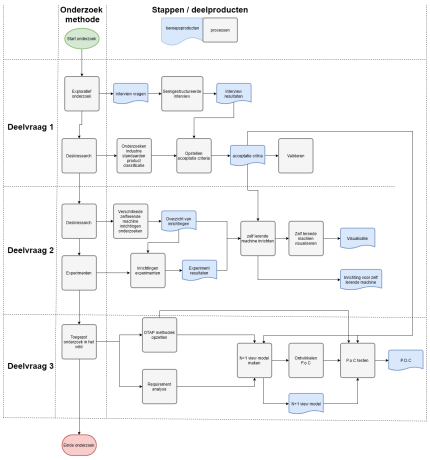
Deelvraag 2" testen, in een omgeving die gelijkstaat aan de praktijk. Een PoC kan gezien worden als een prototype. Als de PoC goed werkt in de test-, acceptatie- en productieomgeving, kan er geconcludeerd worden dat het ook in de praktijk werkt. Hierdoor kan er doormiddel van de OTAP methodiek een inschatting gemaakt worden of de PoC te gebruiken is in de praktijk.

Om de PoC te ontwikkelen zal er een "requirementsanalyse" gedaan worden, waarbij de verschillende eisen van het systeem worden gespecificeerd. Deze eisen worden vervolgens gevalideerd door de Lead Developer van Quintor. Samen met de acceptatiecriteria, welke geformuleerd zijn in "3.2 Deelvraag 2", wordt er een skelet gedefinieerd van het PoC. Om ervoor te zorgen dat de PoC reproduceerbaar is voor eventueel verder onderzoek, wordt het skelet uitgeschreven in een N+1 view model (Kruntchen, 1995). Door dit view model wordt ervoor gezorgd dat de verschillende specialisten binnen de IT duidelijk weten hoe het PoC eruit ziet.

Een PoC dat voldoet aan de criteria voorwaarden uit "3.2 Deelvraag 2" en welke succesvol de testen in een praktijkomgeving heeft doorlopen, zal uiteindelijk het resultaat zijn.

3.4 Totaalbeeld

Om te weten te komen of een zelflerende machine überhaupt de huidige manier van werken kan verbeteren, moet er gekeken worden naar welke verbeteringen het systeem minimaal moet halen wil het rendabel zijn. Door een interview te houden, zal geprobeerd worden hierachter te komen. Via dit interview kunnen er verschillende acceptatiecriteria opgesteld worden, waaraan het systeem uiteindelijk moet voldoen om de huidige manier van classificeren bij ToolMax te verbeteren.



Figuur 3.1: Totaalbeeld van het onderzoek, zie "E Totaalbeeld" voor uitvergroting

Een zelflerend systeem is een groot en breed begrip, waarbij er veel verschillende manieren zijn om het systeem in te richten. Welke inrichting van een zelflerend systeem het beste is voor het classificeren van afbeeldingen, zal dan ook uitgezocht moeten worden. Door met verschillende inrichtingen te experimenteren kan er uiteindelijk een enkele of een samenstelling van verschillende inrichtingen gekozen worden.

Tot slot zal bekeken worden hoe het zelflerend systeem in de praktijk toegepast kan worden en/of het de beoogde resultaten behaalt. Dit zal gedaan worden door het PoC te testen in natuurlijke omgevingen.

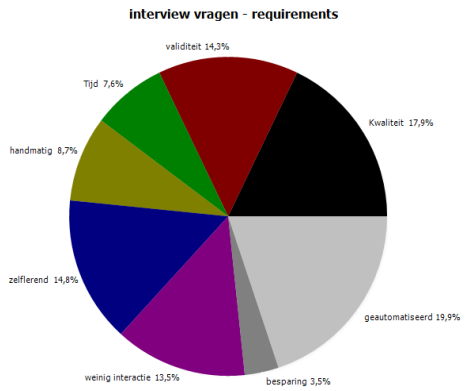
4 Resultaten

In dit hoofdstuk worden de uitkomsten van het onderzoek per deelvraag beschreven, waarbij de belangrijkste uitkomsten kort opgesomd worden. In dit hoofdstuk worden geen conclusies getrokken, deze volgen in hoofdstuk “5 Conclusie”. In hoofdstuk “6 Discussie” zullen de conclusies en resultaten bediscussieerd worden.

4.1 Deelvraag 1

4.1.1 Interview

Om de eisen waaraan het systeem moet voldoen te achterhalen, is op 12-03-2018 de eigenaar van ToolMax geïnterviewd[1]. De verkregen antwoorden zijn gecodeerd, zodat naast de eisen ook andere onderliggende onderwerpen duidelijk worden (zie figuur 4.1). Met deze onderliggende onderwerpen, wordt er getracht de eisen van ToolMax specifieker te maken. Op deze manier kunnen de antwoorden gemakkelijker omgezet worden naar requirements.



Figuur 4.1: Code frequentie requirement interview

In figuur 4.1[2] zijn de verschillende codes c.q. onderwerpen weergegeven. In figuur 4.1 is duidelijk te zien dat ToolMax veel waarde hecht aan automatisering. Kwaliteit en validiteit komen hierbij respectievelijk op de tweede en derde plaats. Deze top drie, aangevuld met de aspecten zelflerend en weinig interactie, sluit aan bij wat er in het vooronderzoek[3] naar voren is gekomen. Namelijk dat: “ToolMax wenst de kwaliteit van ToolMax.nl dusdanig te verbeteren zonder hiervoor meer mankracht in te zetten”

In tabel 4.1 is de top vijf onderwerpen weergegeven. Uit deze top vijf kunnen enkele belangrijke aspecten opgenomen worden:

- Geautomatiseerd: zo min mogelijk interactie van de werknemers;
- Voorspelbaar: het systeem moet aangeven hoe betrouwbaar zijn voorspelling i;
- Voorspelbaar: dit wil zeggen dat het systeem moet aangeven hoe betrouwbaar zijn voorspelling is;
- Reproduceerbaar: de classificatie voor een afbeelding moet constant hetzelfde zijn;
- Zelflerend: het systeem moet leren van de mogelijke foute voorspellingen.

[1] De interview antwoorden kunnen teruggevonden worden in “F Interview antwoorden - requirements”
[2] Uitvergrootte figuur kan teruggevonden worden in “G Code frequentie”
[3] Zie “A Interview antwoorden - vooronderzoek”

Deze vijf punten kunnen als volgt worden samengevat: *"Het systeem moet van zijn fouten kunnen leren en gaandeweg steeds betere resultaten produceren. Deze resultaten moeten gevalideerd, gecontroleerd en gereproduceerd kunnen worden, waarbij de interactie met de medewerkers minimaal is"*. In sectie "4.1.2 Eisen" zullen de eisen van het systeem beschreven worden.

Onderwerp	*frequentie aantal woorden*
Geautomatiseerd	19,9%
Kwaliteit	17,9%
Validiteit	14,3%
Zelflerend	14,8%
Weinig interactie	13,5%
Totaal	*80,4%*

Tabel 4.1: Overzicht van de verschillende onderwerpen

4.1.2 Eisen

De verschillende eisen die de eigenaar van ToolMax aangeeft in het interview zijn als volgt:

- Wanneer de nauwkeurigheid van een classificatie minder is dan 70%, moet deze handmatig geclassificeerd worden;
- Input van medewerkers moet gebruikt worden om betere classificaties te produceren;
- Het systeem moet op basis van twee methodes leren:
  - Geautomatiseerd;
  - Handmatig.
- De handmatige methode moet na een aantal keer automatisch gaan;
- Medewerkers moeten zo min mogelijk betrokken zijn bij het proces;
- Compute power is onbelangrijk;
- De tijd dat het kost om te leren is onbelangrijk.

4.1.3 Requirements

De eisen van ToolMax, samen met de top 5 aspecten uit "4.1.1 Interview", kunnen omgezet worden naar "harde" Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdgebonden (SMART) requirements, die door middel van het **M**o**SC**o**W** principe gesorteerd zijn.

Nummer	*Omschrijving*	*Prioritering*
F01	Het systeem moet automatisch kunnen classificeren.	M
F02	Het systeem moet op basis van twee manieren leren, d.m.v. automatisering o.b.v. de categorie van de afbeelding en d.m.v. wat de medewerker aangeeft.	M
F03	Het systeem moet de interactie met medewerkers tot het minimum te beperken.	M
F04	De resultaten van het systeem moeten reproduceerbaar zijn.	M
F05	Het systeem moet bij een classificatie een zekerheidspercentage geven.	M
F06	Het systeem moet bij een zekerheidspercentage van minder dan 70% overgaan op handmatige classificatie.	S
F07	Het systeem moet, nadat handmatige classificatie enkele keren is voorgekomen, classificaties automatisch uitvoeren.	S

F08	Het systeem moet gecorrigeerd kunnen worden door medewerkers, waarna het systeem leert van deze correctie.	S

Tabel 4.2: Requirements zelflerende systeem ToolMax

4.1.4 Betrouwbaarheid

Uit het interview met de eigenaar van ToolMax is naar voor gekomen dat er gekeken moet worden naar de testnauwkeurigheid, om de betrouwbaarheid van het netwerk te bepalen. Daarnaast wordt er ook gekeken of de top categorie (categorie met de hoogste voorspelling) gelijk is aan de juiste categorie, dit wordt de top 1 score genoemd. Buiten enkel te kijken naar de top 1 score, kan er ook gekeken worden of de juiste categorie in de top 3 of top 5 voorspellingen zit. De top 3 en top 5 geven beter weer hoe het neuraal netwerk presteert, doordat de afbeeldingen overeen kunnen komen met meerdere categorieën (Kostyaev, 2016). Om te zorgen dat de systemen betrouwbaar zijn en goede resultaten halen, is er samen met ToolMax besloten om de volgende requirements, met betrekking tot de betrouwbaarheid, op te nemen:

Nummer	*Omschrijving*	*Prioritering*
F09	Het systeem moet minstens een test nauwkeurigheid behalen van 90%	M
F10	Het systeem moet een top 1 error hebben van minder dan 50%	M
F11	Het systeem moet een top 3 error hebben van minder dan 25%	M
F12	Het systeem moet een top 5 error hebben van minder dan 12,5%	M

Tabel 4.3: Requirements zelflerende systeem ToolMax

De verschillende requirements die met ToolMax opgesteld zijn, worden deels gebruikt om het juiste model te selecteren en deels als requirements voor het systeem in de praktijk.

De requirements uit tabel 4.3 zorgen ervoor dat het systeem voldoet aan de eisen van ToolMax, deze requirements kunnen uitgebreid worden door te kijken naar andere e-commerce bedrijven.

E-commerce bedrijven

Om de de requirements uit te breiden en te verbeteren, kan er gekeken worden naar andere e-commerce bedrijven en de manieren waarop deze werken. Hiervoor is er gekeken naar het grootste e-commerce bedrijf van Nederland, namelijk Bol.com. Bol.com was volgens van Oosterhout, Verdonk en Scheffer (2017), met een omzet van 950 miljoen euro, de grootste online retailer van Nederland in 2017. Met meer dan 20 productcategorieën en talloze subcategorieën is het juist classificeren van producten van groot belang om vindbaarheid van producten te waarbogen (Granada Research, 2001). Binnen Bol.com is dhr. R de Dijcker werkzaam als bemiddelaar tussen externe verkopers en ontevreden klanten, hierbij heeft Dhr. de Dijcker een goed inzicht in de werking van het classificatieproces bij Bol.com. Dhr. de Dijcker heeft geholpen met inzicht te verkrijgen in het classificatieproces bij Bol.com, het classificatieproces werkt als volgt:

1. Indien een wederverkoper het product verkoopt, zal deze het product aan de juiste categorie moeten koppelen;

2. Indien Bol.com het product verkoopt:
   (a) Is het EAN bekend in de database, dan wordt het product door middel van zijn EAN gekoppeld;
   (b) Is het EAN onbekend, dan zal getracht worden om de juiste categorie via het internet te vinden.

Kijkend naar het classificatieproces van Bol.com kan tabel 4.3 aangevuld worden met de volgende requirement: "*F12: Wanneer het systeem producten tegenkomt met een EAN dat reeds geclassificeerd is, dan zal het deze automatisch als de betreffende categorie classificeren*". Deze requirement wordt aangeduid met een "w" (won't), aangezien dit buiten de scope van het project valt.

Een totaaloverzicht van alle requirements, gesorteerd d.m.v. het **M**o**SC**o**W** principe, kan terug gevonden worden in "H Requirements".

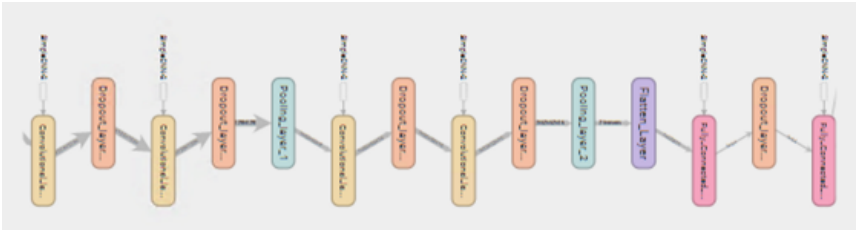
4.2 Deelvraag 2

4.2.1 Modellen

In het onderzoek zijn er vijf verschillende modellen bekeken. De modellen zijn gekozen omdat deze innovaties toevoegen aan de theorie achter CNNs en omdat ze bewezen hebben dat ze werken d.m.v. de ILSVRC. De verschillende modellen die gekozen zijn om te testen, zijn:

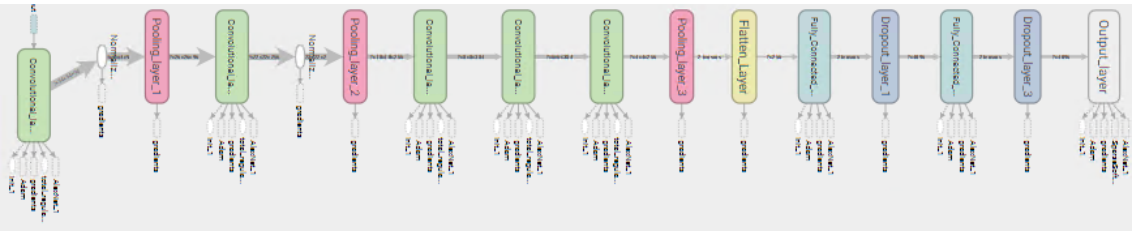
- SimpleCNN;
- AlexNet;
- GoogLeNet;
- MobileNetV2;
- DenseNet.

SimpleCNN is een model dat zelf ontwikkeld is. Dit model past verschillende bevindingen over CNNs toe, zoals: *"Xaviar intilizing"*, *"biases"*, *"adaptive learning"*, *"dropout"*, etc. Dit model heeft slechts zes lagen en kan daardoor, zoals de naam al aangeeft, beschouwd worden als relatief eenvoudig. Figuur 4.2 geeft SimpleCNN schematisch weer.



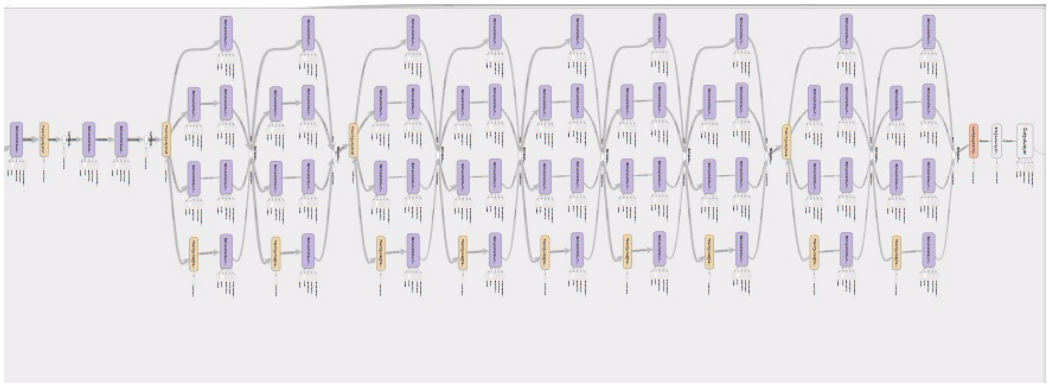
Figuur 4.2: SimpleCNN model

AlexNet is ontwikkeld in 2012 en wordt beschouwd als het CNN waarmee alles begon. In 2012 won het de ILSVRC. De ILSVRC kan beschouwd worden als de olympische spelen voor de beeldverwerking. Volgens Krizhevsky, Sutskever en Hinton (2012) behaalde AlexNet hierbij een top 5 error van slechts 15,3%, terwijl de tweede plaats een top 5 error van 26,2% behaalde. AlexNet gebruikt een relatief simpele vorm met slechts acht lagen. Door gebruik te maken van dropout en ReLu kon de trainingstijd verminderd worden en overfitting vermeden worden. Figuur 4.3 geeft AlexNet schematisch weer.



Figuur 4.3: AlexNet model

GoogLeNet is ontwikkeld in 2014 door Google (Szegedy et al., 2014). In dat jaar won het ook de ILSVRC met een top vijf error van 6,7%. GoogLeNet was een van de eerste modellen dat het idee introduceerde dat verschillende CNNs niet enkel op elkaar gestapeld hoefden te worden. Szegedy et al. (2014) bedachten de zogenaamde "inception module", dit zijn modules waarbij verschillende lagen parallel aan elkaar taken uitvoeren. Volgens Szegedy et al. (2014) zorgen deze "inception modules" ervoor dat het model dieper wordt (22 lagen) zonder dat het extra parameters genereert [4](GoogLeNet heeft 15 keer minder parameters dan AlexNet). Daarnaast zorgen de "inception modules" ervoor dat er minder rekenkracht nodig is om het netwerk te trainen, in vergelijking met conventionele diepere netwerken, aldus Szegedy et al. (2014). Figuur 4.4 geeft GoogLeNet schematisch weer.

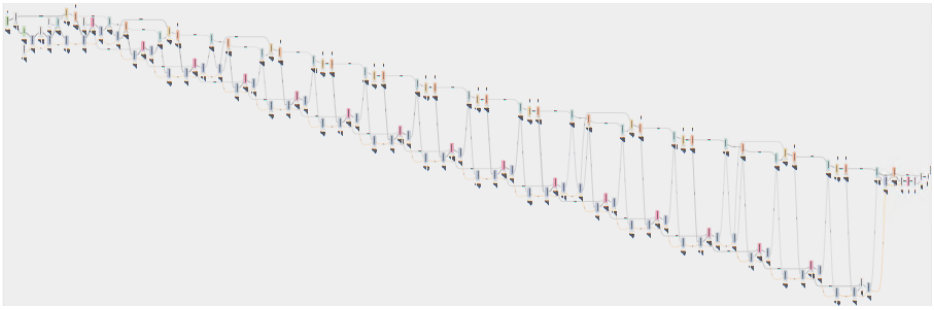


Figuur 4.4: GoogLeNet model

MobileNetV2 is ontwikkeld in het begin van 2018 met efficiëntie van resources in gedachte (Sandler, Howard, Zhu, Zhmoginov & Chen, 2018). Sandler et al. (2018) geeft aan dat MobileNetV2 specifiek ontwikkeld is voor mobile of resource beperkende omgevingen, door gebruik te maken van een nieuw type laag genaamd "inverted residual with linear bottleneck[5]". Volgens Sandler et al. (2018) kan deze laag met simple operaties geïmplementeerd worden, waardoor er minder parameters nodig zijn en het model veel minder resources nodig heeft. Figuur 4.5 geeft MobileNetV2 schematisch weer.

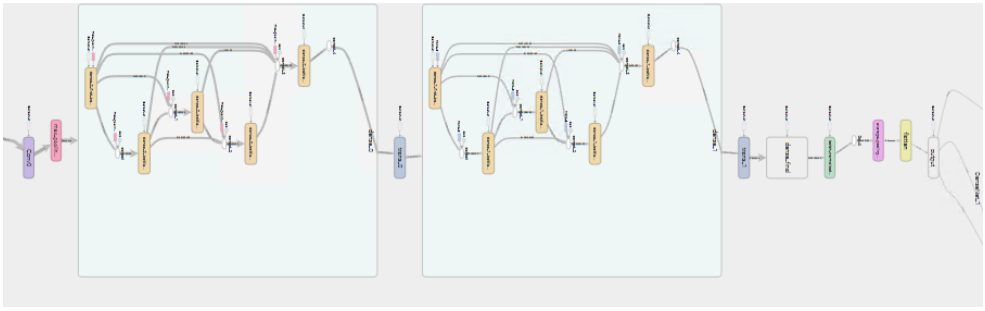
[4]Hoe meer parameters, hoe groter de kans op overfitting.

[5]Deze laag neemt als invoer een laag-dimensionaal gecomprimeerde afbeelding, welke eerst wordt uitgebreid naar een hoge dimensie. Hierna wordt de afbeelding gefilterd met een lichtgewicht diepteconvolutie. Deze afbeeldingen worden vervolgens terug geprojecteerd naar de laagdimensionale invoer(Sandler et al., 2018).



Figuur 4.5: MobileNetV2 model

DenseNet is ontwikkeld in 2016 als reactie op een nieuw opkomend probleem. Doordat netwerken steeds groter werden, kon het zijn dat informatie over de input gaandeweg verdween naarmate het het einde naderde (Huang, Liu, van der Maaten & Weinberger, 2016). Huang et al. (2016) lost dit probleem op, door alle lagen[6] met elkaar te verbinden, waardoor de informatie over de input via verschillende wegen het einde kan bereiken. Dit zorgt ervoor dat het netwerk zeer weinig parameters heeft en overfitting deels vermeden kan worden. Figuur 4.6 geeft DenseNet schematisch weer.



Figuur 4.6: DenseNet model

4.2.2 Test framework

Voor het testen en trainen van verschillende CNN modellen is er een testframework[7] ontwikkeld. Dit testframework is verantwoordelijk voor het testen en trainen van de verschillende modellen. Wanneer het model getraind en getest is, kunnen er door middel van het framework ook voorspellingen gedaan worden over het model. Naast trainen, testen en voorspellen neemt het framework ook het presenteren van data en het opslaan van het model voor zijn rekening.

Het trainen, testen en voorspellen kan geïnitialiseerd worden door middel van een Command Line Interface (CLI) via het volgende commando:

```
$ python dnn.py --mode=<modus> --model=<model> --verbose<boolean>
```

Hierbij heeft de parameter “mode” drie opties, namelijk:

- “train”, het gekozen model wordt getraind;
- “test”, het gekozen model zal getest worden;
- “predict”, met behulp van het gekozen model wordt een classificatie gedaan op een ingevoerde afbeelding.

[6] Het model bestaat uit dense blokken die opgevolgd worden door transition layers.
[7] Zie bijlage “I Class diagram test framework” voor het class diagram

Op het moment van schrijven had het “model” parameter vijf mogelijke opties (welke overeen komt de onderzochte netwerken), namelijk:

- SimpleCNN;
- AlexNet;
- GoogLeNet;
- MobileNetV2;
- DenseNet.

De “verbose” parameter heeft slechts twee opties, namelijk: “True” of “False”.

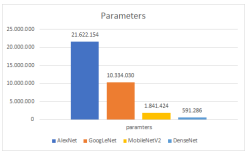
DDe verschillende modellen die gekozen worden bij de “model” parameter, worden in sectie “4.2.1 Modellen” toegelicht.

4.2.3 Het netwerk

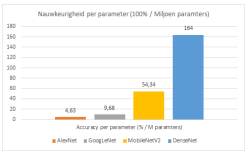
Figuur 4.7: Top error waarde per model

De verschillende modellen zijn net zolang getraind, totdat deze een test nauwkeurigheid van meer dan 90% hadden. Hierna zijn er met de verschillende modellen, op 50 verschillende afbeeldingen[8], classificaties uitgevoerd om zodoende de top error waardes te berekenen. In bijlage “J Top error waardes” zijn per model en per afbeelding de classificatieresultaten opgenomen. In figuur 4.7 zijn de resultaten samengevat.

Uit figuur 4.7 kan afgeleid worden dat GoogLeNet en DenseNet het beste presteren, omdat GoogLeNet meer classificaties goed heeft met de top 1 voorspelling en de DenseNet met de top 3. AlexNet en SimpleCNN presteren het slechtste van de vijf onderzochte modellen.



(a) Parameters per model



(b) Nauwkeurigheid per parameter

Figuur 4.8: Zie “K Parameters per model” en “L Nauwkeurigheid per parameter” voor de uitvergroting

Als er gekeken wordt naar het aantal parameters (zie figuur 4.8a), dan presteert DenseNet het beste, gevolgd door MobileNetV2. MobileNetV2 presteert slechter dan DenseNet en GoogLeNet, hoe minder parameters een netwerk heeft hoe minder gevoelig deze is voor overfitting. Daarnaast kan er gekeken worden naar de nauwkeurigheid per parameter. De informatiedichtheid/nauwkeurigheid per parameter is een eenheid die aangeeft in hoeverre een netwerk potentie heeft om te leren. Met andere woorden de nauwkeurigheid per parameter geeft aan wat de potentiële leermogelijkheid is van een netwerk.

[8] De afbeeldingen zijn random van het internet gedownload en eerlijk verspreid over de verschillende klassen.

Op basis van de requirements beschreven in “H Requirements” samen met het bovenstaande, is DenseNet het meest geschikt voor de doeleinden van ToolMax. Aan requirement F06 wordt niet voldaan, met aanvullende trainging komt DenseNet ook uit op een top een error lager dan 50%. Dit doordat deze 15 maal minder parameters heeft dan de andere modellen en dus lang door kan leren zonder overfitting. DenseNet heeft een zeer grote informatiedichtheid (nauwkeurigheid per parameter), waardoor het een zeer hoog leerpotentie heeft. Dit maakt DenseNet het meest geschikt om te gebruiken voor de classificatieprobleem van ToolMax, DenseNet zal in de volgende deelvraag gebruikt worden als netwerk van keuze.

4.3 Deelvraag 3

In paragraaf “4.1 Deelvraag 1” zijn verschillende requirements opgesteld waaraan het systeem moet voldoen. In deze paragraaf wordt gekeken op welke manier een zelflerende machine ingezet kan worden om aan de eisen te voldoen en hoe aan dit systeem vorm gegeven kan worden. Het systeem om productafbeeldingen te classificeren bestaat uit twee delen, namelijk:

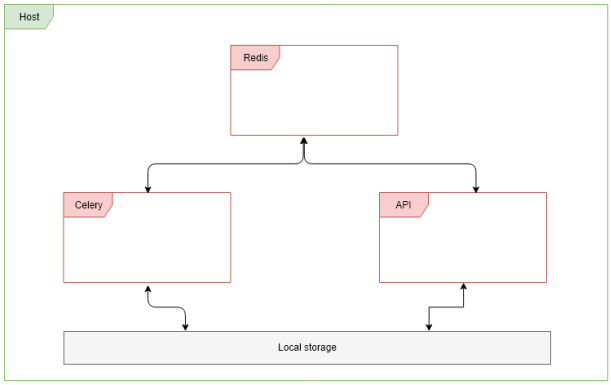
- Classificatie gedeelte;
- User Interface (UI)-gedeelte.

Als basis voor het classificatiegedeelte wordt het testframework uit paragraaf “4.2.2 Test framework” gebruikt. Het testframework bevat de verschillende modellen uit paragraaf “4.2.1 Modellen”, welke volledig getraind zijn. Het UI-gedeelte zal interacteren met de gebruiker en de opdrachten naar het classificatie gedeelte zenden. In bijlage “M N+1 view model” is het complete ontwerp van het systeem terug te vinden.

4.3.1 Classificatie gedeelte

Het classificatiegedeelte van het systeem is een Application Programming Interface (API) om het testframework heen. Door middel van deze API kan het UI-gedeelte met het testframework communiceren. De API bevat de volgende endpoints:

- **/models** hiermee worden alle beschikbare modellen opgevraagd;
- **/models/<model_name>** hiermee wordt informatie opgevraagd voor het betreffende model;
- **/models/<model_name>/<option>** hiermee wordt een taak gestart, er zijn drie taken: trainen, evalueren en voorspellen;
- **/status/<task_id>** hiermee kan de voortgang van een taak opgevraagd worden;
- **/data/<category_name>** hiermee wordt een afbeelding toegevoegd aan de trainingsdata.



Figuur 4.9: Schematische weergave van de docker opbouw

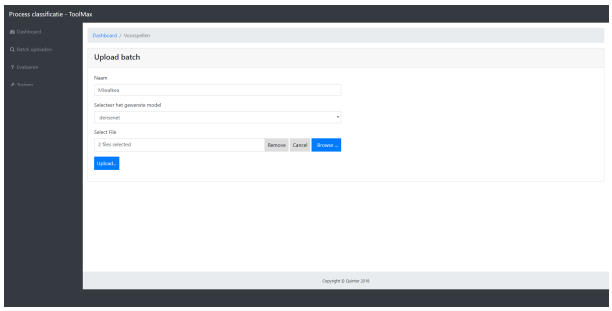
Om te zorgen dat de productclassificatie op elk mogelijke omgeving/situatie uit te voeren is, is de API ingepakt in een docker container. Deze docker bestaat uit 3 containers: een Redis container, een Celery container en een API-server container. De Redis container zorgt ervoor dat de Celery container en de API-server container met elkaar kunnen communiceren. De API-server stuurt taken naar de Celery container, welke deze dan asynchroon uitvoert. De docker container zorgt ervoor dat het productclassificatie gedeelte

makkelijk te distribueren is tussen verschillende systemen en applicaties.

4.3.2 UI gedeelte

Het UI gedeelte is verantwoordelijk voor de interactie met de gebruiker en de communicatie met de API. Daarnaast zal het UI gedeelte de geautomatiseerde processen bevatten om de productclassificatie te initialiseren.

Batch upload



Figuur 4.10: Upload interface (Uitvergroting zie: "N Upload interface")

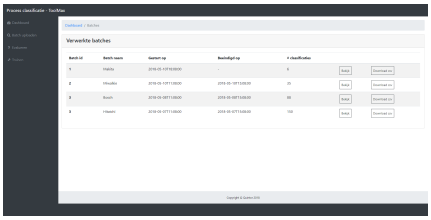
Om het classificatieproces in gang te zetten, moet een gebruiker een batch met afbeeldingen uploaden. In figuur 4.10 is de interface weergegeven waarin de gebruiker, de afbeeldingen die geclassificeerd moeten worden, kan uploaden. Hierbij geeft de gebruiker de batch een naam, selecteert deze een model en kan het de verschillende afbeeldingen selecteren.

Wanneer de afbeeldingen geüpload zijn, zal automatisch een proces geïnitialiseerd worden welke begint met het verwerken van de afbeeldingen.

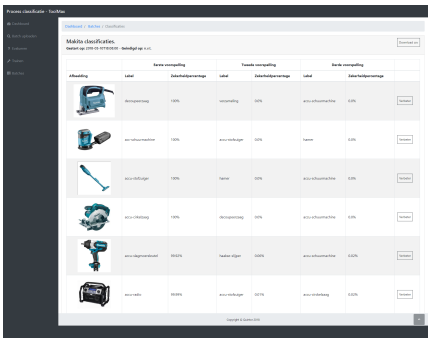
Het proces bestaat uit de volgende stappen:

1. Het systeem registreert dat er afbeeldingen zijn geüpload;
2. Het systeem genereert een URL waarop de voortgang gevolgd wordt;
3. Het systeem schaalt de afbeeldingen naar de gewenste groote (255 x 255 px);
4. Het systeem begint het classificatieproces:
   (a) Het systeem stuurt een de afbeelding naar de API;
   (b) Het systeem slaat het resultaat van de API op;
   (c) Het systeem begint weer bij stap (a) zolang er afbeeldingen zijn.
5. Het systeem slaat alle stappen en resultaten op.

Batch overzicht



(a) Overzicht van alle batches



(b) Overzicht classificaties in een batch

Figuur 4.11: Voor de uitvergrotingen zie: "O Overzicht batches" en "P Overzicht classificaties"

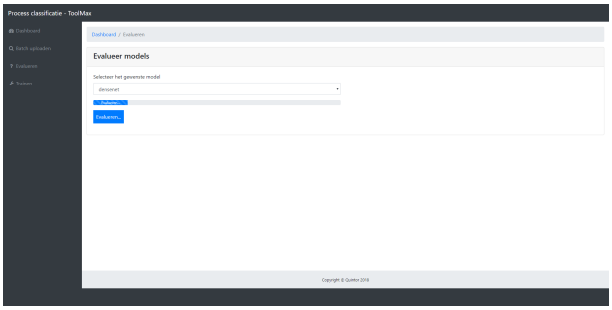
De resultaten van de classificatie kunnen bekeken worden in het batchoverzicht. Hierin staat een historie van de verschillende batches die verwerkt zijn (zie afbeelding 4.11a). De gebruiker

kan hier een .csv downloaden met daarin alle resultaten (deze .csv wordt ook gemaild naar de gebruiker). De gebruiker kan er ook voor kiezen om door te klikken naar de de resultaten van het classificatieproces. De gebruiker krijgt dan een overzicht te zien van alle afbeeldingen en de top 3 voorspellingen die erbij horen (zie figuur 4.11b).

In het overzicht van de verschillende classificaties kan een gebruiker ook aangeven wanneer een afbeelding onjuist geclassificeerd is. Dit doet een gebruiker door het aangeven van een andere classificatie. Indien een gebruiker aangeeft dat een classificatie incorrect is, zal het systeem de afbeelding toevoegen aan de training dataset, zodat het systeem hiervan kan leren.

Optimaliseren netwerk

Het systeem zal het zelflerende gedeelte van het systeem automatisch periodiek elke 2 weken laten trainen. Een gebruiker kan dit ook handmatig in gang zetten. Daarnaast kan een gebruiker ook de nauwkeurigheid van het zelflerende gedeelte controleren door het zelflerend gedeelte te evalueren. Op deze manier kan de gebruiker controleren of het systeem voldoet aan de gestelde eisen.



Figuur 4.12: Evalueren van het zelflerende gedeelte

5 Conclusie

Gedurende het onderzoek is er getracht een antwoord te vinden op de volgende vraag: “Op welke wijze kan het toepassen van een convolutional neural network de huidige manier van classificeren bij ToolMax verbeteren om automatische productclassificatie mogelijk te maken?” Naar aanleiding hiervan is een kwalitatief onderzoek uitgevoerd naar de verschillende eisen die aan het systeem gesteld worden, welke modellen hiervoor nodig zijn en hoe dat toegepast kan worden.

5.1 Deelvraag 1

In deelvraag 1 wordt er bekeken waaraan het systeem moet voldoen, wil het automatische productclassificatie mogelijk maken. Hieruit blijkt dat het systeem de kwaliteit van ToolMax.nl moet verbeteren, zonder dat hiervoor extra werknemers nodig zijn. Daarnaast moet het systeem een geautomatiseerd proces zijn die producten classificeert en waarbij de gebruiker de classificatie met de hand kan overschrijven. Het systeem moet hierbij aangeven wat de zekerheid is van de classificatie.

5.2 Deelvraag 2

In deelvraag 2 wordt er bekeken welke inrichting geschikt is voor productclassificatie op basis van voorafgaande eisen. Hieruit blijkt dat CNNs met weinig parameters een hoge potentie tot leren hebben en een kleinere kans hebben op overfitting. DenseNet is hierdoor met slechts een half miljoen parameters, t.o.v. 10 miljoen bij AlexNet, het beste systeem om te gebruiken voor productclassificatie. Deze kan continu getraind worden, waarbij de kans op overfitting minimaal is. DenseNet is dan ook het meest geschikt om productclassificaties uit te voeren, omdat het de minste parameters heeft en dus de minste kans op overfitting heeft.

5.3 Deelvraag 3

Tot slot wordt er bij deelvraag 3 bekeken hoe de zelflerende machine ingezet kan worden. Het blijkt dat door gebruik van een API het CNN eenvoudig aan te spreken en te distribueren is. Op deze manier kunnen de geleerde modellen verspreid worden en productclassificatie eenvoudig beschikbaar worden gesteld voor verschillende doeleinden. Via een interface kan een gebruiker gemakkelijk het classificatieproces in gang zetten, waardoor het voor de gebruiker slechts een druk op de knop is om de verschillende producten te classificeren.

5.4 Hoofdvraag

Zoals aangegeven in “4.7 Top error waarde per model”, heeft DenseNet een top 1 error van 54%. Dit betekent dat in 46% van de gevallen DenseNet het product exact[1] goed classificeert. Hierdoor kan in 46% van de gevallen het classificatieproces volledig overgenomen worden door het systeem. Een medewerker hoeft hierdoor slechts 54% van de producten zelf te classificeren.

Dankzij een neuraal netwerk kan het handmatige classificatiewerk met minstens 46% gereduceerd worden. Daarnaast zorgt een neuraal netwerk ervoor dat het classificeren de werknemer uit handen wordt genomen. Dit zorgt ervoor dat de medewerker zich kan focussen op de controlerende activiteiten en het inrichten van ToolMax.nl. Een classificatieproces is te initialiseren door het uploaden van de productafbeeldingen naar een webinterface. Doordat elk product één voor één geclassificeerd wordt, voorkomt het automatisch classificeren van producten het probleem dat verschillende classificaties op een hoop gegooid worden.[2] Dit zorgt er voor dat

[1] Correct classificeert met zijn eerste voorspelling

[2] Zie “1.1.1 Knelpunten ToolMax”

de kwaliteit van Toolmax.nl verbeterd wordt. Het is dus mogelijk om door middel van een CNN het classificeren van producten gedeeltelijk te automatiseren.

6 | Discussie

De verschillende eisen die aan het systeem gesteld worden, zijn opgesteld d.m.v. een interview met de eigenaar van ToolMax. De eisen die hieruit zijn gekomen, zijn samen met de lead developer geprioriteerd, waarna deze zijn teruggekoppeld aan ToolMax. Door deze terugkoppeling wordt ervoor gezorgd dat het systeem voldoet aan de verschillende eisen die de eigenaar van Toolmax heeft aangegeven.

De verschillende modellen zijn gekozen omwille van het feit dat ze nieuwe inzichten of aspecten toevoegen aan het kennisgebied over CNNs. Indien het onderzoek herhaald wordt kan het zijn dat er andere modellen geselecteerd worden op basis van de dan gestelde eisen. Echter wanneer dezelfde modellen geselecteerd worden zullen deze, na training op dezelfde dataset, altijd hetzelfde resultaat produceren. Dit zorgt ervoor dat de verschillende resultaten gebaseerd op het CNN valide zijn.

Om ervoor te zorgen dat het model niet specifieke kenmerken van een klasse leert[1] en dat de nauwkeurigheid verhoogd wordt, is er data augmentation (vermeerdering) toegepast (Wang & Perez, 2017). Wang en Perez (2017) geven daarnaast aan, dat data augmentation zorgt dat overfitting verminderd wordt. Data augmentation zorgt er in dit geval dus voor dat de resultaten die het netwerk produceert van betere kwaliteit zijn, gezien de hogere nauwkeurigheid en lagere kans op overfitting.

De oorspronkelijke trainingdataset bestaat uit 178 klassen met 342.484 afbeelding[2]. Het trainen van zo'n grote dataset kan weken duren, daarom is er voor gekozen om een subset te gebruiken. Deze subset bestaat uit 10 klassen met 10.980 bestanden. Werkt het CNN op de subset, dan zal deze ook werken op de gehele dataset. Het is hierdoor valide om een subset te gebruiken voor het onderzoek en mede daardoor tijd te besparen.

De spreiding van de trainingsdata vormt een grote impact op de nauwkeurigheid en prestatie van het netwerk (Hensman & Masko, 2015). Vaak is er oververtegenwoordiging van één klasse. Dit is een van de redenen dat er gekozen is voor een subset van de totale dataset. Hierdoor wordt de spreiding van de data over de klassen gebalanceerd. De resultaten van het CNN zijn daardoor nauwkeuriger en het CNN is minder geneigd om naar een specifieke classificatie te gaan. Dit maakt de resultaten betrouwbaarder. Er is voor gekozen om in deze subset productafbeeldingen te gebruiken die één classificatie kennen en dus slechts één object bevatten. Op deze manier wordt getracht fouten door ambiguïteit te voorkomen.

De verschillende netwerken dienen getraind te worden, waarbij het trainen van het netwerk op een grafische kaart dient te gebeuren. De huidige modellen die gebruikt worden om te classificeren, kunnen getraind worden op een GTX780 met 2048 MB geheugen. Hierbij kunnen maximaal 20 afbeeldingen, met een maximale grootte van 255x255 pixels, tegelijk gebruikt worden om het netwerk te trainen. Indien het netwerk sneller getraind moet worden, met grotere afbeeldingen, of er dienen grotere en complexere netwerken gebruikt te worden, dan zal er zwaardere hardware nodig zijn. De kosten van deze hardware[3] wegen echter niet op tegen de 46% werkbesparing die mogelijk gemaakt wordt door het CNN.

Het onderzoek behandelt niet de verschillen in performance tussen de verschillende CNN-modellen, het kijkt enkel naar de resultaten die de de verschillende modellen geven. Bij het trainen en het voorspellen van klassen is dus enkel gekeken naar de resultaten die geleverd worden, niet naar hoe lang dit duurt of hoeveel resources hiervoor nodig zijn. De performance per model is niet bepalend voor de haalbaarheid van de PoC. Van groter belang is de besparing die het oplevert voor ToolMax, oftewel de nauwkeurigheid van het model.

[1] Bijvoorbeeld: kleuren, posities, achtergronden etc

[2] Dit is na data augmentation, zonder data augmentation zijn het 38.053 afbeeldingen

[3] Op 15-05-2018 zat de prijs voor een GTX-1080 met 8192MB tussen de €669 en €729

De interface die ontwikkeld is voor de API, is goedgekeurd door Quintor en ToolMax. Voor de bevestiging is echter geen user test uitgevoerd. De resultaten van de interface zijn toch betrouwbaar aangezien ToolMax heeft aangegeven dat de interface datgene is wat ze voor ogen hadden. Tevens heeft Quintor bevestigd dat de verschillende views geschikt zijn voor de taken die het CNN uitvoert. Daarnaast lag de kern van het onderzoek in het machine learning aspect, de UI was een should have. In de PoC heeft de UI een dermate uitgebreide functionaliteit dat de opdrachtgever en klant tevreden zijn. Door het ontwikkelen van een separate API en UI, is de UI dermate eenvoudig aan te passen dat deze ondergeschikt is aan de resultaten van de modellen.

In het onderzoek is ervoor gekozen om de CNN-modellen te trainen tot deze een test score hebben van meer dan 90%. Deze keuze is gemaakt om de benodigde trainingstijd te reduceren, dit betekent niet dat de modellen dan klaar zijn met trainen . De resultaten en de nauwkeurigheid die gerealiseerd worden door de verschillende CNN-modellen zijn daardoor aan de lage kant[4]. De resultaten zijn echter goed genoeg om aan te tonen dat het model werkt en meer trainen de resultaten zal verbeteren, waardoor dit een valide methode is.

De belangrijkste beperkende factor binnen het onderzoek was de tijd, het trainen van CNN-modellen kost tijd. Elk model heeft ongeveer een week nodig gehad om te trainen. Dit is de reden dat ervoor gekozen is om slechts 5 modellen te testen in het onderzoek. Het framework is echter zo opgezet dat eenvoudig andere modellen toegevoegd kunnen worden, enkel een model beschrijving moet toegevoegd worden. Hierdoor kunnen er te allen tijde andere modellen getest worden

De meeste trainingstijd zit in de initiële training, ofwel het trainen van compleet nieuwe modellen. Wanneer deze modellen hertraind worden, zal de trainingstijd dankzij een reeds opgedane basis afnemen. Hierdoor zullen de reeds in het framework bestaande modellen een korte trainingstijd hebben.

[4] onder de 70%

7 | Aanbeveling

Uit onderzoek is gebleken dat de productclassificatie overgenomen kan worden door een neuraal netwerk, waarbij het een werkbesparing van 46% kan leveren. Om een optimaal resultaat en de hoogst mogelijke nauwkeurigheid te verkrijgen, zijn er 5 aanbevelingen die uit dit onderzoek volgen.

Langer trainen. Om een beter resultaat te verkrijgen is het aan te bevelen de verschillende modellen langer te laten trainen en de dataset uit te breiden met meer afbeeldingen. De afbeeldingen moeten per klasse beter verspreid zijn over het gehele netwerk, zodat de data evenredig over de klasses is verdeeld. Op deze manier kunnen er betere resultaten behaald worden en kan het meer werkbesparing opleveren omdat de modellen nauwkeuriger worden.

Resultaten verbeteren. Om hogere nauwkeurigheid te behalen (dus een betere top een error), luidt het advies om ook de metadata van de producten te gebruiken bij het classificeren. Dit zorgt ervoor dat het neurale netwerk meer aspecten heeft waarmee het een specifieke klasse kan herkennen. Als alternatief hiervoor kan ook Google Search gebruik worden. Google Search vormt dan een extra classificatiestap. De eigen resultaten kunnen dan vergeleken worden met de resultaten van Google Search, om zo de resultaten gaandeweg te verbeteren.

Externe server. Om een CNN te trainen is er een flinke hoeveelheid rekenkracht nodig, hierdoor is het aan te bevelen om het classificatieproces op een externe server uit te voeren i.p.v. in een Docker container. Via deze externe server kunnen leertijden geminimaliseerd en het leerproces gegarandeerd worden. Ook wanneer de dataset in de loop der tijd groeit, blijft het leerproces stabiel.

Controle. Aangezien er altijd foutpositieve resultaten aanwezig zullen zijn in de resultaten van het systeem, is het aan te bevelen om de resultaten altijd te controleren en niet blindelings op het resultaat af te gaan. Dit zorgt ervoor dat de kwaliteit van productclassificatie voor ToolMax.nl gewaarborgd kan worden en dat er geen grove fouten op de site optreden.

Objectdetectie. Een product kan uit meerdere objecten met verschillende classificaties bestaan, waardoor het is aan te bevelen om objectdetectie toe te passen. Het netwerk kan hierdoor meerdere classificaties per afbeelding uitvoeren (Ren, He, Girshick & Sun, 2015) en kan dan complexere afbeeldingen verwerken, zoals combodeals, acties of verzamelingen.

Het PoC dat ontwikkeld is heeft potentie, kijkend naar de “5 Conclusie” is het aan te bevelen om dit onderwerp verder te onderzoeken, waardoor dit mogelijk nog meer efficiëntie op kan leveren voor ToolMax

Literatuur

Benenson, R. (2016). *What is the class of this image ?* Op 2018-02-02 verkregen van http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#494c5356524332303132207461736b2031

Capgemini. (2014). *Scrummend de straat door.* Op 2018-02-13 verkregen van https://www.capgemini.com/nl-nl/2014/10/scrummend-de-straat-door/

Ebay Inc. (2018). *Who are we.* Op 2018-02-12 verkregen van https://www.ebayinc.com/our-company/who-we-are/

Gorden, R. (1998). Coding interview responses. *Basic Inteviewing Skills*, 180–198.

Granada Research. (2001). *Why Coding and Classifying Products is Critical to Success in Electronic Commerce (October 2001)* (Rapport).

Gupta, M. & Aggerwal, N. (2010). Classification techniques analysis. In *National conference on computational instrumentation* (pp. 128–131). Chandigarh, INDIA.

Hagan, M. T., Demuth, H. B. & Beale, M. H. (2014). Neural Network Design. *Boston Massachusetts PWS*, *2*, 734. doi: 10.1007/1-84628-303-5

Harrach, M. A. (z. j.). *Architectural graph of a multilayered Artificial Neural Network (ANN) model.* Angers, Frankrijk: University of Angers.

Hensman, P. & Masko, D. (2015). The Impact of Imbalanced Training Data for Convolutional Neural Networks. *KTH, Skolan för datavetenskap och kommunikation (CSC)*.

Hloewe. (z. j.). *Artificial Neuron.* Openclipart.

Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. *arXiv:1409.4842*. Verkregen van http://arxiv.org/abs/1608.06993 doi: 10.1109/CVPR.2017.243

Karn, U. (2016). *A Quick Introduction to Neural Networks.* Op 2018-01-30 verkregen van https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/

Karpathey, A. (2017). *Convolutional Neural Networks for Visual Recognition.* Op 2018-01-31 verkregen van http://cs231n.github.io/

Kostyaev, D. (2016). *Why top-5 error is more fair metric than top-1 for ILSVRC models.* Op 13-02-2018 verkregen van http://blog.kostyaev.me/blog/Why-top-5-error-is-more-fair-metric-than-top-1-for-ImageNet-classification-task

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007

Kruntchen, P. (1995). Architectural blueprintsâĂŞtheâĂİ 4+ 1âĂİ view model of software architecture. *IEEE Software*, *12*(November), 42–50. doi: 10.1145/216591.216611

Maron, O. & Ratan, A. L. (1998). Multiple-Instance Learning for Natural Scene Classification.

International Conference on Machine Learning, 341–349.

Quintor. (z. j.). *Quintor.* Op 2018-01-29 verkregen van `https://www.quintor.nl/wie-we-zijn/`

Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, *abs/1506.01497*. Verkregen van `http://arxiv.org/abs/1506.01497`

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018). Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *arXiv:1801.04381*. Verkregen van `http://arxiv.org/abs/1801.04381`

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2014). Going Deeper with Convolutions. *arXiv:1409.4842*. Verkregen van `https://arxiv.org/abs/1409.4842` doi: 10.1109/CVPR.2015.7298594

TensorFlow. (z. j.). *TensorFlow.* Op 2018-03-03 verkregen van `https://www.tensorflow.org/`

ToolMax. (2018). *ToolMax.* Op 2018-01-29 verkregen van `https://www.toolmax.nl/about`

Tweakers.net. (2018). *MSI GeForce GTX 1080 GAMING X 8G.* Op 15-05-2018 verkregen van `https://tweakers.net/pricewatch/545523/msi-geforce-gtx-1080-gaming-x-8g.html`

van Oosterhout, A., Verdonk, M. & Scheffer, D. (2017). *Twinkle 100: de top van de nederlandse e-commerce gebundeld*.

Wang, J. & Perez, L. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *Unpublished*.

A | Interview antwoorden - vooronderzoek

Geïnterviewde Johan Tillema – datum: 29-02-2018

1. Worden er op dit moment producten met de hand geclassificeerd?

Leverancies leveren Excelsheets, waarin productpropereties, adviesprijzen en verschillende afbeeldingen vermeld staan. Daarbij bevat een Excelsheet de categorie van het product, zoals aangegeven door de toeleveranciers. Deze categorie hoeft echter niet overeen te komen met de categorie die ToolMax aan het product geeft. Aan de hand van ''mapping'' wordt het product naar de categorie ''gemapped'' waar ToolMax het wil hebben.

''Mappping'' is belangrijk, omdat alle leveranciers andere categorieën hanteren voor hetzelfde product. Toeleverancies maken dus geen gebruik van een gestandaardiseerde categorisatie methode. Door ''mapping'' zullen de dezelfde producten van verschillende leverancieres toch in dezelfde categorie ''gemapped'' worden. Op deze manier is het eigenlijk meer een ''mapping proces'' dan een classificatieproces.

2. Zo ja, hoeveel personen zijn hier verantwoordelijk voor?

Binnen ToolMax zijn er twee man verantwoordelijk voor het instellen van de ''mapping'' c.q. classificering van de verschillende producten. Meerdere malen op een dag komen er verschillende Excelsheets binnen. ToolMax biedt 90.000 producten aan welke onderverdeeld zijn in ongeveer 200 categorieën. Daarbij zijn er ongeveer 129 leveranciers die deze producten leveren. Voor de medewerkers van ToolMax is het dus een dagtaak om al deze sheets te correct verwerken.

3. Hoe lang duurt het gemiddelde productclassificatie proces?

De Exelsheets bevatten de verschillende producten van een leveranciers. Iedere Exelsheet heeft dan ook een ander formaat, welke kan variëren tussen 50 en 1000+ regels. De tijd om een Excelsheet correct te verwerken kan dus erg verschillen. Indien een bepaalde ''mapping'' gemaakt is, wordt deze teruggekoppeld naar de toeleverancier. Dit zorgt ervoor dat de meeste ''mappings'' worden overgenomen door de leveranciers, wat in het vervolg tijd zal besparen. Nieuwe producten zorgen echter voor veel uitzoek/discussie werk, omdat deze voor het eerst geclassificeerd worden. Excelsheets waarop veel nieuwe producten staan, zijn dus erg tijdrovend. Het verwerken van specifieke subproducten kosten daarentegen de meeste tijd. Zo heb je bijvoorbeeld een schroevendraaier, maar deze kan je onderverdelen in een kruiskopschroevendraaier , torx schroevendraaier e.d. De platte schroevendraaiers kun je daarnaast weer onderverdelen in verdere categorieën. Dit allemaal bij elkaar zorgt er voor dat een medewerker gemiddeld één uur per Excelsheet bezig is.

4. Worden deze classificaties nog gevalideerd?

De verschillende "mappings" c.q. classificaties die gedaan worden, worden niet gevalideerd.

5. Hoe ziet het proces er globaal uit?

Een toeleverancier zendt een Excelsheet naar ToolMax. Een van de twee medewerkers pakt deze Excelsheet op om te verwerken, waarbij vervolgens de volgende stappen worden uitgevoerd:

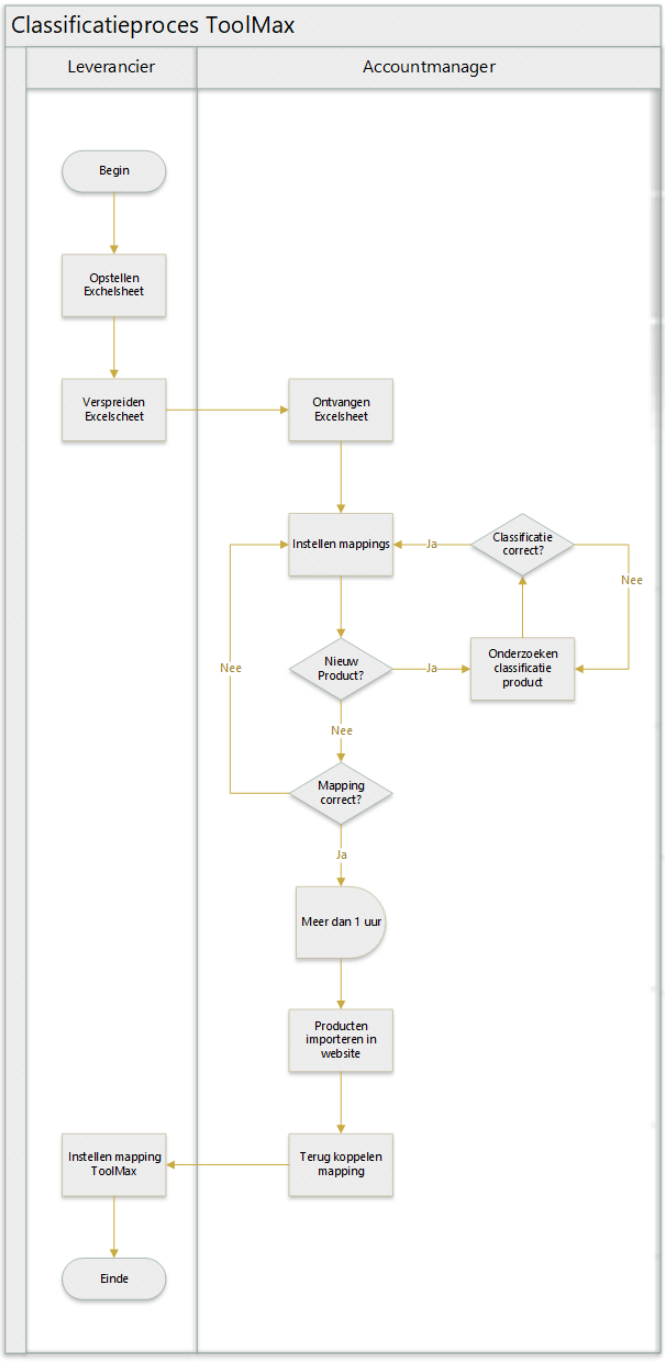
1. *Productgegevens worden gecontroleerd;*
2. *De "mapping" wordt ingesteld;*
    a. *Leverancier categorie naar ToolMax categorie.*
3. *Wanneer het een nieuwe product is:*
    a. *Er wordt gekeken binnen welke categorie dit product valt:*
        i. *Onderzocht*
        ii. *gediscussieerd*
    b. *De "mapping" van dit product wordt goed gezet;*
    c. *Er wordt gecontroleerd of dit de juiste "mapping" is;*
4. *Alles wordt in magneto geïmporteerd;*
5. *"Mapping" wordt teruggekoppeld naar de toeleverancier;*

6. Wat zijn de grote knelpunten in het huidige proces?

Het ''mappen'' van de verschillende producten is erg tijdrovend en kostbaar, aangezien er meerdere keren op een dag verschillende Excelsheets binnen komen. Aan één Excelsheet werkt een werknemer minstens één uur.

Binnen het proces is kwaliteit een knelpunt. De kwaliteit lijdt momenteel onder het feit dat het ''mappen'' c.q. classificeren veel tijd kost. Dit zorgt ervoor dat men sneller geneigd is om over de verschillende producten heen te lopen en ''mappings'' in te stellen die niet specifiek genoeg zijn. Op deze manier worden er veel producten naar dezelfde categorie ''gemapped'' of over het hoofd gezien. De kwaliteit van ToolMax daalt op deze manier, omdat categorieën verkeerde producten kunnen bevatten en een klant dus langer moet zoeken om het desbetreffende product te vinden. De ''bounce percentage'' zal hierdoor waarschijnlijk omhoog gaan en klanten kunnen worden misgelopen

B | Classificatieproces ToolMax



Figuur B.1: Classificatieproces ToolMax

C | Interviewvragen - requirements

Geïnterviewde Johan Tillema – datum: 12-03-2018

1.. Wat kan een geautomatiseerd proces oplossen?

2. Wat moet een geautomatiseerd proces oplossen?

3. Wanneer is een geautomatiseerd proces geschikt om gebruikt te worden?

4. Wat moet een geautomatiseerd proces minimaal kunnen?

5. Welke voorwaarden zitten er aan een geautomatiseerd proces? (denk aan nauwkeurigheid, handelingen, etc)

D | Pakketselectie

Het doel van deze bijlage is het kiezen van een geschikte library voor dit project.

Requirements en knock-out criteria

De meest geschikte library voor het project, moet voldoen aan verschillende voorwaarden c.q requirements. De requirements worden gerangschikt aan de hand van het MoSCow principe. Het MoSCoW principe beschrijft de prioriteit van de verschillende requirements, waarbij de M staat voor ''must have'', welke aanwezig moeten zijn in de library. De S staat voor ''should have'', welke zeer gewenst zijn in het framework, maar niet absoluut noodzakelijk zijn. De C staat voor ''could have'', deze eisen kunnen aanwezig zijn en geven de library functionaliteiten welke niet noodzakelijk zijn voor de werking van het systeem maar wel bepaalde voordelen bieden of relevant zijn bij de vergelijking met andere systemen. De W staat voor ''won't have'', deze zullen niet gebruik worden bij het kiezen van een specifieke library.

Must haves

- Snelheid;
- Inzetbaarheid;
- Contrability;
- Testability.

Het classificeren van afbeeldingen is een lastig proces, waarbij het realtime classificeren van producten voor het huidig project van belang is. Snelheid van de library is dan ook de belangrijkste requirement, omdat deze ervoor zorgt dat er op grotere datasets getraind kan worden. Het framework moet echter wel eenvoudig inzetbaar zijn op het gebied van architectuur en infrastructuur van bedrijven, daarbij in het bijzonder die van ToolMax.

Het classificeren van producten is, zoals eerder beschreven, een tijdrovend en complex proces, waarbij veel acties nodig zijn om het gewenste eindresultaat te behalen. Het is dan ook van groot belang dat de stappen gecontroleerd en getest kunnen worden, zodat eventuele fouten inzichtelijk worden gemaakt.

Should haves

- Schaalbaarheid;
- Trainbaarheid.

Zoals reeds eerder beschreven, is snelheid een belangrijke requirement. Indien de load van een library verdeeld kan worden over meerdere systemen, zal de snelheid van de library vergroot worden. Het trainen van de library is ook een zeer belangrijk onderdeel, waarbij het gemakkelijk kunnen trainen van de library en het kunnen opslaan van de resultaten van invloed zijn op de training.

Could haves

- GPU gebruik;
- Usability.

Een GPU in combinatie met het schalen van de library, zorgt ervoor dat de snelheid van de library aanzienlijk verbeterd. Libraries bieden veel verschillende mogelijkheden en functies, waardoor een goede "usability" (documentatie, referenties, toepasbaarheid, installatie, learncurce etc.) een waardevolle toevoeging is voor de uiteindelijke library.

Knock-out criteria

De definitie van een knock-out criteria is als volgt: "Eisen waaraan een item absoluut moet voldoen, zo niet dan valt het af".

Aan de hand van eerder beschreven requirements kunnen we de volgende knock-out criteria beschrijven:

- De library moet snel en efficiënt verschillende taken uitvoeren.
- De library moet gemakkelijk ingezet kunnen worden op zowel zakelijke infrastructuren als architecturen.
- De acties die de library uitvoert, moet gemakkelijk te controleren zijn.
- De acties die de library uitvoert, moet gemakkelijk getest kunnen worden.

Om een framework op te kunnen nemen in de longlost, moet het framework voldoen aan bovenstaande eisen. Opvallend is dat de eisen overeenkomen met de requirements die als ''must haves'' aangeduid zijn. In principe is dit logisch aangezien dit de belangrijkste eisen van een library zijn. Indien de library niet voldoet aan de requirements, is het niet geschikt voor het project.

Selectie

In deze paragraaf wordt de daadwerkelijke selectie gemaakt, aan de hand van een long- en shortlist. Na de selectie worden de twee overgebleven libraries getest.

Libraries

Vele libraries bieden mogelijkheden voor datascience, in het bijzonder voor deep learning en image classificatie. Voordat er een longlist opgesteld wordt, zal er eerst een lijst met mogelijke libraries samengesteld worden. Vervolgens zullen de verschillende knock-outs criteria, welke eerder zijn beschreven, worden toegepast om de longlist op te stellen.

Gevonden libraries

LIBRARY	TAAL	SNELHEID	INZETBAARHEID	CONTRABILITY	TESTABILITY	SCHAALBAARHEID	TRAINBAARHEID	GPU	USABILITY
THEANO	Python	✓	✓	✓	✓	✓	✓	✓	×
TORCH	Luna	✓	×	✓	✓	×	✓	✓	×
CAFFE	Python	✓	✓	×	✓	×	✓	✓	×
DEEPLEARNING4J	Java	✓	✓	✓	✓	✓	✓	✓	✓
TENSORFLOW	Python, C++, Java, Go)	✓	✓	✓	✓	✓	✓	✓	✓
DARCH	R	×	×	×	×	×	×	×	×
COVNET.JS	Javascript	×	×	×	×	×	×	×	×
MOCHA	Julia	✓	×	✓	✓	×	✓	✓	×
CNTK	.Net	✓	✓	✓	✓	✓	✓	✓	×
H2O	R, Python, Java, Scala	✓	✓	✓	✓	✓	✓	✓	×

Bovenstaande tabel is absoluut niet volledig. Op de lijst staan slechts de meest bekende libraries, op het gebied van data science en deep learning, vermeld. Het tabel bevat vinkjes en kruisjes. De vinkjes geven aan wanneer een library voldoet aan een requirement, en de kruisjes geven aan wanneer dit niet het geval is.

Longlist

De longlist wordt opgesteld aan de hand van de tabel welke vermeld is in paragraaf ‘’Gevonden Libraries’’. De longlist zal bestaan uit 6 libraries, welke voldoen aan de knock-out criteria.

Hieronder zullen de libraries vermeld worden:

theano

ONTWIKKELAAR
Université de Montréal

OMSCHRIJVING
Theano is een Python library waarmee wiskundige expressies, met betrekking tot multidimensionale matrixen, efficiënt kan definiëren, optimaliseren en evalueren.

ONTWIKKELAAR
Skymind
OMSCHRIJVING
Deeplearning4j is een distributed neural network library geschreven in Java en Scala. Deeplearning4j intergreert Hadoop en Spark en kan met zowel CPUs als GPUs overweg.

ONTWIKKELAAR
Google Brain Team
OMSCHRIJVING
Tensorflow is een library voor numerieke berekeningen met behulp van gegevensstroomgrafieken. TensorFlow werkt op een of meerdere CPU's, GPU's, desktops of server. Tensorflow is ontwikkeld door een team bij Google dat zich bezig houdt met machine learning

ONTWIKKELAAR
Microsoft
OMSCHRIJVING
De Microsoft Cognitive Toolkit, ookwel bekend als CNTK, stelt de programeur in staat om de intelligentie van enorme datasets te gebruiken door middel van deep learning door ongecompliceerde schaalbaarheid, snelheid en nauwkeurigheid te bieden met commerciële kwaliteit en compatibiliteit met programmeertalen en algoritmen die u al gebruikt

ONTWIKKELAAR
H2O.ai
OMSCHRIJVING
H2O's Deep Learning is gebaseerd op een multi-layer feedforward artificieel neuraal netwerk dat is getraind met stochastische gradiëntdaling met behulp van back-propagatie. Het netwerk kan een groot aantal verborgen lagen bevatten bestaande uit neuronen met anh, rectifier, en maxout activeringsfuncties.

Shortlist

Een shortlist is een lijst van frameworks, welke na de eerste selectie van de longlist zijn overgebleven. De shortlist is een onderdeel van de longlist en bevat frameworks die voldoen aan zowel de knock-out criteria als requirements.

Totstandkoming

Om de shortlist te creëren is er gebruikt gemaakt van verschillende documentaties over libraries, welke online te vinden zijn. Ook zijn er van de gekozen liberaries reviews te vinden, welke deze volledig uitpluizen. Tot slot is het mogelijk om de source code van een library te downloaden, zodat men er zelf mee kan experimenteren.

Door gebruik te maken van informatieverzameling zoals hierboven beschreven, zal de shortlist betrouwbaar en degelijk tot stand komen. De shortlist is hierbij zo goed mogelijk afgestemd op de objectiviteit en puurheid van de requirements.

Op deze manier blijft er voor de shortlist enkel de libraries Deeplearning4j en Tensorflow over. De resterende libraries kwamen vooral te kort op: installatiegemak, de grootte van de leercurve, het leren van een nieuwe taal en de volwassenheid van de taal. Tot slot vielen enkele uit vanwege een tekort op schaalbaarheid en de mogelijkheid tot het controleren van de acties.

De shortlist

Deeplearning4j

Ontwikkelaar

Skymind

Omschrijving

Deeplearning4j is een distributed neural network library, geschreven in Java en Scala. Deeplearning4j intergreert met Hadoop en Spark en kan met zowel CPUs en GPUs overweg.

Technieken

Restricted Boltzman machine (RBM), deep belief network (DBN), deep autoencoder, stacked denoising autoencoder, recursive neural network (RNN), artificial neural networks (ANN), convolutional neural network (CNN), word2vec, doc2vec en GloVe (Global Vectors)

Bestemd voor

Ontwikkelaars die ontwikkelen binnen Java Virtual Machine (JVM) en die behoefte hebben aan een library voor het bouwen, trainen en het inzetten van neurale netwerken. Deeplearning4j is een compleet pakket van meerdere tools, welke gebruikt kunnen worden voor datascience.

Gemiddelde rating

De gemiddelde rating van deeplearining4j is een 7.8 (Today, sd)

Prijs

Deeplearing4j is een gratis library

Waarom in de shortlist?

Deeplearning4j is gebaseerd op een techniek die veelal gebruikt wordt in grote bedrijven, namelijk de JVM en daarbij de programmeertaal Java. Dit zorgt ervoor dat Deeplearning4j gemakkelijk in te zetten is in deze bedrijfsstructuren. Daarnaast is het ook mogelijk om het systeem gedistribueerd te laten trainen, waardoor de training sneller gaat en er op meer en grotere datasets getraind kan worden. Op deze manier kunnen er nauwkeurigere resultaten behaald worden. Tot slot is de usability van deeplearning4j goed en is het mogelijk om de neurale netwerken en acties te visualiseren, waardoor deze goed controleerbaar zijn.

TensorFlow

Ontwikkelaar
Google Brain team
Omschrijving
Tensorflow is een library voor numerieke berekeningen, met behulp van gegevensstroomgrafieken. TensorFlow werkt op een of meerdere CPU's, GPU's, desktops of servers. Tensorflow is ontwikkeld door een team bij Google die zich bezig houdt met machine learning

Technieken
Logistic regression, fully connected networks, recursive neural network (RNN), artificial neural networks (ANN), convolutional neural network (CNN), random forests, reinforcement learning, generative models
Bestemd voor
Ontwikkelaars en onderzoekers, die een framework willen gebruiken , welke makkelijk is in gebruik en waarbij alles zelf bepaald kan worden.
Gemiddelde rating
Tensorflow heeft een gemiddelde rating van 8.2 (Today, sd)
Prijs
Tensorflow is een gratis library en kan dan ook gratis gebruikt worden voor commerciële doeleinden.
Waarom in de shortlist?
Tensorflow is een bekende en veel gebruikte library voor datascience, in het specifiek machine learning, waarbij er een groot community is ontstaan voor tensorflow en er dus veel documentatie te vinden is voor TensorFlow. Dit zorgt ervoor dat de leercurve van TensorFlow laag is en er makkelijk op ingestapt kan worden, wat de usability ten goede komt. TensorFlow is ook gedistribueerd opgezet, waardoor dit een extra snelheid winst kan opleveren voor machine learning taken.

Testopstelling

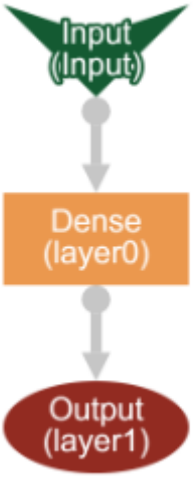
Om de twee libraries goed met elkaar te kunnen vergelijken, zal er een testopstelling opgesteld worden. Beide libraries zullen aan de hand van input proberen te bepalen welke soort iris de data beschrijft. Dit wordt gedaan aan de hand van de ''iris dataset''

De iris dataset is een van de bekendste patroonherkenning datasets. De dataset bevat drie iris soorten met elke 50 waardes. De bedoeling van deze dataset is om de soort iris plant te voorspellen aan de hand van: de lengte van het kelkblad, de breedte van het kelkblad, de lengte van het bloemblad en de breedte van het bloemblad (Lichman, 2013).

Deze testopstelling bestaat uit een neuraal netwerk, welke getraind wordt en zo leert om verschillende iris soorten te herkennen. Er is gekozen voor deze opstelling, omdat daarmee de eigenschappen en de mogelijkheden van de twee libraries goed vergeleken kunnen worden.

Tijdens de opstelling wordt er gekeken naar de verschillende requirements, maar ook naar flexibiliteit, nauwkeurigheid en aanpasbaarheid.

Deeplearning4j



Om de testopstelling te realiseren met Deeplearning4j, worden eerst de twee CSV's ingelezen en in het geheugen gezet. Er zijn twee types, een met alle train data en een met de test data. Vervolgens moet de ANN in zijn geheel opgezet en gedefinieerd woorden, zodat de ANN getraind en geëvalueerd kan worden.

Het uitvoeren van de iris test heeft een nauwkeurigheid van 93% en een precisie van 93%, waarbij het systeem van de 30 test records 28 keer true positive was en 2 keer true negatative.

Wat opvalt is dat Deeplearning4J relatief veel geheugen gebruikt voor een zeer simpel classificatieprobleem.

TensorFlow

Ook bij de testopstelling van TensorFlow worden eerst de twee CSV's ingelezen en daarna in het geheugen gezet. Vervolgens wordt er dezelfde ANN opgezet als bij Deeplearning4j.

De nauwkeurigheid van TensorFlow is 96% en is daarmee nauwkeuriger dan Deeplearning4j. De overige informatie wordt niet teruggekoppeld door TensorFlow en kan dus niet gedefinieerd worden

Conclusie/ aanbeveling

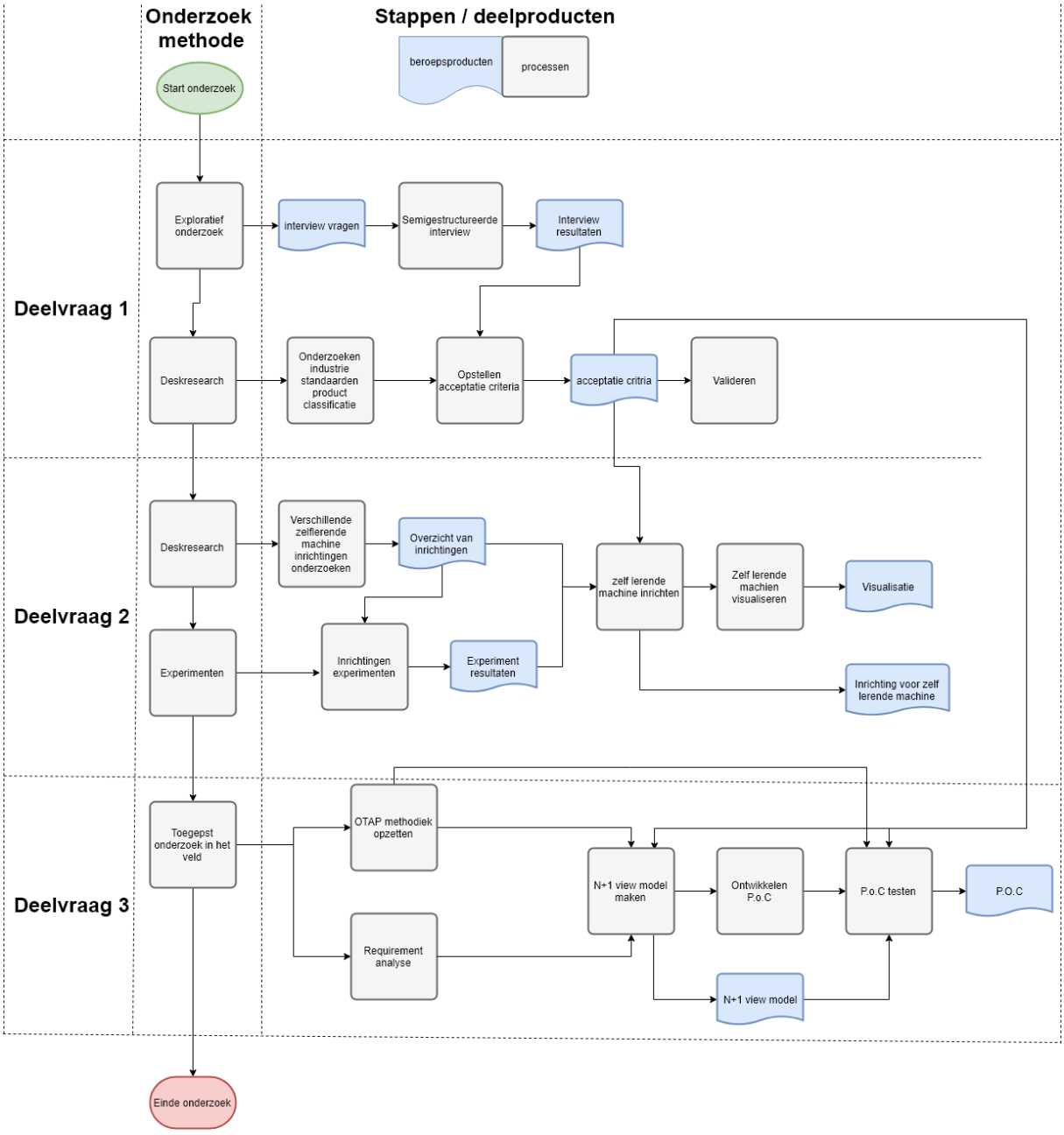
TensorFlow is ten opzichte van Deeplearning4j nauwkeuriger en simpeler in gebruik (met een kleinere leercurve). Echter kan de nauwkeurigheid van Deeplearning4j nog verbeterd worden.

Naast de kleine leercurve van TensorFlow, zorgt het grotere geheugengebruik van Deeplearning4J zorgt ervoor dat de voorkeur uitgaat naar TensorFlow. Dit omdat de testopstelling met een relatief kleine dataset werkt t.o.v. het uiteindelijke dataset. Het grotere geheugengebruik kan ervoor zorgen dat Deeplearning4J niet in staat is om te classificeren met een grote dataset.

TensorFlow heeft een betere rating, lagere leercurve en een kleiner geheugenverbruik. Het feit dat TensorFlow niet gemakkelijk integreerbaar is binnen de huidige architectuur van ToolMax , aangezien deze binnen de JVM werkt, is dit op te lossen door gebruik te maken van microservices.

Door bovenstaande argumentaties wordt TensorFlow aanbevolen als framework voor PoC

E | Totaalbeeld



Figuur E.1: Totaalbeeld van de verschillende stappen in het onderzoek

F | Interview antwoorden - requirements

Geïnterviewde Johan Tillema – datum: 12-03-2018

1.. Wat kan een geautomatiseerd proces oplossen?

Het voordeel van een geautomatiseerd proces is dat deze ‘s nachts kan draaien, voor elke productclassificatie de tijd neemt en een deel van het classificeren kan overnemen. Als een geautomatiseerd systeem alleen al bijvoorbeeld 40% automatisch kan classificeren zorgt dit ervoor dat er heel wat tijd en ook kosten bespaard kan worden.

Daarnaast zal een automatische proces het kwaliteit probleem oplossen, doordat deze evenveel tijd zal nemen voor elk product en per product zal bekijken tot welke categorie hij behoort. Dit zorgt ervoor dat de verschillende producten beter geclassificeerd kunnen worden.

Tot slot kan het systeem een controlerende functie vervullen, aangezien het proces over ToolMax heen kan lopen en elke product afzonderlijk kan classificeren, waarbij er gekeken kan worden of de verschillende klasses overheen komen. Indien deze twee klasses niet overeen komen, kan er gekeken worden of er toevallig verkeerd is gemapped.

2. Wat moet een geautomatiseerd proces oplossen?

Het systeem moet minstens het proces van mappen versnellen, Dit wil zeggen dat het met minder werknemers dezelfde workload moet kunnen verwerken. Het versnellen van het proces mag echter niet ten kosten gaan van de kwaliteit van ToolMax. Dus de verschillende producten moeten nog steeds op een betrouwbare manier geclassificeerd worden.

‘‘Sneller’’ betekend in deze context echter niet dat de machine het sneller moet kunnen classificeren dan een mens, maar dat dezelfde workload met minde rmensen verwerkt kan worden. Als het systeem bijvoorbeeld twee keer zoveel tijd nodig heeft ten opzichte van een persoon, is dit geen probleem aangezien het systeem ook 's nachts kan werken.

3. Wanneer is een geautomatiseerd proces geschikt om gebruikt te worden?

De uitkomst van de zelflerende machine moet voorspelbaar zijn, dit wil zeggen dat de machine zelf aan moet geven hoe betrouwbaar deze classificatie is. Ofwel de machine moet een percentage aangeven bij de classificatie, , welke aangeeft hoe zeker de machine is over zijn classificatie.

Daarnaast moet de machine zelflerend zijn, , wat wil zeggen dat het elke keer weer moet bijleren. Dit moet doormiddel van twee methodes:

1. *Geautomatiseerd leren op basis van de categorie.*
2. *Leren op basis van wat de medewerker aangeeft.*

De machine krijgt een basis dataset waarmee hij de basis leert waarna de machine gaat classificeren, wanneer hij het fout heeft zal deze door de medewerker geclassificeerd worden. Deze handmatige classificaties zullen gebruikt moeten worden om de machine verder te trainen zodat de machine gaande weg in de tijd steeds beter wordt in het classificeren van verschillenden producten

4. Wat moet een geautomatiseerd proces minimaal kunnen?

Het proces moet minstens de mapping kunnen creëren. Dit betekent dat de machine minstens de verschillende producten moet kunnen classificeren. Het proces moet daarbij continu kunnen leren, zodat de classificatie steeds betrouwbaarder wordt en de productclassificatie kan worden uitgebreid.

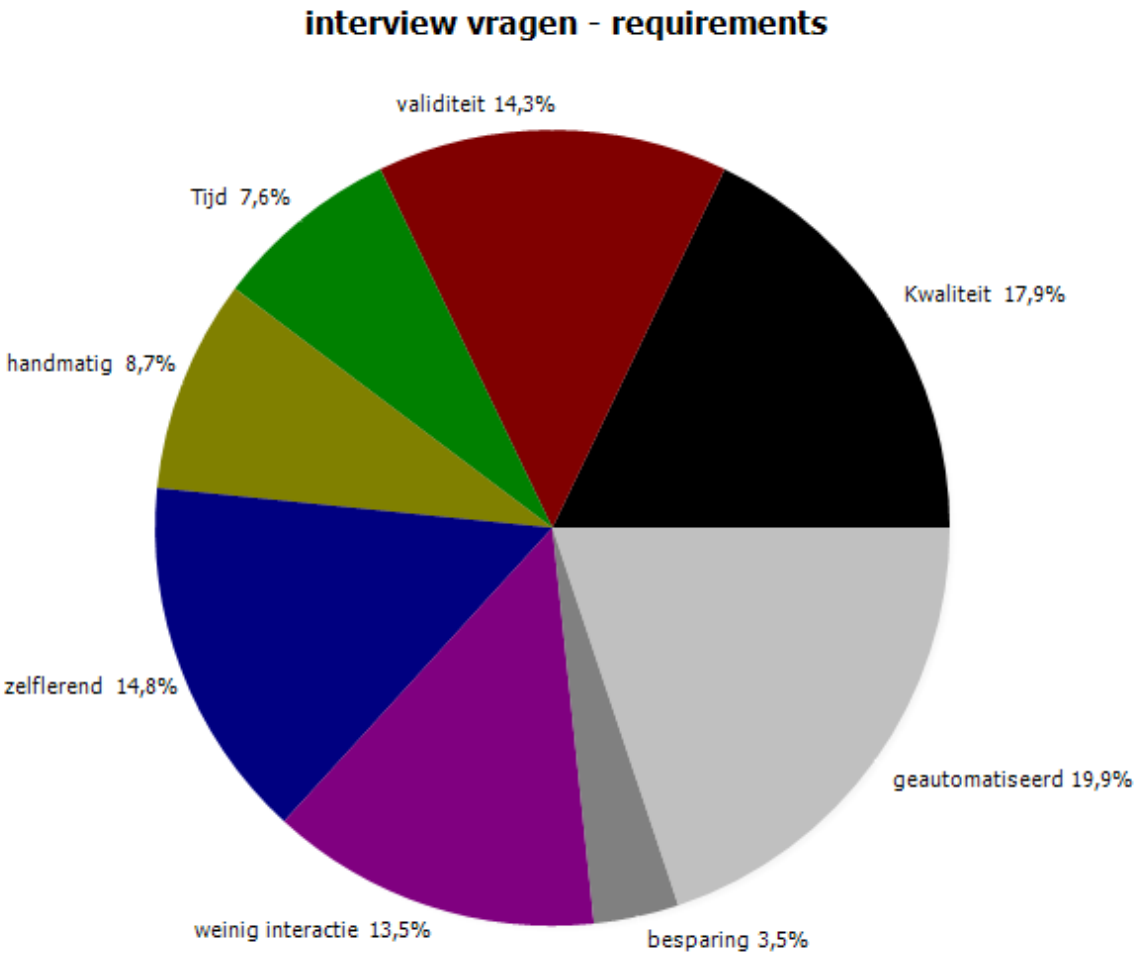
Daarnaast zal het een mooie toevoeging zijn als de tool gebruikt kan worden om de eigen mapping c.q. classificatie te controleren. Op deze manier kan er gecheckt worden of de bestaande productmappings correct zijn.

Tot slot zorgt een geautomatiseerd proces ervoor dat ToolMax zijn producten extra kan valideren, zonder dat hiervoor extra personeel noodzakelijk is.

5. Welke voorwaarden zitten er aan een geautomatiseerd proces? (denk aan nauwkeurigheid, handelingen, etc)

1. *Alle classificaties met een nauwkeurigheid van minder dan 70% moeten handmatig gevalideerd worden.*
2. *Aan de hand van input van gebruikers, zal de machine leren.*
3. *De machine moet minstens twee methodes van leren bevatten:*
    a. *Geautomatiseerd leren op basis van de categorie.*
    b. *Leren op basis van wat de medewerker aangeeft.*
4. *De machine moet zelflerend zijn wanneer het een aantal keer een afbeelding fout geclassificeerd heeft en de medewerker de classificatie met de hand gedaan heeft. Na deze handelingen moet de machine het product met minstens 70% nauwkeurigheid kunnen classificeren.*
5. *Medewerkers moeten zo min mogelijk betrokken worden in het proces van het classificererend systeem.*
6. *De compute power is onbelangrijke voor ToolMax. Dit wil zeggen dat er geen harde eisen aan RAM verbruik of CPU gebruik zit.*
7. *De tijd is onbelangrijk voor ToolMax aangezien het proces ook 's nachts kan draaien.*

G | Code frequentie



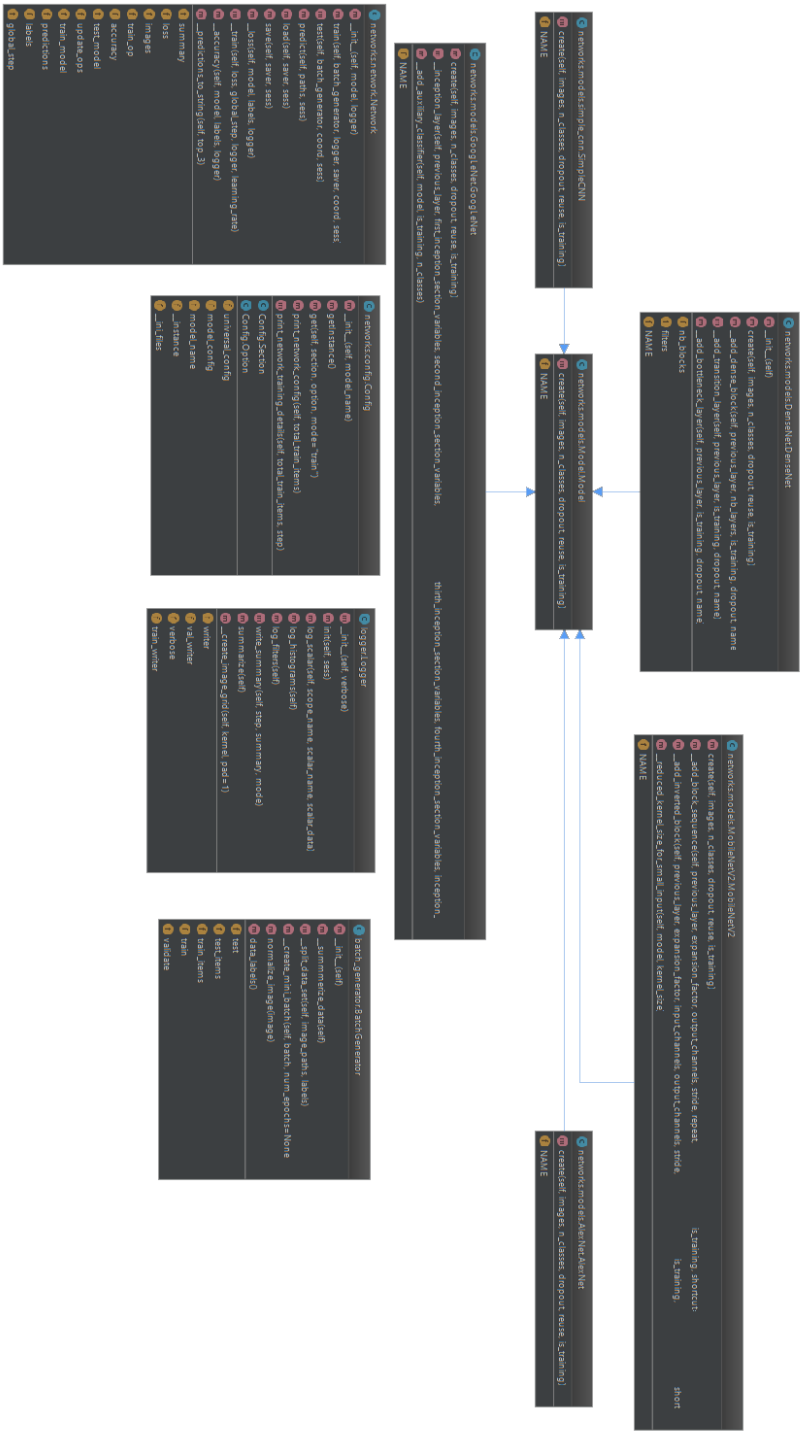
Figuur G.1: Code frequenties interview

H | Requirements

Nummer	*Omschrijving*	*Prioritering*
F01	Het systeem moet automatisch kunnen classificeren	M
F02	Het systeem moet op basis van twee manieren leren. D.m.v. automatisering op bais van de categorie van de afbeelding en d.m.v. wat de medewerker aangeeft	M
F03	Het systeem moet de interactie met medewerkers tot een minimaal beperken	M
F04	De resultaten van het systeem moeten herproduceerbaar zijn	M
F05	Het systeem moet een zekerheidspercentage geven bij een classificatie	M
F06	Het systeem moet een top 1 error hebben van minder dan 50%	M
F07	Het systeem moet een top 3 error hebben van minder dan 25%	M
F08	Het systeem moet een top 5 error hebben van minder dan 12,5%	M
F09	Het systeem moet bij een zekerheidspercentage van minder dan 70% overgaan op handmatige classificatie	S
F10	Het systeem moet, nadat handmatige classificatie enkele keren is voorgekomen, classificaties automatisch uitvoeren	S
F11	Het systeem moet gecorrigeerd kunnen worden door medewerkers, waarna het systeem leert van deze correctie	S
F12	Het systeem moet minstens een test nauwkeurigheid hebben van 90%	M
F13	Wanneer het systeem producten tegenkomt met een EAN die reeds geclassificeerd is, zal het deze automatisch naar de betreffende categorie classificeren	W

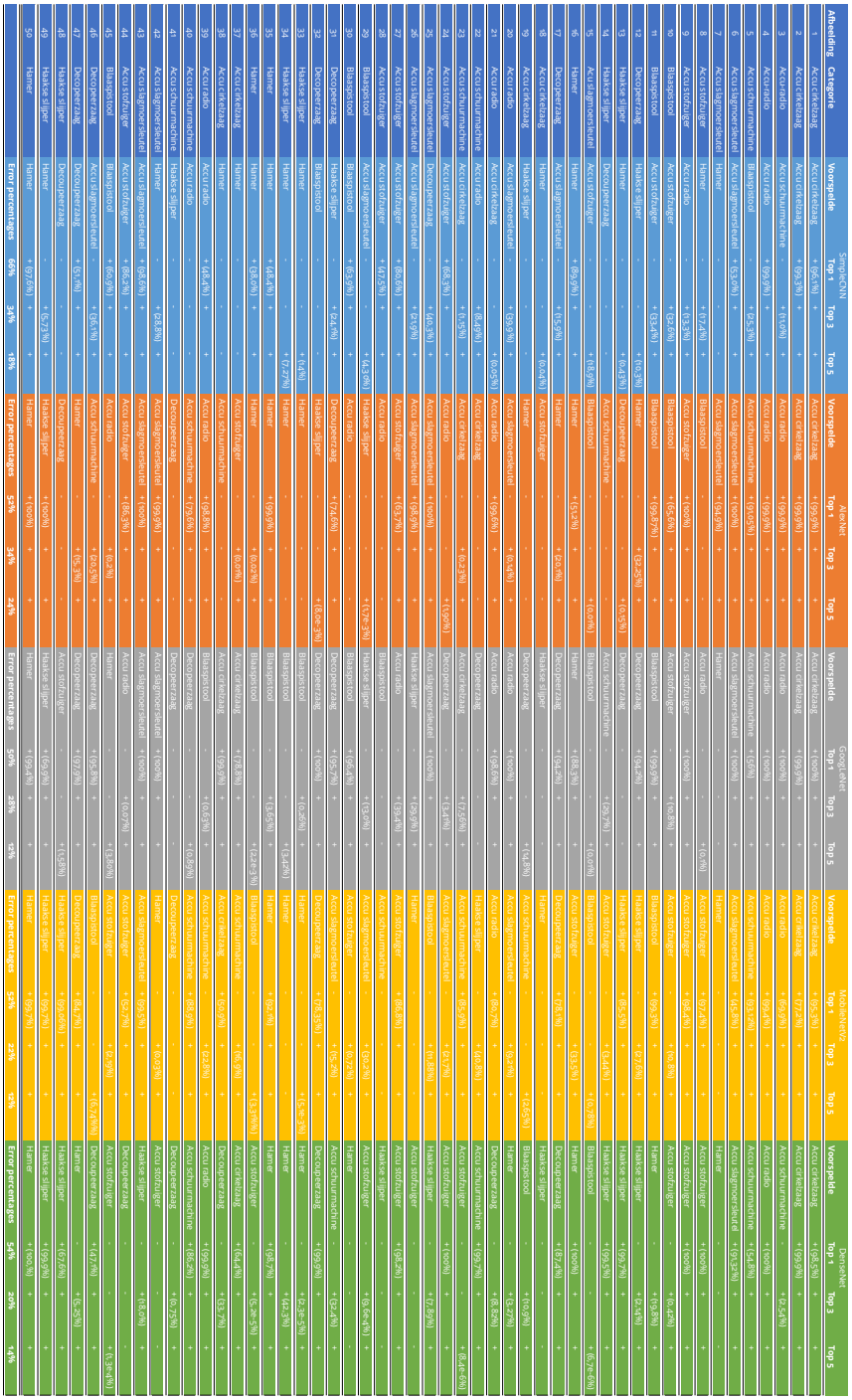
Tabel H.1: Requirements zelflerende systeem ToolMax

Class diagram test framework



Figuur I.1: Class diagram test framework

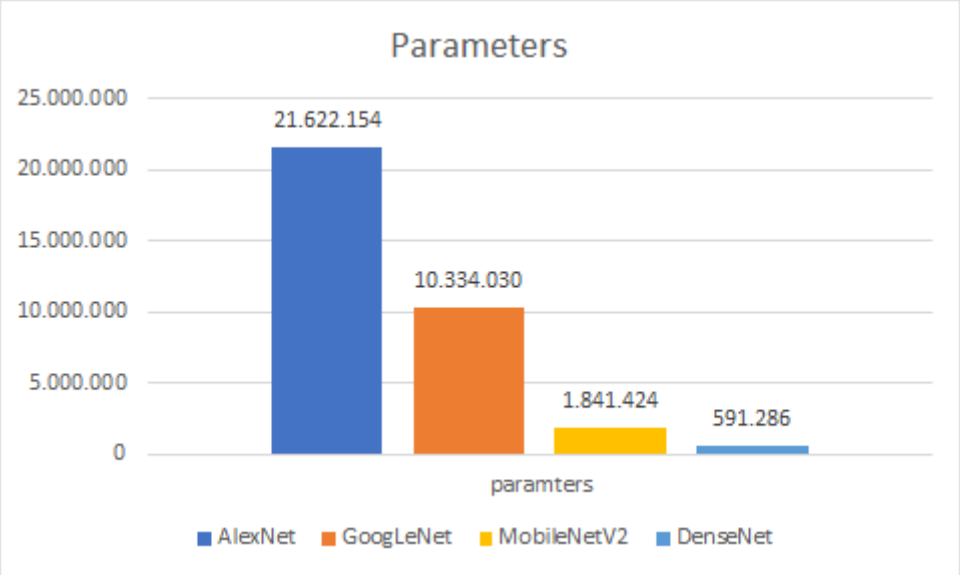
J | Top error waardes



Afbeelding	Categorie	SimpliCNN			AlexNet			GoogLeNet			MobileNetV2			DenseNet		
		Voorspelde	Top 1	Top 3	Top 5	Voorspelde	Top 1	Top 3	Top 5	Voorspelde	Top 1	Top 3	Top 5	Voorspelde	Top 1	Top 3	Top 5
1	Accukeukenzaag	Accukeukenzaag	+ (64,1%)	+	+	Accukeukenzaag	+ (99,9%)	+	+	Accukeukenzaag	+ (100%)	+	+	Accukeukenzaag	+ (98,5%)	+	+
2	Accukeukenzaag	Accukeukenzaag	+ (99,3%)	+	+	Accukeukenzaag	+ (99,9%)	+	+	Accukeukenzaag	+ (99,9%)	+	+	Accukeukenzaag	+ (99,9%)	+	+
3	Accuradio	Accuschuurmachine	+ (69%)	+	+	Accuradio	+ (99,9%)	+	+	Accuradio	+ (100%)	+	+	Accuschuurmachine	+ (100%)	+	+
4	Accuradio	Accuradio	+ (69,9%)	+	+	Accuradio	+ (99,9%)	+	+	Accuschuurmachine	+ (100%)	+	+	Accuradio	+ (100%)	+	+
5	Accuschuurmachine	Hamer	+ (64,3%)	+	+	Accuschuurmachine	+ (100%)	+	+	Accuradio	+ (100%)	+	+	Accuradio	+ (100%)	+	+
6	Accuslagmoerssleutel	Accuslagmoerssleutel	+ (67,3%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+
7	Accuslagmoerssleutel	Hamer	-	-	-	Hamer	-	-	-	Hamer	-	-	-	Hamer	-	-	-
8	Accustofzuiger	Hamer	-	-	-	Blaspistool	-	-	-	Accuradio	-	-	-	Hamer	-	-	-
9	Accustofzuiger	Accuradio	-	+ (63,3%)	+	Accustofzuiger	+ (100%)	+	+	Accustofzuiger	+ (100%)	+	+	Accustofzuiger	+ (100%)	+	+
10	Blaspistool	Accustofzuiger	-	+ (52,6%)	+	Blaspistool	+ (65,6%)	+	+	Accustofzuiger	+ (100%)	+	+	Accustofzuiger	-	+ (0,42%)	+
11	Blaspistool	Accustofzuiger	-	+ (61,4%)	+	Blaspistool	+ (69,9%)	+	+	Accustofzuiger	+ (99,9%)	+	+	Hamer	-	+ (0,93%)	+
12	Decoupeerzaag	Haakse slijper	-	+ (63,3%)	+	Hamer	+ (82,3%)	+	+	Decoupeerzaag	+ (82,4%)	+	+	Haakse slijper	+ (82,7%)	+ (2,3%)	+
13	Haakse slijper	Hamer	-	+ (62,4%)	+	Accuschuurmachine	+ (56,5%)	+	+	Decoupeerzaag	+ (82,4%)	+	+	Haakse slijper	+ (82,7%)	+ (2,3%)	+
14	Haakse slijper	Decoupeerzaag	-	-	-	Accuschuurmachine	-	-	-	Accuschuurmachine	+ (67,3%)	+	+	Haakse slijper	+ (99,5%)	-	+
15	Accuslagmoerssleutel	Accustofzuiger	-	+ (64,9%)	Blaspistool	-	+ (60,6%)	Blaspistool	-	+ (64,9%)	Blaspistool	-	+ (0,26%)	Blaspistool	-	+ (6,7e-4%)	
16	Hamer	Hamer	+ (89,6%)	+	+	Hamer	+ (55,2%)	+	+	Hamer	+ (88,3%)	+	+	Accustofzuiger	-	+ (0,15%)	+
17	Decoupeerzaag	Accuslagmoerssleutel	-	+ (65,8%)	+	Hamer	-	+ (69,8%)	+	Decoupeerzaag	+ (94,3%)	+	+	Decoupeerzaag	+ (87,4%)	+	+
18	Accuschuurmachine	Hamer	-	+ (67,4%)	Accustofzuiger	-	-	-	Haakse slijper	-	-	-	Hamer	-	-	-	
19	Accuschuurmachine	Haakse slijper	-	+ (53,3%)	+	Hamer	-	+ (74,3%)	+	Accuschuurmachine	+ (100%)	+	+	Accuschuurmachine	-	+ (0,93%)	+
20	Accuradio	Accuslagmoerssleutel	-	+ (53,3%)	+	Accuschuurmachine	+ (100%)	+	+	Accuschuurmachine	+ (100%)	+	+	Blaspistool	-	+ (0,42%)	+
21	Accuradio	Accuradio	-	+ (8,49%)	+	Accuradio	+ (69,6%)	+	+	Accuradio	+ (73,8%)	+	+	Hamer	-	+ (3,97%)	+
22	Accuschuurmachine	Accuradio	-	+ (8,49%)	+	Accuradio	+ (69,6%)	-	-	Decoupeerzaag	-	-	-	Accuradio	-	+ (3,82%)	+
23	Accuschuurmachine	Accuschuurmachine	-	+ (6,5%)	+	Accuschuurmachine	+ (0,23%)	+	+	Accuschuurmachine	+ (100%)	+	+	Accuschuurmachine	+ (100%)	+	+
24	Accustofzuiger	Accustofzuiger	+ (68,5%)	+	+	Accuradio	+ (0,23%)	+ (100%)	+	Decoupeerzaag	-	+ (2,46%)	+	Haakse slijper	-	+ (0,44e-6%)	
25	Accuslagmoerssleutel	Decoupeerzaag	+ (68,5%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Blaspistool	-	+ (0,42%)	+
26	Accuslagmoerssleutel	Accuslagmoerssleutel	+ (64,9%)	+	+	Haakse slijper	-	+ (59,7%)	+	Haakse slijper	-	+ (59,4%)	+	Accustofzuiger	-	+ (98,2%)	+
27	Accustofzuiger	Accustofzuiger	+ (86,6%)	+	+	Accuradio	-	+ (59,4%)	+	Accuradio	-	+ (59,4%)	+	Accustofzuiger	+ (98,2%)	+	+
28	Accustofzuiger	Accustofzuiger	+ (44,2%)	+	+	Accuradio	-	+ (59,4%)	+	Blaspistool	-	+ (59,4%)	+	Haakse slijper	-	+ (0,44%)	+
29	Blaspistool	Haakse slijper	-	+ (63,3%)	+	Accuradio	-	+ (74,3%)	+	Haakse slijper	-	+ (59,4%)	+	Accuschuurmachine	-	+ (0,6e-6%)	
30	Blaspistool	Blaspistool	-	+ (63,3%)	+	Accuradio	-	+ (74,3%)	+	Blaspistool	-	+ (59,4%)	+	Accuschuurmachine	-	+ (0,6e-6%)	
31	Decoupeerzaag	Haakse slijper	-	+ (74,9%)	+	Decoupeerzaag	+ (74,3%)	+	+	Decoupeerzaag	+ (74,3%)	+	+	Decoupeerzaag	+ (74,3%)	+	+
32	Decoupeerzaag	Blaspistool	-	+ (74,9%)	+	Decoupeerzaag	+ (74,3%)	+	+	Decoupeerzaag	+ (74,3%)	+	+	Decoupeerzaag	+ (74,3%)	+	+
33	Haakse slijper	Hamer	-	+ (74,9%)	+	Hamer	-	+ (74,3%)	+	Hamer	-	+ (74,3%)	+	Decoupeerzaag	+ (74,3%)	+	+
34	Haakse slijper	Hamer	-	+ (74,9%)	+	Hamer	-	+ (74,3%)	+	Hamer	-	+ (74,3%)	+	Hamer	-	+ (42,3e-3%)	+
35	Hamer	Hamer	+ (48,4%)	+	+	Hamer	+ (99,9%)	+	+	Hamer	+ (3,66%)	+	+	Hamer	+ (98,7%)	+	+
36	Hamer	Hamer	+ (98,6%)	+	+	Hamer	+ (99,9%)	+	+	Blaspistool	-	+ (3,66%)	+	Accustofzuiger	-	+ (5,2e-5%)	+
37	Accuschuurmachine	Hamer	-	-	-	Accuschuurmachine	+ (99,8%)	+	+	Accuschuurmachine	+ (72,3%)	+	+	Accuschuurmachine	+ (99,8%)	+	+
38	Accuradio	Accuradio	+ (49,4%)	+	+	Accuschuurmachine	+ (99,8%)	+	+	Accuschuurmachine	+ (72,3%)	+	+	Accuradio	+ (99,8%)	+	+
39	Accuradio	Accuschuurmachine	-	-	-	Accuschuurmachine	+ (99,8%)	+	+	Accuschuurmachine	+ (72,3%)	+	+	Accuschuurmachine	+ (99,8%)	+	+
40	Accuschuurmachine	Haakse slijper	-	-	-	Accuschuurmachine	+ (99,8%)	+	+	Accuschuurmachine	+ (72,3%)	+	+	Accuschuurmachine	+ (99,8%)	+	+
41	Accuschuurmachine	Haakse slijper	-	-	-	Decoupeerzaag	-	-	-	Accuschuurmachine	+ (72,3%)	+	+	Accuschuurmachine	+ (99,8%)	+	+
42	Accuslagmoerssleutel	Hamer	-	+ (68,8%)	+	Accuslagmoerssleutel	+ (100%)	+	+	Hamer	-	+ (98,7%)	+	Decoupeerzaag	-	-	-
43	Accuslagmoerssleutel	Accuslagmoerssleutel	+ (98,6%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Accuslagmoerssleutel	+ (100%)	+	+	Accuslagmoerssleutel	+ (99,9%)	+	+
44	Accustofzuiger	Accustofzuiger	+ (89,9%)	+	+	Accuradio	+ (99,9%)	+	+	Accustofzuiger	+ (100%)	+	+	Accustofzuiger	+ (100%)	+	+
45	Blaspistool	Accustofzuiger	+ (84,3%)	+	+	Accustofzuiger	+ (99,8%)	+	+	Hamer	-	+ (98,7%)	+	Decoupeerzaag	-	+ (16,9%)	
46	Decoupeerzaag	Accuschuurmachine	+ (55,1%)	+	+	Accuschuurmachine	+ (99,8%)	+	+	Accuschuurmachine	+ (72,3%)	+	+	Accuschuurmachine	+ (72,3%)	+	+
47	Decoupeerzaag	Decoupeerzaag	-	-	-	Hamer	-	-	-	Decoupeerzaag	+ (72,3%)	+	+	Hamer	-	+ (5,2e-5%)	+
48	Haakse slijper	Decoupeerzaag	-	-	-	Decoupeerzaag	-	-	-	Hamer	-	-	-	Accuschuurmachine	-	+ (5,2e-5%)	+
49	Haakse slijper	Hamer	-	+ (6,72%)	+	Haakse slijper	+ (100%)	+	+	Haakse slijper	+ (65,9%)	+	+	Haakse slijper	+ (99,9%)	+	+
50	Hamer	Hamer	+ (97,6%)	+	+	Hamer	+ (100%)	+	+	Hamer	+ (99,4%)	+	+	Hamer	+ (100%)	+	+
Error percentage		6%	34%	41%		34%	24%			38%	4%			23%	4%		
Error percentage		6%	34%	41%		34%	24%			38%	4%			23%	4%		
Error percentage		94%				66%				62%				77%			

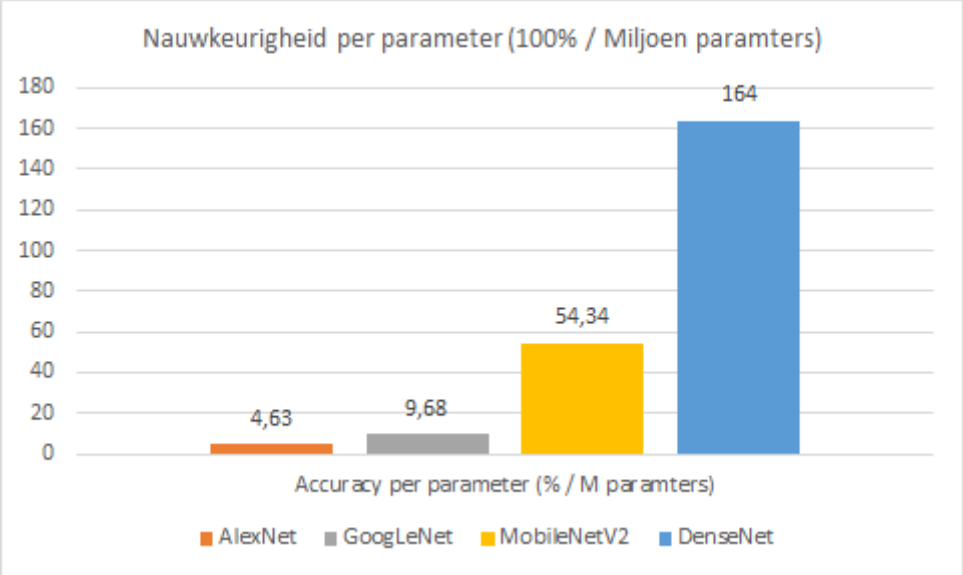
Tabel J.1: Top error waardes

K | Parameters per model



Figuur K.1: De parameters per model

L | Nauwkeurigheid per parameter



Figuur L.1: Nauwkeurigheid per parameter

M | N+1 view model

1 Inleiding

1.1 Doel/scope document

Het N+1 view-document wordt gebruikt om architectuur te beschrijven en de gemaakte keuzes en ontwerpen te documenteren. Deze keuzes/ontwerpen worden op zo een manier gedocumenteerd dat dit document geschikt is voor verschillende doelgroepen. Het is geschikt voor elke functie die invloed/input heeft op/tijdens de ontwikkeling van het project.

1.2 Doel/Scope project

Het doel van het onderzoek is om producten te classificeren door middel van de bijbehorende afbeeldingen (meta-data). Dit moet gebeuren op een manier waarbij het systeem kan leren en constant bij kan leren. De bedoeling is dan ook dat het systeem gaande weg steeds betere resultaten oplevert. Het zelf lerende zal toegepast worden doormiddel van Artificial Neural Networks (ANN).

In het algemeen zal het mogelijk moeten worden om verschillende afbeeldingen te uploaden naar het systeem en dat deze dan een response terug geeft met een label aan elke afbeelding. Zodat de ontvanger het response kan gebruiken om de producten toe te voegen aan de daar toe behorende categorie.

Metadata uit producten bestaat uit: Een titel, een beschrijving, producteigenschappen en verschillende afbeeldingen, van uit deze verschillende metadata kan veel informatie gehaald worden. Het project zal echter zich focussen op productclassificatie aan de hand van de productafbeeldingen, de rest van de metadata zal niet gebruikt worden om de producten te classificeren.

1.3 Publiek

Dit document is gericht aan de software engineers die werken aan dit project. Dit kan tijdens het aanvangsproject zijn maar ook tijdens vervolgprojecten. Daarnaast is dit document gericht aan de opdrachtgever waar deze de verschillende keuzes en afwegingen kan terug vinden en hierover kan oordelen

1.4 Status

De architectuur is op het moment van schrijven nog in ontwikkeling, het document zal nadat het geïmplementeerd is geupdate worden met de op- en aanmerkingen die naar boven zijn gekomen tijdens het ontwikkelen. Daarnaast zal input van de verschillende stakeholders nog toegevoegd worden aan dit document.

1.5 Architectuurontwerpbenadering

Het algemene idee van het systeem is een web interface, waar naar toe een gebruiker een collectie van afbeeldingen kan uploaden en waarop deze gebruiker een response krijgt van welke klasse elke afbeelding is. Daarnaast kan de gebruiker de voortgang van het systeem volgen. De viewpoint die hierbij gehanteerd wordt is, de Interactor viewpoint. Hierbij wordt de focus gelegd op hoe mensen en andere systemen interacteren met het te ontwikkelen systeem. Deze viewpoint kan dan nog verder gespecificeerd worden in de Engineering viewpoint en de Systems viewpoint.

2 Systeem stakeholders en requirements

2.1 Stakeholders

Dit project kent verschillende stakeholders, een stakeholder is iemand/een groep die belang of interesse heeft in het project. De verschillende stakeholders zijn de volgende groepen:

ToolMax

ToolMax heeft opdracht gegeven voor het ontwikkelen van het systeem en zal de eindgebruiker zijn, het systeem moet voor hun tijd besparing en efficiëntie opleveren. Dus ToolMax is zowel de opdrachtgever als de gebruiker, ToolMax is hierdoor een van de belangrijkere stakeholders.

Quintor

Quintor is de ontwikkelaar van het systeem, Quintor zal het systeem ontwikkelen van het begin tot het einde.

Bezoekers ToolMax

Doordat ToolMax makkelijker en sneller producten kunnen bezoekers van ToolMax sneller nieuwe producten verwachten en vinden op ToolMax. De bezoekers hebben dus belang dat het systeem ontwikkeld wordt.

2.2 Requirements

2.2.1 Functional requirements

Requirement	**Beschrijving**	**prioritering**
R01	De convolutional neural network wordt beschikbaar gesteld doormiddel van een docker container	M
R02	Doormiddel van verschillende api endpoints kan de convolutional neural network benaderd/aangestuurd worden	M
R03	Het systeem beschikt over een webinterface waardoor de gebruiker kan communiceren met de convolutional neural network	M
R04	Een gebruiker kan een zip file met afbeeldingen uploaden, via de webinterface, om deze te laten classificeren	S
R05	Het systeem zal na uploaden van het .zip bestand de gebruiker een url presenteren waarmee de gebruiker de voortgang kan monitoren en de (sub)resultaten kan downloaden	S
R06	De gebruiker kan in de monitorings overzicht zien in welke fase het systeem zich bevindt	M
R07	De gebruiker kan in de monitorings overzicht zien welke afbeeldingen niet verwerkt konden worden en waarom	S
R08	De gebruiker kan in de monitorings overzicht de (sub)resultaten downloaden	M
R09	Het systeem accepteert .jpeg en .png bestanden	M
R10	De minimale dimensie van de afbeeldingen is 255 x 255 px	M
R11	Het systeem transformeert de afbeeldingen zodat deze voldoen aan de eisen van het convolutional neural network	M
R12	Wanneer het systeem een afbeelding niet kan verwerken zal het systeem dit terug koppelen naar de monitorings overview	S
R13	Het systeem houd per batch de verschillende classificaties bij	M
R14	De gebruiker kan de voorgaande classificaties terug bekijken in een overzicht per batch	M

R15	De gebruiker kan in de classificatie overzicht per batch aangeven wanneer een afbeelding onjuist geclassificeerd is	M
R16	De gebruiker kan de juiste classificatie aangeven wanneer hij/zij heeft aangegeven dat een afbeelding incorrect geclassificeerd is.	M
R17	Het systeem zal de afbeelding toevoegen aan de trainingset, wanneer de gebruiker de juiste classificatie heeft aangegeven	M
R18	Het systeem zal zichzelf wekelijks (opnieuw) trainen	M
R19	Het systeem moet minstens een test nauwkeurigheid hebben van 90%	M
R20	Het systeem mag hoogstens een top 1 error rate hebben van 50%	M
R21	Het systeem mag hoogstens een top 3 error rate hebben van 25%	M
R22	Het systeem mag hoogstens een top 5 error rate hebben van 12,5%	M
R22	Wanneer het systeem na het trainen onder de grens van de testnauwkeurigheid valt zal deze een voorgaande checkpoint gebruiken die wel voldeed aan deze grens	S
R23		

2.2.2 Non-Functional requirements

Requirement	**Beschrijving**	**prioritering**
NFR01	Het systeem moet een batch van 1000 bestanden binnen 5 min geclassificeerd hebben	S
NFR02	Het systeem moet, wanneer het getraind is, telkens dezelfde resultaten terug geven	M
NFR03	Het systeem bewaart de voorgaande 5 classificatie batch resultaten	M

2.3 Scenario's

Situatie	**Beschrijving**
Naam	Initialiseren classificatie proces
ID	S01
Requirements	R03 + R04 + R05 + R09 + R10 + R11
Beschrijving	Gebruiker initialiseren het classificatie proces door middel van het uploaden van een .zip file.
Primaire acteur	De gebruiker
Secundaire acteur	-
Pre-condities	-+
Main flow	1. Gebruiker upload een .zip file met afbeeldingen 2. Het systeem transformeert de afbeeldingen 3. Het systeem geeft de huidige status terug 4. Het systeem slaat de afbeeldingen op 5. Het systeem geeft de huidige status terug 6. Het systeem genereert een batch nummer 7. Het systeem geeft de huidige status terug 8. Het systeem geneert een monitorings url en verzend deze naar de gebruiker
Post-condities	Er is een batch nummer, de bestanden zijn opgeslagen, er is een monitorings url gegenereerd

Alternatieve flow	AF 1: Afbeelding is onleesbaar 1. Het systeem geeft een foutmelding in de monitorings overzicht 2. De scenario gaat verder op MF 2 AF 2: Afbeelding is van onjuiste minimale dimensie 1. Het systeem geeft een foutmelding in de monitorings overzicht 2. De scenario gaat verder op MF 2

Situatie	**Beschrijving**
Naam	Classificatie proces
ID	S02
Requirements	R01 + R02 + R05
Beschrijving	Het classificatie proces wordt uitgevoerd door het systeem
Primaire acteur	Het systeem
Secundaire acteur	-
Pre-condities	De afbeeldingen zijn opgeslagen, er is een batch nummer
Main flow	1. Het systeem stuurt elk afbeelding naar de cnn 2. De cnn stuurt de classificatie terug naar het systeem 3. Het systeem geeft de huidige status terug 4. Het systeem slaat de classificatie op in de DB bij de juiste afbeelding en batchnummer
Post-condities	De DB bevat classificaties voor de huidige batch
Alternatieve flow	-

Situatie	**Beschrijving**
Naam	Overzicht van het proces weer geven
ID	S03
Requirements	R06 + R07 + R08
Beschrijving	Een gebruiker kan de voortgang van het systeem terugzien in het monitorings weergave en de (sub)resultaten downloaden
Primaire acteur	De gebruiker
Secundaire acteur	Het systeem
Pre-condities	Er is een batchnummer
Main flow	1. Het systeem stuurt de gebruiker een url waar hij/zij de monitoring kan volgen 2. De gebruiker ziet de verschillende stappen waar het systeem geweest is 3. De gebruiker ziet in welke stap het systeem zich nu bevindt 4. De gebruiker kan zien welke afbeeldingen niet verwerkt kunnen worden 5. De gebruiker kan (sub)resultaten downloaden

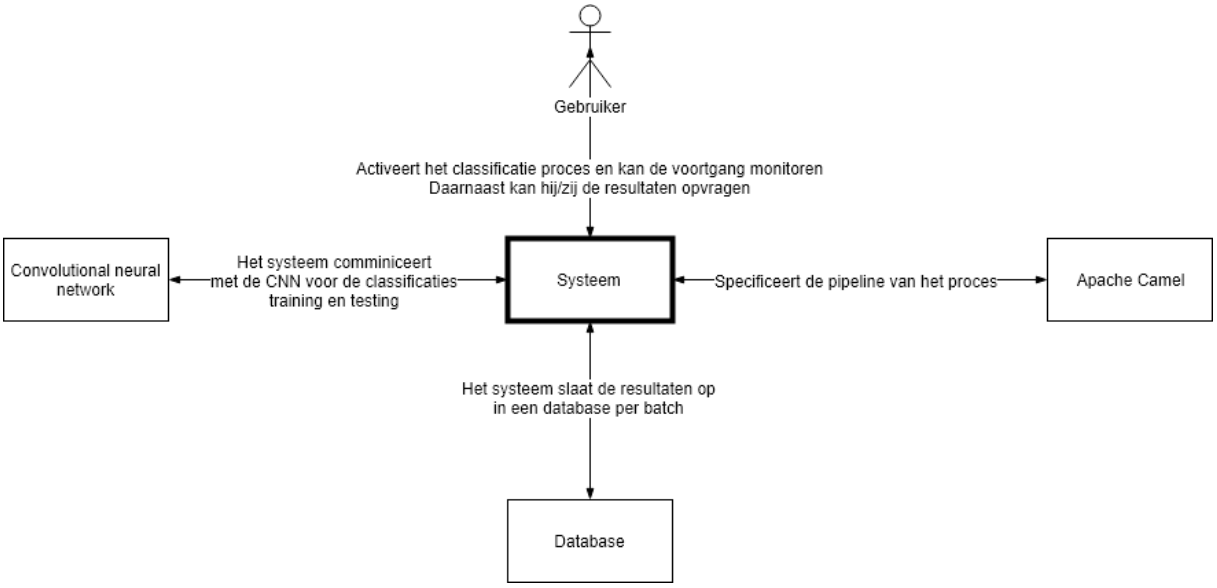
Post-condities	Json met monitoring data, (sub) resultaten in de db
Alternatieve flow	-

Situatie	**Beschrijving**
Naam	Classificatie proces
ID	S04
Requirements	R013 + R14 + R15 + R16 + R17
Beschrijving	De gebruiker kan aangeven of een afbeelding correct of incorrect geclassificeerd is en daarnaast kan het aangeven wat het daadwerkelijke categorie moet zijn
Primaire acteur	De gebruiker
Secundaire acteur	Het systeem
Pre-condities	De volledige batch is geclassificeerd
Main flow	1. De gebruiker gaat naar het classificatie overzicht 2. De gebruiker kiest de batch uit waar foute classificaties instaan 3. Het systeem geeft een overzicht met afbeeldingen 4. De gebruiker geeft aan welke afbeeldingen foutief geclassificeerd zijn 5. De gebruiker geeft de afbeeldingen de juiste classificatie 6. Het systeem voegt de afbeeldingen toe aan de juiste categorieën in de trainings data
Post-condities	Afbeeldingen zijn toegevoegd aan de trainings data
Alternatieve flow	-

3 Architectuur views

3.1 Context View

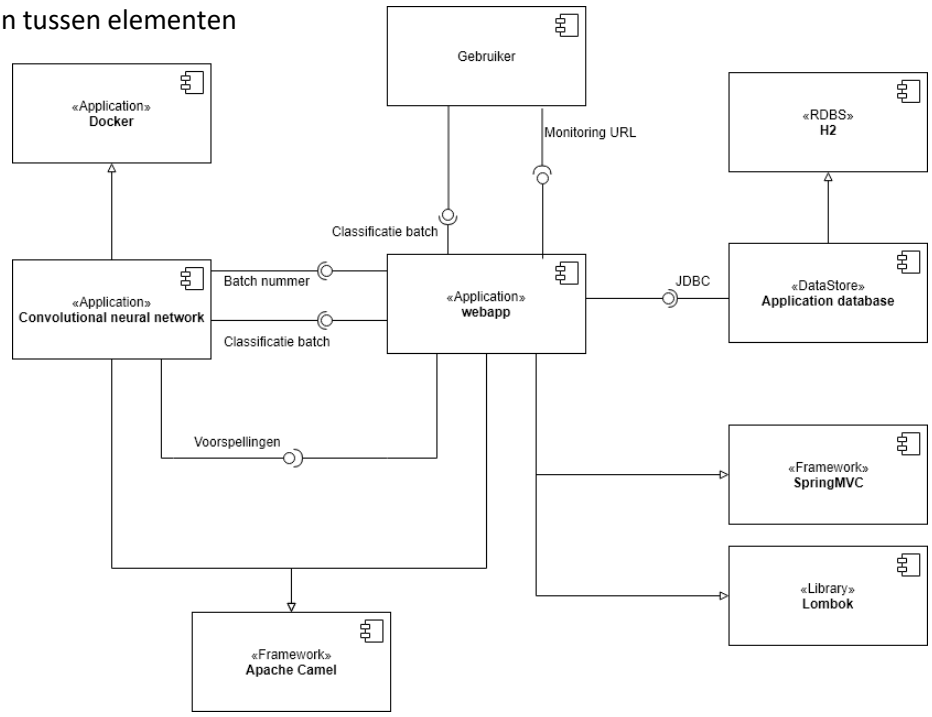
De contextweergave van het systeem beschrijft de relaties, afhankelijkheden en interacties tussen het systeem en zijn omgeving (de mensen, systemen en externe entiteiten waarmee het communiceert).



Het systeem communiceert met de CNN d.m.v. een API, met deze API kan het systeem de CNN aansturen en resultaten ontvangen. Doormiddel van Apache Camel zal het systeem de pipeline beschrijven voor de datatransformatie, manipulatie en de daadwerkelijke classificatie. In de database worden de verschillende resultaten per batch opgeslagen.

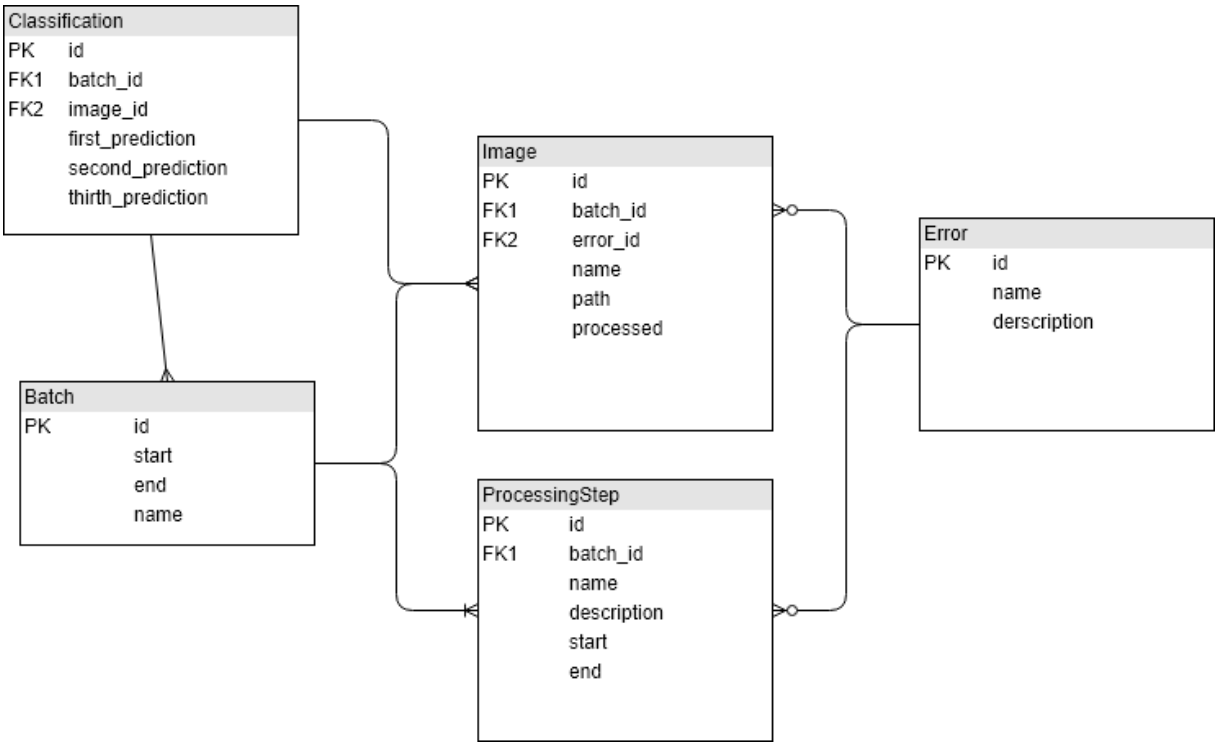
3.2 Functional view

De functionele weergave van het systeem definieert de architectonisch belangrijke functionele elementen van het systeem, de verantwoordelijkheden van elk, de interfaces die ze aanbieden en de afhankelijkheden tussen elementen



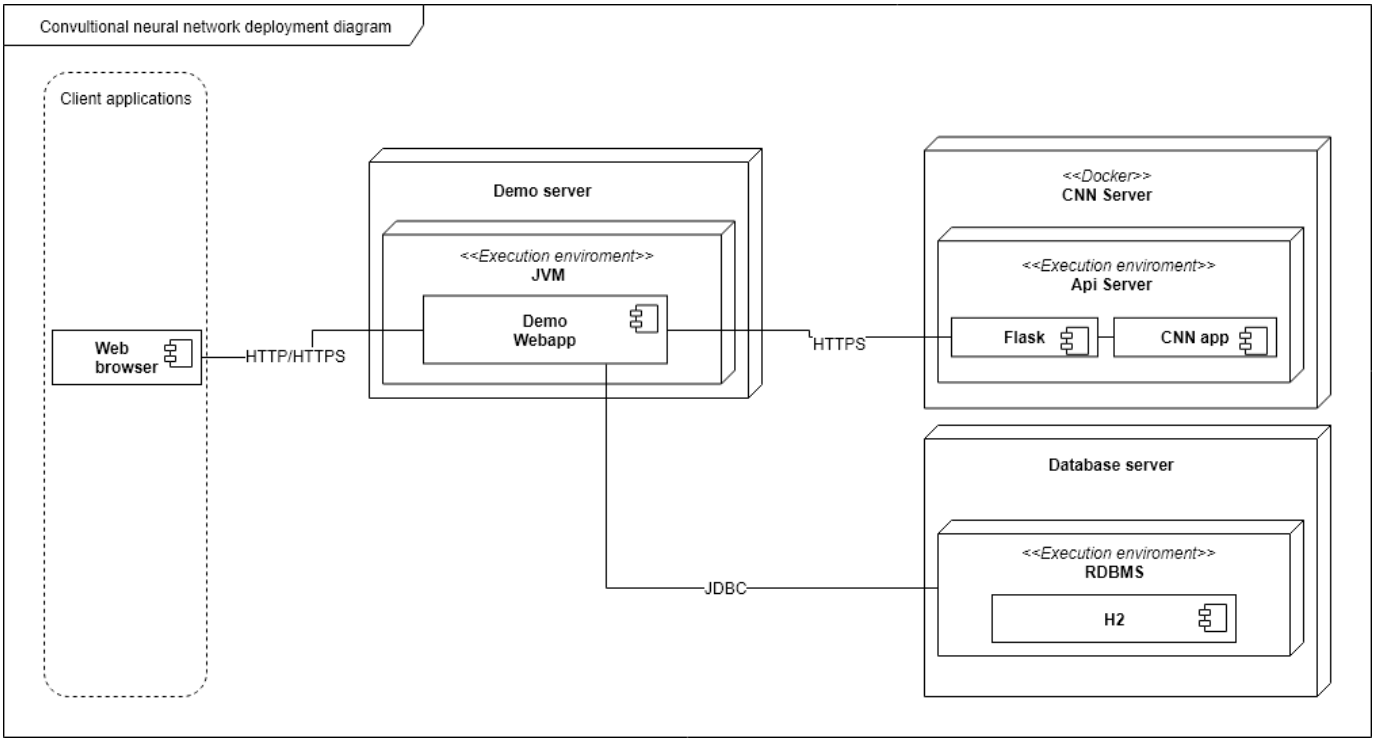
3.3 Information view

De informatieweergave van het systeem definieert de structuur van de opgeslagen informatie (bijvoorbeeld databases en berichtschema's).



3.4 Physical view

De implementatieweergave van het systeem definieert de belangrijkste kenmerken van de operationele inzetomgeving van het systeem. Deze weergave bevat de details van de verwerkingsknooppunten die het systeem nodig heeft voor de installatie (dat wil zeggen het runtime-platform), de softwareafhankelijkheden op elk knooppunt (zoals vereiste bibliotheken) en details van het onderliggende netwerk dat het systeem nodig heeft.

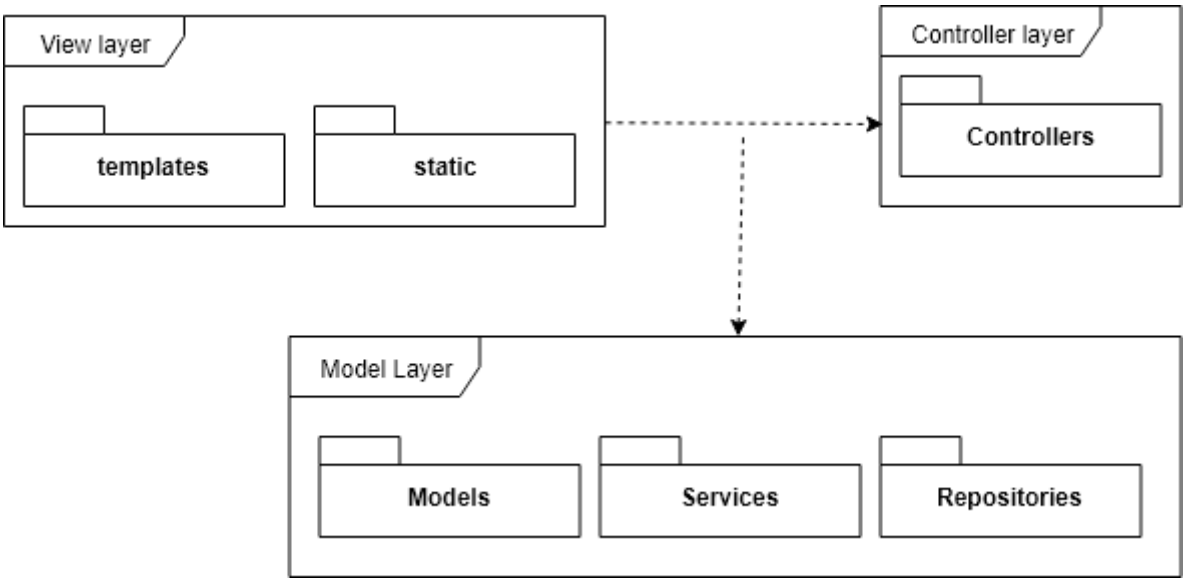


3.4.1 Software dependencies

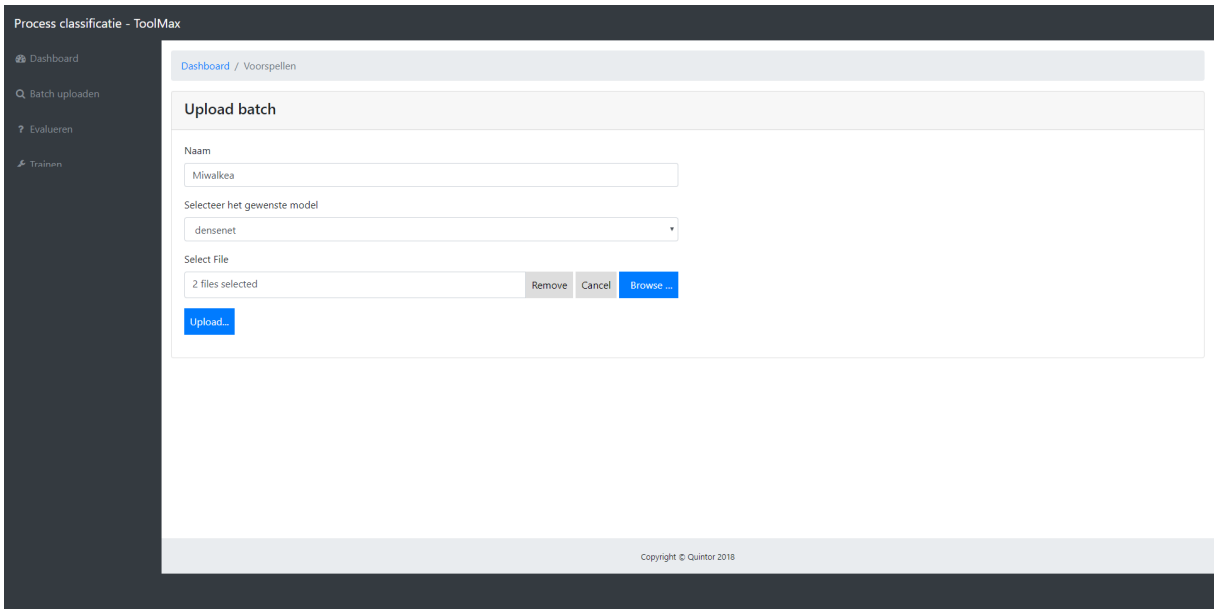
Node	**Software dependency**
Demo server	Spring boot
	Lombok
	H2
	Apache Camel
	H2
	Junit
CNN server	Flask
	Tensorflow
	OpenCV
	scikit-learn

3.5 Development view

De ontwikkelingsweergave van het systeem definieert eventuele beperkingen van het software-ontwikkelproces die vereist zijn door de architectuur. Dit omvat de moduleorganisatie van het systeem, gemeenschappelijke verwerking die alle modules moeten implementeren, elke vereiste standaardisatie van ontwerp, codering en testen en de organisatie van de codeline van het systeem.

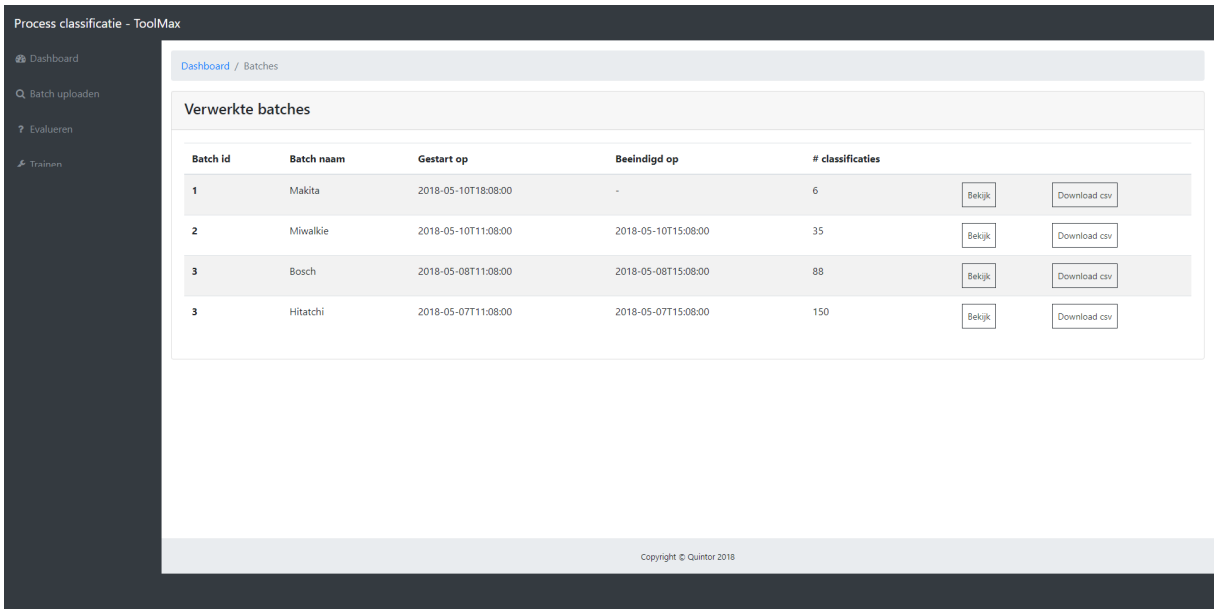


N | Upload interface



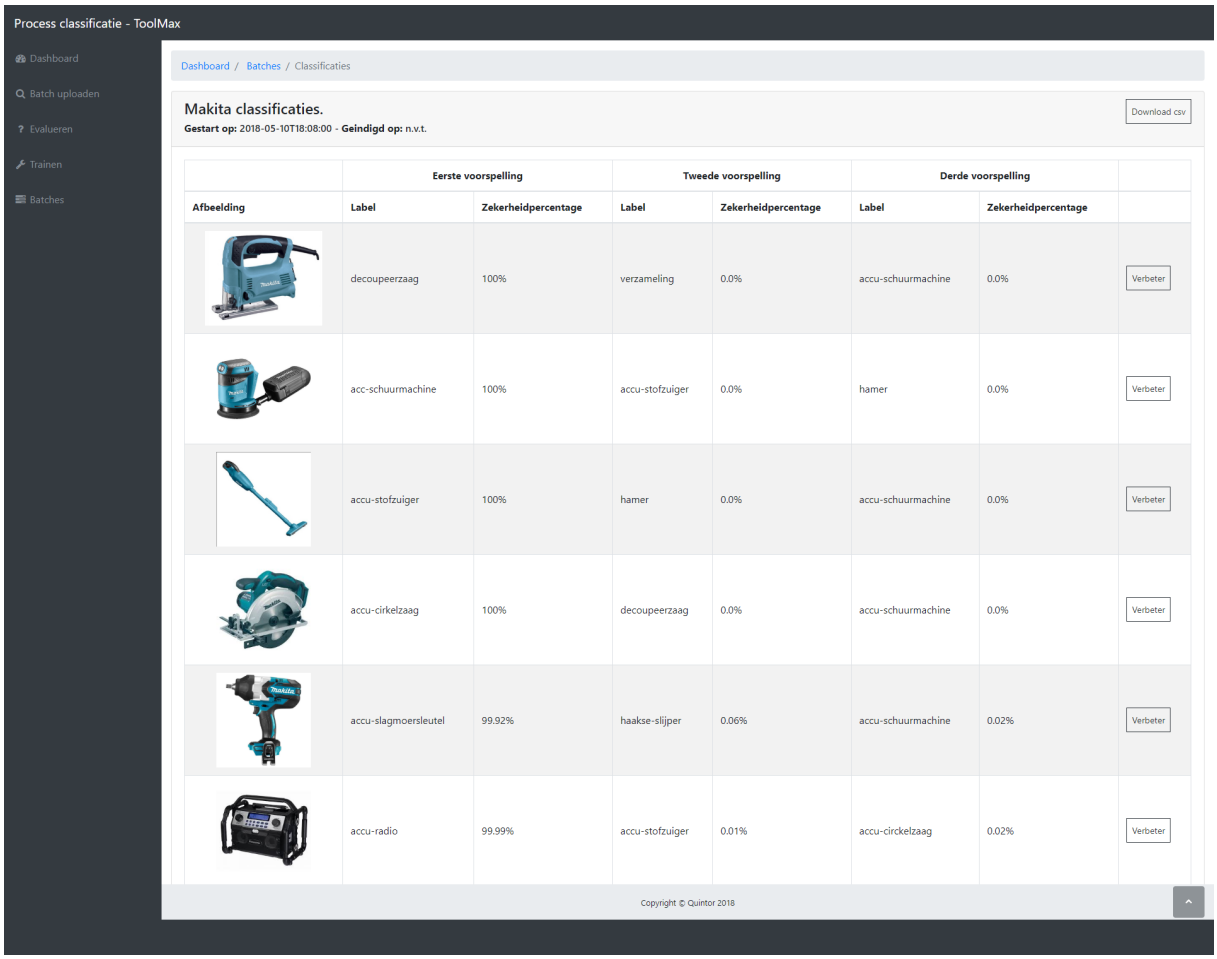
Figuur N.1: Upload interface

O | Overzicht batches



Figuur O.1: Overzicht batches

P | Overzicht classificaties



Figuur P.1: Overzicht classificaties