(Gedeeltelijke) automatisering van de controle bij substitutiescannen

Extended abstract

Datum	**13-06-2023**	Eigenaar	**Finn Alberts**
Auteur	**Finn Alberts**	Status	**Definitief**
Documenttype	**Eindrapport**	Opleiding	**HBO-ICT, Zuyd Hogeschool**
Classificatie	**Publiek**	Begeleiders	**Ing. Marc Polmans,**
Distributie	**Intern, mag extern**		**Drs. Marc Bertrand**
Versie	**1.0**		

Abstract

Archive-IT is een archiverings- en digitaliseringsbedrijf met hun hoofdkantoor gevestigd in Reuver en biedt hun klanten onder andere substitutiescannen aan. Bij substitutiescannen wordt een dossier gedigitaliseerd, waarna het originele fysieke dossier wordt vernietigd. De rechtsgeldigheid gaat daarbij over op de digitale versie. De kwaliteit van deze digitale versie is daarbij van groot belang. Daarom wordt hierbij een nacontrole uitgevoerd.

Momenteel wordt deze controle met de hand uitgevoerd, wat een tijdrovend proces is. Binnen dit project is met behulp van het Design Science Research framework van Hevner een prototype gerealiseerd voor het (gedeeltelijk) automatiseren van deze nacontrole. Dit prototype detecteert visuele fouten en maakt daarnaast een vergelijking met het originele dossier wat een tweede keer wordt gescand om de volledigheid en chronologische volgorde te controleren.

Voor de ontwikkeling hiervan is middels een rapid review gezocht naar geschikte technieken voor het inhoudelijk vergelijken van afbeeldingen. Dit heeft een achttal technieken in kaart gebracht. Deze zijn geëvalueerd op performance en accuraatheid. Op basis van deze evaluatie is een architectuurontwerp gemaakt middels UML-diagrammen, waarbij voor een microkernel-architectuur is gekozen. Aan de hand van dit ontwerp is het prototype gerealiseerd, wat middels een performance- en accuraatheidstest is geëvalueerd.

Aan de hand van de resultaten van de tests kan worden gesteld dat het prototype de (gedeeltelijke) automatisering van de nacontrole realiseert.

Introductie

Aanleiding

Archive-IT is een archiverings- en digitaliseringsbedrijf met hun hoofdkantoor gevestigd in Reuver. Het bedrijf biedt een combinatie van zowel fysiek als digitaal archiveren. Ook biedt zij de mogelijkheid voor digitaliseren on demand, wat betekent dat een dossier pas gedigitaliseerd wordt als het wordt opgevraagd. Daarnaast focust het bedrijf zich de laatste jaren ook steeds meer op vitalisatie, waarbij data wordt omgezet in informatie. Hierbij kan worden gedacht aan het (automatisch) classificeren van documenten of het leggen van verbanden tussen verschillende documenten. Vaak worden hier technieken uit de hoek van artificial intelligence (AI) voor gebruikt. Klanten van Archive-IT zijn onder andere het Slingeland Ziekenhuis, gemeente Roermond en Porsche Groep Zuid. (Archive-IT, sd-a)

De missie van Archive-IT is om bedrijven te helpen om data eenvoudiger tot informatie te verwerken en daarmee de samenwerking en efficiëntie binnen de organisatie te verbeteren. Haar visie is om “in 2025 de meest complete en vertrouwde archiveringsspecialist in West-Europa te zijn die extra toegevoegde waarde kan bieden aan haar klanten in de ondersteuning bij de evolutie van data naar informatie en van werken naar samenwerken” (Archive-IT, sd-b).

Een van de digitaliseringsprocessen die binnen Archive-IT wordt ingezet is het substitutieproces (ook wel substitutiescannen genoemd). Hierbij worden papieren documenten/dossiers gedigitaliseerd, waarna de papieren versies worden vernietigd. De rechtsgeldigheid gaat daarbij over op de nieuwe digitale versie (Archive-IT, 2020). Omdat de papieren versie wordt vernietigd, is de kwaliteit van de scan van groot belang (Velde, et al., 2016). Er kan immers niet op de papieren versie worden teruggevallen. Daarom vindt een uitgebreide controle plaats, nadat de dossiers zijn gedigitaliseerd. Hierbij wordt het papieren document/dossier vergeleken met de digitale versie om hierin verschillen te vinden. Momenteel moet deze controle met de hand worden uitgevoerd voor iedere pagina binnen een steekproef, wat een zeer tijdrovend proces is. De wens vanuit Archive-IT is daarom om dit proces middels een applicatie (gedeeltelijk) te automatiseren.

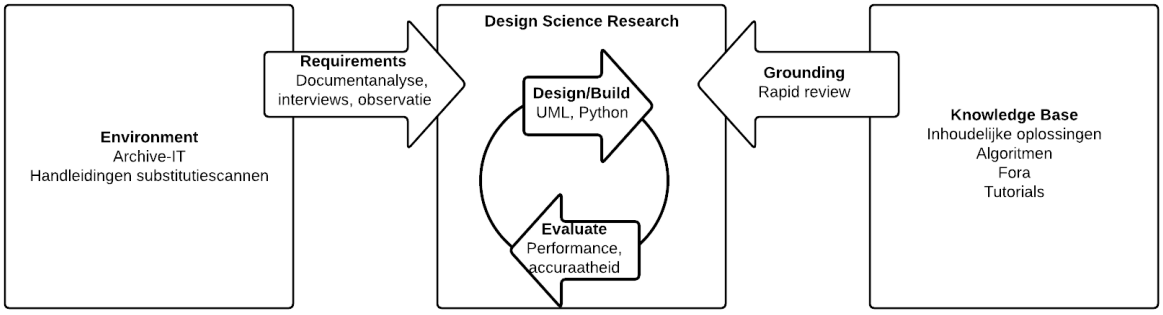
De belangrijkste stakeholders hierbij zijn de manager en de teamleider van de serviceafdeling van Archive-IT, waar het substitutiescannen plaatsvindt. Daarnaast is de eindverantwoordelijke voor de ontwikkeling van nieuwe producten een belangrijke stakeholder.

Doelstelling

De doelstelling van dit project is het (deels) automatiseren van het controleproces van het gedigitaliseerde document/dossier en de vergelijking met het fysieke document/dossier. Bij de aanvang van het project waren er geen verdere randvoorwaarden.

Methode

Voor de uitvoering van dit project is gebruik gemaakt van het Design Science Research framework (DSR-framework) van Hevner (Hevner, 2007). Er is van dit framework gebruik gemaakt, vanwege de onzekerheden die er binnen dit project zijn, zoals het vinden van welke oplossingsinhoudelijke technieken geschikt zijn. Het DSR-framework van Hevner biedt daarin handvaten voor het gestructureerd uitvoeren van ontwerpgericht onderzoek met een artefact (prototype) als eindresultaat. In Figuur 1 is te zien hoe binnen dit project het DSR-framework is toegepast.



Figuur 1 Toepassing van Hevner

Als eerste stap om tot een prototype te komen, zijn requirements verzameld vanuit de environment. Hiervoor zijn allereerst handleidingen voor substitutiescannen van gemeentes en het Nationaal Archief (Velde, et al., 2016) bestudeerd middels een documentanalyse. Deze handleidingen bevatten vaak een hoofdstuk of paragraaf met aandachtspunten waar tijdens de nacontrole op gelet moet worden en vormen daarmee een basis voor de requirements. Naar dit hoofdstuk of deze paragraaf is in de documenten gezocht, waarna de aandachtspunten zijn geëxtraheerd.

Vervolgens zijn ongestructureerde interviews gehouden met de betrokken stakeholders. Bij deze interviews zijn vooraf geen vragen opgesteld, maar staan wel een aantal onderwerpen centraal (Merkus, 2022). Deze onderwerpen zijn te zien in Bijlage A – Onderwerpen interviews. Er is gekozen voor deze manier van interviewen om een breed beeld te verkrijgen van de nacontrole en waaraan het artefact moet voldoen. Ook biedt deze manier van interviewen de mogelijkheid tot diepgang, middels het stellen van vervolgvragen. Omdat de stakeholders direct betrokken zijn bij het substitutiescannen kunnen deze daardoor waardevolle input geven voor de requirements.

Als laatste stap in de requirementselicitatie is het proces van de nacontrole bij substitutiescannen geobserveerd middels een ongestructureerde observatie. Bij deze vorm van observeren wordt een verhalend verslag geschreven en kunnen tijdens de observatie aanvullende vragen worden gesteld (Dingemanse, 2023). Met deze observatie wordt voorkomen dat requirements worden gemist die voor de betrokkenen erg vanzelfsprekend zijn, maar dat voor de buitenwereld niet zijn.

Om de opgehaalde requirements te prioriteren wordt gebruik gemaakt van de Bubble Sort Technique (Hudaib, Masadeh, Qasem, & Alzaqebah, 2018). Deze techniek prioriteert de requirements op een manier

welke vergelijkbaar is met het Bubble Sort sorteeralgoritme. Er is voor deze techniek gekozen, omdat de requirements qua prioriteit dicht op elkaar zullen liggen. Hierdoor is een categorische prioriteringsmethode, zoals bijvoorbeeld de MoSCoW-methode (Hudaib, Masadeh, Qasem, & Alzaqebah, 2018) niet geschikt.

Tijdens de interviews en de observatie is tevens achterhaald welke structuren[1] de dossiers kunnen hebben. Deze structuren zijn middels wireframes in kaart gebracht.

Nadat de requirements zijn achterhaald is onderzoek gedaan naar mogelijke oplossingen (grounding), middels een rapid review. Hiermee wordt op een systematische wijze praktijkgericht bronnenonderzoek gedaan. De focus van een rapid review ligt op het vinden van onderbouwbare, geschikte oplossingen. Het verschil van een rapid review ten opzichte van een systematic review, is dat de rapid review enkele stappen versimpelt of weglaat. De rapid review is daarmee minder uitputtend dan een systematic review, waardoor deze een kortere doorlooptijd heeft. Ook is een rapid review meer gefocust op het verkrijgen van concrete kennis en inzichten (Cartaxo, Pinto, & Soares, 2020). Deze twee punten vormen de reden dat voor een rapid review is gekozen.

De rapid review focust zich op het vinden van technieken om afbeeldingen inhoudelijk met elkaar te vergelijken, zodat kan worden vastgesteld hoeveel procent van de pagina overeenkomt. Voor deze vergelijking wordt een opname gemaakt van het fysieke dossier, welke wordt vergeleken met de scan van het dossier. Deze vergelijking is cruciaal voor het functioneren van de applicatie, vanwege meerdere redenen:

- Er moet kunnen worden vastgesteld of de juiste pagina van het fysieke dossier met de juiste pagina van de scan wordt vergeleken (synchroniseren);
- Voor de controle op volledigheid is het nodig om vast te stellen of een pagina in het fysieke dossier dezelfde pagina is als de pagina in de scan;
- Voor de controle op paginavolgorde is het noodzakelijk om vast te stellen of een pagina in het fysieke dossier dezelfde pagina is als de pagina in de scan;
- Bij controles op andere soorten fouten is het (vaak) nodig om een vergelijking te maken met het fysieke dossier om te controleren of een afwijking daar ook aanwezig is. Bijvoorbeeld, een zwarte streep op een scan kan zijn ontstaan door een kras op de glasplaat van de scanner, maar kan ook onderdeel zijn van het dossier zelf (vanwege bijvoorbeeld de layout). In dat laatste geval is het dus geen fout.

Door de rapid review af te bakenen tot het vinden van technieken voor de vergelijking, wordt dus de techniek voor de kern van de applicatie onderzocht. Zonder deze techniek is het niet mogelijk de rest van

[1] Met structuur wordt hier de opbouw van het dossier bedoelt. Denk hierbij aan of een dossier bestaat uit allemaal A4-vellen, of bijvoorbeeld uit een map in boekvorm, met pagina's aan zowel de linker- als rechterkant. Ook mogelijk lege pagina's zijn onderdeel van de structuur.

de applicatie te ontwikkelen. Het zoekplan dat voor deze rapid review is gebruikt is te vinden in Bijlage B – Zoekplan.

Indien een bron welke middels het zoekplan wordt gevonden enkel liet zien hoe een techniek kan worden geïmplementeerd, maar niet toelicht hoe deze werkt, is de sneeuwbalmethode (B.D. Owens Library, sd) gebruikt met een diepte van één. De toepassing van de sneeuwbalmethode is niet te zien in de uitvoering van het zoekplan, daar pas tijdens het bestuderen van de bronnen duidelijk werd of deze methode toegepast diende te worden.

Met de kennis en inzichten welke tijdens de rapid review zijn opgedaan, is een ontwerp gemaakt. De focus hierbij ligt op het in kaart brengen van de verschillende componenten van de oplossing, hun relaties en hun functie. In dit ontwerp wordt naast tekstuele toelichting ook gebruik gemaakt van UML-klassendiagrammen, UML-componentdiagrammen en UML-sequentiediagrammen (Bennett, McRobb, & Farmer, 2010). Dit eerste ontwerp is nog globaal van opzet en focust zich op de structuur/architectuur van het prototype.

Op basis van het ontwerp is iteratief het prototype gerealiseerd. Door iteratief te werken, kon sneller een bruikbaar prototype worden gerealiseerd en geëvalueerd, wat vervolgens kon worden uitgebreid. Naast het uitbreiden van het prototype tijdens iedere iteratie, werd ook iedere iteratie het ontwerp voor de specifieke onderdelen/componenten ontworpen en vastgelegd. Binnen de iteraties zijn, na het realiseren van de synchronisatie, ook de controles voor de verschillende specifieke afwijkingen, zoals zwarte strepen, gerealiseerd.

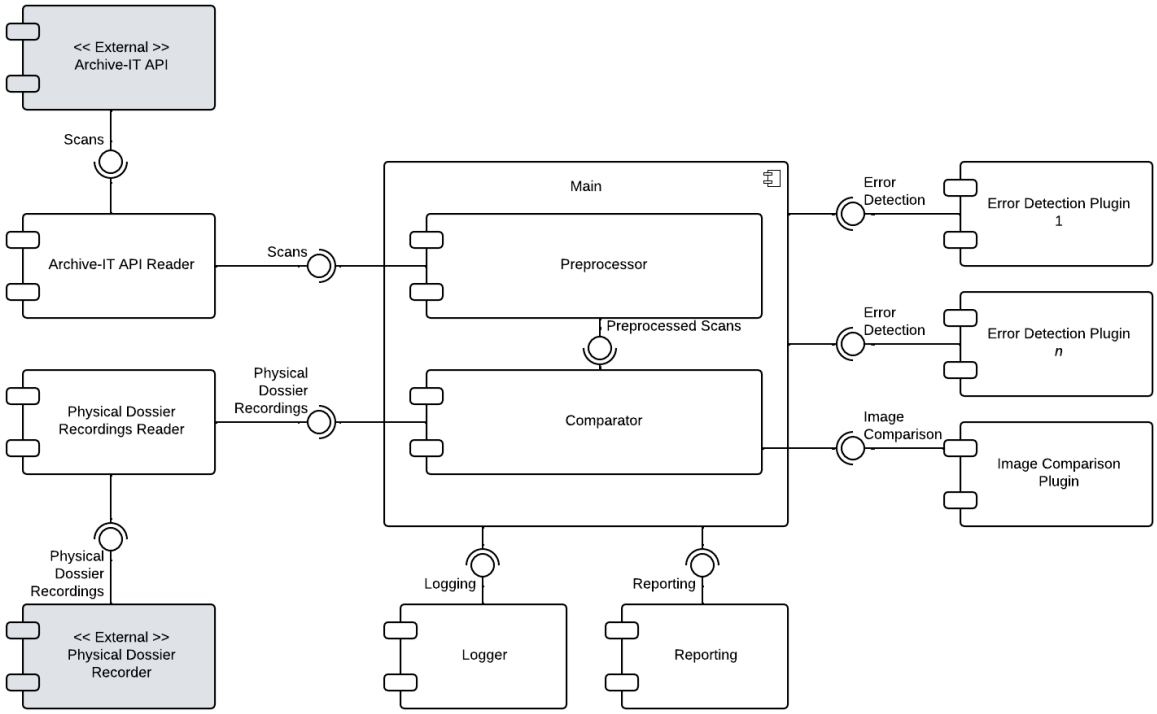
Het prototype is getest op performance en accuraatheid. Bij het testen van de performance wordt de performance gemeten op basis van de log-bestanden die de applicatie genereert. Deze bestanden bevatten ook de tijdstippen van de losse acties van de applicatie. Voor het testen van de accuraatheid worden de resultaten van een handmatige controle vergeleken met de resultaten van de geautomatiseerde controle (voor dezelfde batch[2]). Hierbij wordt een confusion matrix[3] opgesteld en worden daarnaast de percentages voor type I en type II fouten[4] berekend.

[2] Een controle wordt uitgevoerd op batchniveau. Een batch bestaat uit meerdere dossiers.
[3] Een confusion matrix zet de predicted positives en negatives uit tegen de actual positives en negatives in een 2x2 tabel (Chelliah, 2020).
[4] Een type I fout, ook false positive genoemd, wil zeggen dat een fout op een pagina wordt gedetecteerd, terwijl dit in werkelijkheid geen fout is. Een type II fout, ook false negative genoemd, wil zeggen dat een fout op een pagina niet wordt gedetecteerd. (Bhandari, 2021)

Resultaten

Beschrijving artefact

Het gerealiseerde artefact is een in Python ontwikkelde applicatie voor het automatisch controleren van de gescande dossiers. Hiervan is in Figuur 2 het UML-componentdiagram (Bennett, McRobb, & Farmer, 2010) weergegeven.



Figuur 2 UML-componentdiagram van de applicatie

In het grijs worden de *Archive-IT API* en de *Physical Dossier Recorder* weergegeven. Dit zijn externe componenten. Via de *Archive-IT API* kunnen de gedigitaliseerde dossiers worden opgehaald. Dit zijn dus de dossiers waarvoor de nacontrole moet worden uitgevoerd. De *Physical Dossier Recorder* is hierin een bulkscanner[5], waarmee het dossier een tweede keer wordt gescand, zodat deze kan worden vergeleken. De bulkscanner is een andere bulkscanner dan de bulkscanner, waarmee het dossier gedigitaliseerd is. Hiermee wordt voorkomen dat fouten worden verworpen, omdat afwijkingen door defecten in de scanner worden gecreëerd[6]. De tweede bulkscanner mag van hetzelfde merk/type zijn als de eerste bulkscanner, maar dit is niet noodzakelijk.

[5] Een bulkscanner, ofwel productiescanner, is een scanner waar grote hoeveelheden pagina's op hoge snelheid door kunnen worden gescand (Canon Nederland, sd).

[6] Voorbeeld: de glasplaat van de scanner bevat een kras wat resulteert in een zwarte streep op de scan. Indien tweemaal met dezelfde scanner wordt gescand zal de applicatie de fout verwerpen, omdat de streep zowel op het gedigitaliseerde dossier als de scan van het fysieke dossier voorkomt. De applicatie gaat er dan namelijk vanuit dat de streep onderdeel is van het dossier.

Een volledige controle vindt in twee stappen plaats. De *Preprocessor* verwerkt eerst alle gedigitaliseerde dossiers en controleert daarbij op alle soorten fouten, waarbij het fysieke dossier niet noodzakelijk is, middels verschillende *Error Detection Plugins*. De *Comparator* vergelijkt vervolgens, onder toezicht van de gebruiker, de scans uit de *Archive-IT API* met de scans uit de *Physical Dossier Recorder* om de gevonden fouten uit de *Preprocessor* te verifiëren en fouten te detecteren, waarvoor het fysieke dossier noodzakelijk is. Beide stappen genereren een rapport met daarin de gevonden fouten. Zie ook Figuur 6 in Bijlage F – Ontwerpen op architectuurniveau.

Fouten waarop gecontroleerd wordt zijn:

- Compleetheid;
- Paginavolgorde;
- Verkeerde resolutie;
- Zwarte randen;
- Zwarte strepen;
- Omgevouwen hoeken;
- Scheve pagina's;
- Doorschijnpagina's;
- Pagina's in een verkeerde leesrichting.

Requirements

Binnen de documentanalyse van de handleidingen voor substitutiescannen van gemeentes en het Nationaal Archief (Velde, et al., 2016) zijn 37 documenten geanalyseerd. Dit heeft geleid tot dertig aandachtspunten.

Deze dertig aandachtspunten zijn tijdens de interviews met de drie betrokken stakeholders besproken. Hierbij is besloten om vijftien aandachtspunten tot requirements te verwerken. De overige vijftien aandachtspunten zijn niet te controleren middels een geautomatiseerde controle of zijn niet relevant in de context van een nacontrole. Middels de interviews, een observatie en één e-mail van een stakeholder zijn nog 48 requirements opgehaald. Het totaal aantal requirements komt daarmee op 63. De requirements zijn te vinden in Bijlage C – Requirements.

Volgens requirement N14 dient de applicatie rekening te houden met de verschillende structuren van de dossiers tijdens de controle. Deze structuren zijn schematisch weergegeven in Bijlage D – Structuren van dossiers.

Grounding

Op basis van de requirements is duidelijk geworden dat compleetheid en paginavolgorde de belangrijkste controles zijn. Voor visuele fouten zoals strepen is het daarnaast nodig om de fout te verifiëren. Daarvoor

is het nodig om vast te stellen of de juiste pagina van de scan met de juiste pagina van het fysieke dossier wordt vergeleken. Er is daarom, zoals beschreven in de aanpak, middels een rapid review gezocht naar technieken om afbeeldingen inhoudelijk met elkaar te vergelijken, om vast te stellen of deze overeenkomen. Op basis van het zoekplan zijn dertien geschikte bronnen gevonden (zie ook Bijlage E – Totstandkoming bronnen rapid review).

Op basis van deze dertien bronnen zijn acht vergelijkingstechnieken gevonden, zie Bijlage G – Vergelijkingstechnieken. Twee van de gevonden technieken kijken enkel naar kleurverdeling en zijn daarmee op zichzelf niet geschikt voor een volledige vergelijking. Wel zijn ze mogelijk geschikt als aanvulling op een ander algoritme.

Elk van deze acht technieken zal in de praktijk moeten worden getest, zodat ook kan worden getest of een techniek een goede vergelijking kan maken voor de documenten. Daarnaast zal in deze praktijktesten ook moeten worden gekeken of een techniek aan niet-functionele requirements, zoals performance, voldoet.

Architectuur

Aan de hand van de requirements wordt duidelijk dat er veel verschillende soorten fouten kunnen voorkomen. Daarnaast is het vanuit de requirements vereist om de controle op te splitsen in twee stappen: voorverwerken (preprocessen) en vergelijken (zie requirement N15 in Bijlage C – Requirements). Ook vanuit een performanceperspectief, waarbij de tijd per pagina tijdens de vergelijking beperkt is tot slechts 1 seconde (zie requirement N8 in Bijlage C – Requirements), heeft het voordelen om alles wat vooraf kan worden gedaan ook vooraf te doen.

Daarnaast wordt aan de hand van de rapid review duidelijk dat er voor de inhoudelijke vergelijking een aantal verschillende technieken bestaan. Op basis van een performance- en accuraatheidstest voor de verschillende technieken (zie Bijlage H – Accuraatheid per vergelijkingstechniek en Bijlage I – Performance per vergelijkingstechniek) kan worden gesteld dat de multi-scale structural similarity index (MS-SSIM) (Wang, Simoncelli, & Bovik, 2003) het meest geschikt is. Echter, vanwege de grote rol die zowel performance als accuraatheid spelen (zie requirement N7 en N8 in Bijlage C – Requirements), is gebleken dat de techniek geoptimaliseerd moet worden om aan beide requirements te voldoen.

Voor het maken van de vergelijking met het fysieke dossier, moet hiervan eerst een opname worden gemaakt. Hiervoor bestaan verschillende opties, zoals een hoge resolutie webcam (Logitech, sd) of CZUR boekenscanner (CZUR, sd), waarbij voor de camera door het dossier kan worden gebladerd. Daarnaast is ook het gebruikmaken van een tweede bulkscanner een optie.

Op basis van bovenstaande aspecten is een ontwerp op architectuurniveau gemaakt, bestaande uit een aantal UML-diagrammen (Bennett, McRobb, & Farmer, 2010), zie Bijlage F – Ontwerpen op architectuurniveau. Zoals in de diagrammen te zien is, wordt gebruik gemaakt van een microkernel-

architectuur (Richards & Ford, 2021). Hierbij bestaat de kern uit de *Preprocessor* (voor het voorverwerken) en de *Comparator* (voor de vergelijking[7]).

Omdat er meerdere opties zijn voor het maken van een opname van het fysieke dossier wordt het uitlezen van de opnames uitgewerkt in een losse component, de *Physical Dossier Recordings Reader*. Hiermee wordt het eenvoudiger deze component te vervangen. Dit is een voordeel, daar voorafgaand aan de realisatie nog niet duidelijk is welke optie wordt gekozen voor de *Physical Dossier Recorder*.

Vanwege de nodige optimalisaties aan het MS-SSIM algoritme, is deze in een losse component geplaatst, de *Image Comparison Plugin*. Hiermee wordt het tevens mogelijk om de techniek volledig te vervangen door een andere techniek in de toekomst, indien dit nodig blijkt te zijn.

De detectie voor specifieke soorten fouten wordt afgehandeld door de *Error Detection Plugins*. Hierbij controleert iedere plugin op één specifieke soort fout. Hierdoor wordt het eenvoudig om controles voor de verschillende soorten foutdetectie aan te passen of nieuwe soorten foutdetectie toe te voegen.

Realisatie

Aan de hand van het ontwerp op architectuurniveau is de applicatie in Python gerealiseerd. Er is voor Python gekozen, omdat dit een toegankelijke taal is welke veel in de wereld van artificial intelligence (en daarmee computer vision) wordt gebruikt (Patel, 2022).

De *Physical Dossier Recorder* is gerealiseerd in de vorm van een tweede bulkscanner. Op basis van praktijktesten is gebleken dat een webcam een te lage kwaliteit geeft voor een goede overeenkomst (volgens de MS-SSIM slechts circa 15% overeenkomstig bij overeenkomende pagina's) en de CZUR boekenscanner een te lage performance (circa 3 tot 4 seconden per opname). Ondanks de initiële voorkeur voor een oplossing, waarbij door het fysieke dossier gebladerd kan worden, zodat deze ook nog door menselijke handen gaat, is de tweede bulkscanner dus de beste optie gebleken.

Voor de optimalisatie van de berekening van de MS-SSIM wordt onder andere gebruik gemaakt van een implementatie van dit algoritme in Pytorch (Fang, et al., 2023). Pytorch kan middels CUDA[8] op de videokaart worden uitgevoerd, wat resulteert in een vele malen hogere performance (PyTorch, sd; NVIDIA, sd). Daarnaast worden de gewichten van de verschillende schalen van detail waar de MS-SSIM mee werkt, aangepast naar 5% voor de laagste drie schalen, 35% voor de vierde schaal en 50% voor de vijfde en laatste schaal. Hiermee wordt meer nadruk gelegd op de latere schalen, welke minder op detailniveau kijken door gebruik te maken van een lagere resolutie, waardoor de accuraatheid verbeterd.

[7] Vergelijken betekent in deze niet enkel het vergelijken of de afbeeldingen overeenkomen, maar ook het verifiëren van fouten middels een vergelijking.
[8] CUDA (volledig: Compute Unified Device Architecture) maakt het mogelijk om op videokaarten (GPUs) van NVIDIA code uit te voeren. (NVIDIA, sd)

Verder wordt binnen de *Preprocessor* aan de hand van de verschillende *Error Detection Plugins* op de verschillende foutsoorten gecontroleerd, welke vervolgens binnen de *Comparator* geverifieerd worden. Volledigheid en paginavolgorde worden niet met een *Error Detection Plugin* gecontroleerd, maar binnen de *Comparator*, omdat het onmogelijk is om deze te controleren zonder vergelijking met het fysieke dossier.

Testen

De resultaten van de performancetest zijn in twee onderdelen gesplitst: preprocessen en vergelijken. Beide hebben hun eigen requirement (N8 en N19, zie ook Bijlage C – Requirements). Op basis van de test wordt duidelijk dat voor het preprocessen de gemiddelde tijd 5,232 (± 1,141) seconden per pagina is. Dit is inclusief het downloaden van het dossier vanaf de API. Tijdens het vergelijken is de tijd gemiddeld 0,687 (± 0,274) seconden per pagina. Zie ook Bijlage J – Performancetesten.

Ook de resultaten van de accuraatheidstest zijn gesplitst in de resultaten voor het preprocessen en de resultaten voor het vergelijken (zie requirements N6 en N7 in Bijlage C – Requirements). Bij het preprocessen kan het percentage type I fouten worden vastgesteld op 4,7% en het percentage type II fouten op 0%. Voor het vergelijken is het percentage type I fouten 11,1% en voor type II fouten is het percentage wederom 0%. De type I fouten tijdens het vergelijken zijn voornamelijk pagina's waarbij de applicatie een pagina niet ziet als overeenkomstig en dus rapporteert als zijnde een missende pagina. Zie ook Bijlage K – Accuraatheidstest.

Discussie

Een aspect wat een positieve impact heeft gehad op dit project is het gebruikmaken van meerdere methodes voor het ophalen van de requirements (documentanalyse, interviews, observatie). Hiermee kon een goed compleet overzicht van de requirements worden gecreëerd.

Een van de belangrijke discussiepunten binnen dit project is de onderrepresentatie van fouten in de testdossiers. In de dossiers waarmee is getest zijn slechts vier fouten aanwezig. Dit is een gevolg van de kwaliteit van de door Archive-IT gedigitaliseerde dossiers, welke over het algemeen zeer weinig tot zelfs geen fouten bevatten. Het creëren van een goede representatieve testset is daarmee erg lastig. Bij het vaststellen van het foutpercentage voor type II fouten op absoluut 0% zijn daarom kanttekeningen te plaatsen. Echter, de vier fouten welke in de testdossiers aanwezig waren, zijn ook allen eruit gefilterd.

Een volgend discussiepunt is het hoge percentage type I fouten tijdens het vergelijken (11,1%). Hierbij moet echter worden opgemerkt dat bij iedere gevonden fout tijdens de vergelijking de gebruiker om een bevestiging van de fout wordt gevraagd, waardoor type I fouten direct kunnen worden verworpen. Daarmee is het percentage type I fouten in het eindrapport in de praktijk 0%. Uiteraard is het wel wenselijk vanuit een gebruikersperspectief om het percentage type I fouten zo laag mogelijk te houden.

Een ander punt van discussie is dat de controles voor vlekken en scheuren (requirements F19 en F20, zie Bijlage C – Requirements) niet zijn gerealiseerd, ondanks dat deze een hogere prioriteit hebben dan enkele andere requirements welke wel gerealiseerd zijn. Deze keuze is gemaakt, omdat tijdens de ontwikkeling is gebleken dat voor het detecteren van deze visuele afwijkingen nog geen algoritmes bestaan. Echter, het zou mogelijk kunnen zijn om met behulp van deep learning[9] modellen te trainen welke deze afwijkingen kunnen detecteren. Het is echter niet realistisch binnen de tijdsscope om een dergelijk model te realiseren, mede omdat de hoeveelheid data die hiervoor nodig is momenteel niet beschikbaar is.

De gekozen microkernel-architectuur maakt het prototype echter wel toekomstbestendig en flexibel. Het toevoegen van controles voor nieuwe soorten fouten, zoals voor de vlekken en scheuren, wordt daarmee eenvoudig. Daarnaast is deze architectuur erg geschikt voor het kunnen doorontwikkelen, testen en/of vervangen van bestaande componenten.

Als laatste discussiepunt kan worden genoemd dat de dossiers waarmee de tests zijn uitgevoerd, onderdeel zijn van een regulier digitaliseringsproject, in plaats van een substitutiescanproces, omdat ten tijde van dit project geen testset beschikbaar was uit een substitutiescanproject. Ondanks dat de dossiers geen onderdeel zijn van een substitutiescanproject, komen deze wel zoveel mogelijk overeen met dossiers zoals deze in een substitutiescanproject voorkomen. Ze zijn dus niet volledig representatief, maar komen wel erg in de buurt.

[9] Deep learning is een vorm van kunstmatige intelligentie, waarbij een model wordt getraind op grote hoeveelheden data (IBM, sd).

Conclusie

Tijdens dit project is een automatiseringstool gerealiseerd voor het (gedeeltelijk) automatiseren van de nacontrole bij substitutiescannen. Hierbij kan een gebruiker een gedigitaliseerd dossier controleren op verschillende visuele fouten. Daarnaast kan de volledigheid en paginavolgorde van het gedigitaliseerde dossier worden gecontroleerd met de tool door het dossier een tweede keer te scannen en hiermee een vergelijking uit te voeren. Hiermee wordt de doelstelling, het (deels) automatiseren van het controleproces van het gedigitaliseerde document/dossier en de vergelijking met het fysieke document/dossier, behaald. Door deze automatisering kan een tijdswinst worden geboekt en wordt het nacontroleproces tevens meer geformaliseerd en consistenter.

Kijkende naar de requirements wordt aan slechts 5 van de 33 functionele requirements niet voldaan. De vijf waaraan niet wordt voldaan betreffen requirements voor extra soorten controles. De ontwikkeling van deze foutcontroles zou binnen een vervolgproject kunnen worden uitgewerkt.

Voor de niet-functionele requirements geldt dat aan slechts 3 van de 22 niet wordt voldaan. Dit zijn de drie requirements met de laagste prioriteit. Bij requirement N6, waarin wordt gesteld dat de foutmarge voor type I fouten maximaal 4% mag zijn, moet echter wel een kanttekening worden geplaatst. Ondanks dat de applicatie een foutpercentage van 11,1% heeft voor type I fouten bij de vergelijking, dient een fout altijd door de gebruiker bevestigt te worden. In het eindrapport zullen dus geen type I fouten meer voorkomen.

Aanbevolen wordt om te starten met het gebruik van de tool. Daarbij moet worden geëvalueerd of de tool ook in de praktijk prettig werkt (field testing). Tevens moet worden gemonitord of het percentage type I fouten ook onder de 2,5% blijft (zie requirement N7 in Bijlage C – Requirements), wanneer er een grotere hoeveelheid fouten aanwezig is in de populatie.

Daarnaast wordt aanbevolen om de mogelijkheden voor het doorontwikkelen van de tool te blijven monitoren en daarmee aan de requirements te voldoen, waaraan tijdens dit project nog niet voldaan kon worden. Daarmee zou in de toekomst een nog completere applicatie kunnen worden gerealiseerd.

Verwijzingen

Agrawal, P. (2021, maart 4). *A Beginners' Guide to Image Similarity using Python*. Opgeroepen op maart 8, 2023, van Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/03/a-beginners-guide-to-image-similarity-using-python/

Alberts, F. (2023a, maart 15). *FinnAlberts/Image-comparison-metrics*. Opgeroepen op maart 15, 2023, van GitHub: https://github.com/FinnAlberts/Image-comparison-metrics

Archive-IT. (2020, september 8). *Substitutie scannen binnen de overheid*. Opgeroepen op februari 22, 2023, van Archive-IT: https://www.archive-it.nl/blogs/substitutie-scannen-binnen-de-overheid

Archive-IT. (sd-a). *Homepagina*. Opgeroepen op februari 22, 2023, van Archive-IT: https://www.archive-it.nl/

Archive-IT. (sd-b). *De normen & waarden van Archive-IT*. Opgeroepen op februari 22, 2023, van Archive-IT: https://www.archive-it.nl/onze-organisatie

B.D. Owens Library. (sd). *Snowball Research*. Opgeroepen op maart 8, 2023, van B.D. Owens Library: https://libguides.nwmissouri.edu/snowball

Benard, O. (2022, september 19). *differences-between-two-images (or multiple images)*. Opgeroepen op maart 8, 2023, van GitHub: https://github.com/olivierbenard/differences-between-two-images

Bennett, S., McRobb, S., & Farmer, R. (2010). *Object-Oriented Systems Analysis and Design Using UML 4th Edition*. Maidenhead: McGraw-Hill Higher Education.

Bhandari, P. (2021, januari 18). *Type I & Type II Errors | Differences, Examples, Visualizations*. Opgeroepen op juni 7, 2023, van Scribbr: https://www.scribbr.com/statistics/type-i-and-type-ii-errors/

Canon Nederland. (sd). *Documentscanners*. Opgeroepen op mei 24, 2023, van Canon Nederland: https://www.canon.nl/business/products/scanners/document-scanners/

Cartaxo, B., Pinto, G., & Soares, S. (2020). Rapid Reviews in Software Engineering. *Contemporary Empirical Methods in Software Engineering*, pp. 357-384.

Chelliah, I. (2020, december 23). *Confusion Matrix — Clearly Explained*. Opgeroepen op juni 5, 2023, van Towards Data Science: https://towardsdatascience.com/confusion-matrix-clearly-explained-fee63614dc7

CZUR. (sd). *ET Series Professional Book Scanner*. Opgeroepen op maart 31, 2023, van CZUR: https://shop.czur.com/products/etscanner?variant=39593283289136

DigitalSreeni. (2021, januari 12). *191 - Measuring image similarity in python*. Opgeroepen op maart 8, 2023, van YouTube: https://www.youtube.com/watch?v=16s3Pi1InPU

Dingemanse, K. (2023, februari 27). *Observatie als methode in je scriptie | Uitleg & voorbeelden*. Opgeroepen op maart 7, 2023, van Scribbr: https://www.scribbr.nl/onderzoeksmethoden/observaties/

Ekhtiari, N. (2021, januari 18). *Comparing ground truth with predictions using image similarity measures*. Opgeroepen op maart 8, 2023, van Up42: https://up42.com/blog/image-similarity-measures

Fang, G., Beuttenmüller, F., Pang, Y., Brummer, B., Kim, T. H., Ma, X., . . . One-sixth. (2023, maart 13). *VainF/pytorch-msssim*. Opgeroepen op april 19, 2023, van GitHub: https://github.com/VainF/pytorch-msssim

Gonzalez-Audicana, M., Saleta, J. L., Catalan, R. G., & Garcia, R. (2004, juni). Fusion of multispectral and panchromatic images using improved IHS and PCA mergers based on wavelet decomposition. *IEEE Transactions on Geoscience and Remote Sensing Vol. 42 Iss. 6*, pp. 1291-1299.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems Vol. 19 Iss. 2*, 87-92.

Hudaib, A., Masadeh, R., Qasem, M., & Alzaqebah, A. (2018, januari 13). Requirements Prioritization Techniques Comparison. *Modern Applied Science Vol 12. No. 2*, pp. 62-80.

IBM. (sd). *What is deep learning?* Opgeroepen op juni 7, 2023, van IBM: https://www.ibm.com/topics/deep-learning

Khan, S. A. (2022, september 28). *How to compare two images in OpenCV Python?* Opgeroepen op maart 8, 2023, van TutorialsPoint: https://www.tutorialspoint.com/how-to-compare-two-images-in-opencv-python

Logitech. (sd). *Brio Ultra HD Pro Zakelijke Webcam*. Opgeroepen op april 19, 2023, van Logitech: https://www.logitech.com/nl-nl/products/webcams/brio-4k-hdr-webcam.960-001106.html

Merkus, J. (2022, september 29). *Ongestructureerde of open interviews in je scriptie*. Opgeroepen op mei 12, 2023, van Scribbr: https://www.scribbr.nl/onderzoeksmethoden/ongestructureerd-interview/

Müeller, M., Almeida, R., Ekhtiari, N., Rieke, C., Koskela, A., & Stenius, M. (2023, januari 14). *Image Similarity Measures*. Opgeroepen op maart 8, 2023, van GitHub: https://github.com/up42/image-similarity-measures

Müller, M. U., Ekhtiari, N., Almeida, R. M., & Rieke, C. (2020). *image-similarity-measures 0.3.5*. Opgeroepen op maart 8, 2023, van PyPi: https://pypi.org/project/image-similarity-measures/

NVIDIA. (sd). *CUDA Toolkit*. Opgeroepen op april 19, 2023, van NVIDIA Developer: https://developer.nvidia.com/cuda-toolkit

Patel, U. (2022, februari 17). *Why is Python the best for artificial intelligence and machine learning?* Opgeroepen op maart 7, 2023, van Tristate Technology: https://www.tristatetechnology.com/blog/why-is-python-the-best-for-artificial-intelligence-and-machine-learning

PlsWork, & Lam, N. (2022, mei 7). *Detect and visualize differences between two images with OpenCV Python*. Opgeroepen op maart 8, 2023, van Stack Overflow: https://stackoverflow.com/questions/56183201/detect-and-visualize-differences-between-two-images-with-opencv-python

PyTorch. (sd). *End-to-end machine learning framework*. Opgeroepen op april 19, 2023, van PyTorch: https://pytorch.org/features/

Raval, P. (2021, augustus 18). *Measuring similarity in two images using Python*. Opgeroepen op maart 8, 2023, van Towards Data Science: https://towardsdatascience.com/measuring-similarity-in-two-images-using-python-b72233eb53c6

Richards, M., & Ford, N. (2021). *Fundamentals of Software Architecture An Engineering Approach*. O'Reilly Media.

Rosebrock, A. (2021a, juli 1). *How-To: Python Compare Two Images*. Opgeroepen op maart 8, 2023, van PyImageSearch: https://pyimagesearch.com/2014/09/15/python-compare-two-images/

Rosebrock, A. (2021b, juni 19). *Image Difference with OpenCV and Python*. Opgeroepen op maart 8, 2023, van PyImageSearch: https://pyimagesearch.com/2017/06/19/image-difference-with-opencv-and-python/

Sonkar, J. P. (2023, januari 3). *Measure similarity between images using Python-OpenCV*. Opgeroepen op maart 8, 2023, van GeeksForGeeks: https://www.geeksforgeeks.org/measure-similarity-between-images-using-python-opencv/

Statcounter GlobalStats. (2023, februari). *Search Engine Market Share Worldwide*. Opgeroepen op maart 7, 2023, van Statcounter GlobalStats: https://gs.statcounter.com/search-engine-market-share

Velde, M. v., Karelse, J., Kiens, P., Leeuwen, A. v., Schenk, N., Schouten, D., & Zwagerman, G. (2016). *Handreiking vervanging archiefbescheiden versie 2.0.* Den Haag: Rijksoverheid.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers Vol. 12*, pp. 1389-1402.

Yin, K. (2021, mei 4). *How To Measure Image Similarities in Python*. Opgeroepen op maart 8, 2023, van Better Programming: https://betterprogramming.pub/how-to-measure-image-similarities-in-python-12f1cb2b7281

Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011, augustus). FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing Vol. 20 Iss. 8*, pp. 2378-2386.

Zhou, J., Civco, D. L., & Silander, J. A. (1998). A wavelet transform method to merge Landsat TM and SPOT panchromatic data. *A wavelet transform method to merge Landsat TM and SPOT panchromatic data Vol. 19 Iss. 4*, pp. 743-757.

Bijlage A – Onderwerpen interviews

Onderwerpen voor interview met eindverantwoordelijke voor de ontwikkeling van nieuwe producten:

- Bespreken reeds opgestelde requirements;
- Aanvullende requirements vinden;
- Bespreken lijst van aandachtspunten uit handleidingen van overheidsinstanties.

Onderwerpen voor interview met manager en teamleider serviceafdeling:

- Bespreken reeds opgestelde requirements;
- Aanvullende requirements vinden;
- Specificeren van bepaalde requirements, zoals tijd per pagina.

Bijlage B – Zoekplan

Zoektermen en zoekmachine

Voor het vinden van bronnen wordt gebruik gemaakt van Google als zoekmachine. Er wordt voor Google gekozen, omdat dit wereldwijd de meest gebruikt zoekmachine is met in februari 2023 een marktaandeel van 93,37% (Statcounter GlobalStats, 2023). Op basis hiervan kan worden gesteld dat Google een goede zoekmachine is, welke door veel mensen wordt vertrouwd als het gaat om het vinden van de juiste resultaten.

In Google worden een aantal zoektermen gebruikt voor het vinden van geschikte bronnen. De zoektermen die worden gebruikt zijn:

- Image similarity python;
- Visual image similarity python;
- Image comparison python;
- Visual image comparison python;
- Detect differences between images python;
- Detect visual differences between images python.

Zoals te zien in de zoektermen, wordt specifiek gezocht op oplossingen in Python. Er wordt bewust hiervoor gekozen, omdat Python een toegankelijke taal is welke veel in de wereld van artificial intelligence (en daarmee computer vision) wordt gebruikt (Patel, 2022). Door gebruik te maken van deze zes bovenstaande zoektermen worden zoveel mogelijk relevante bronnen gevonden voor het vergelijken van afbeeldingen.

Bij iedere zoekterm worden de eerste tien resultaten meegenomen naar de volgende stap. Hiermee wordt het totaal aantal zoekresultaten beperkt tot zestig (inclusief mogelijk dubbele resultaten).

Selectiecriteria

Nadat de zoekresultaten zijn verzameld, worden allereerst dubbele resultaten eruit gefilterd. Vervolgens worden de resultaten beoordeeld op de volgende criteria:

- Resultaten zijn in het Engels;
- Resultaten zijn maximaal 5 jaar oud indien vermeld. Indien geen datum vermeld, wordt het resultaat meegenomen. Hiermee wordt voorkomen dat oude technieken die niet meer gebruikt worden, worden meegenomen in het onderzoek.

Inhoudelijke beoordeling

De resulterende resultaten zullen inhoudelijk worden beoordeeld op bruikbaarheid. Hiervoor worden de bronnen bestudeerd. Daar de bronnen zullen verschillen van vorm, kan hiervoor geen standaard aanpak worden gedefinieerd.

Bijlage C – Requirements

Tabel 1 Functionele requirements

Nummer	Requirement
F1	De applicatie geeft de juiste pagina('s) van de scan weer op basis van de pagina('s) van het origineel tijdens het controleren met menselijke handelingen, zolang er geen lege pagina's in het origineel zitten, die niet in de scan zitten en de volgorde van het origineel en de scan hetzelfde is.
F3	De applicatie geeft de juiste pagina('s) van de scan weer op basis van de pagina('s) van het origineel tijdens het controleren met menselijke handelingen, als er lege pagina's in het origineel zitten, die niet in de scan zitten.
F2	De gebruiker krijgt een onderscheidende geluidsmelding, als de applicatie een fout in de scan detecteert tijdens het controleren met menselijke handelingen.
F4	De applicatie geeft automatisch de juiste pagina('s) weer op basis van de pagina('s) van het origineel tijdens het controleren met menselijke handelingen, als de volgorde van de scan en het origineel van elkaar afwijken.
F14	De gebruiker krijgt een geluidsmelding tijdens het controleren met menselijke handelingen, als de applicatie een pagina gecontroleerd heeft.
F7	De gebruiker kan handmatig het aantal toelaatbare fouten in een batch instellen.
F9	De applicatie geeft tijdens het controleren met menselijke handelingen een signaal als de grens van het aantal toelaatbare fouten binnen een batch is bereikt.
F16	De applicatie controleert of de paginavolgorde van de scan overeenkomt met het origineel.
F15	De gebruiker kan een gedetecteerde fout markeren als zijnde geen fout.
F8	De gebruiker kan aan iedere soort fout een waardeoordeel toekennen.
F5	De applicatie geeft na het controleren van een batch een rapport met de resultaten van de controle.
F6	De rapportage bevat een lijst met gevonden fouten en waar deze zitten.
F10	De applicatie controleert of alle documenten van een dossier zijn gescand.
F11	De applicatie controleert of alle pagina's van een document zijn gescand.
F34	De applicatie controleert of de resolutie van de scan overeenkomt met de afgesproken resolutie, op basis van de metagegevens van de scan.
F17	De applicatie controleert of er geen hoeken van een pagina zijn omgevouwen.
F40	De applicatie controleert of er in de scan pagina's missen, welke onterecht zijn gezien als lege pagina's en derhalve uit de scan verwijderd.
F21	De applicatie controleert of er geen zwarte lijnen in de scan zitten, die niet in het origineel zitten.
F19	De applicatie controleert of er geen scheuren in de scan zitten, die niet in het origineel zitten.
F20	De applicatie controleert of er geen vlekken in de scan zitten, die niet in het origineel zitten.

F24	De applicatie controleert of het document in de juiste leesrichting is gescand.
F26	De applicatie controleert of de scan recht staat, met een maximaal toelaatbare afwijking van 2 graden.
F23	De applicatie controleert of er geen zwarte randen zijn, aan de rand van een pagina van de scan, welke niet in het origineel zitten.
F18	De applicatie controleert of er geen doordrukpagina's in de scan zitten.
F25	De gebruiker kan selecteren, op welke soort fouten gecontroleerd moet worden.
F32	De voortgang van het controleren van een batch kan worden opgeslagen en op een later moment worden vervolgd.
F31	De applicatie controleert of er in de scan nog lege pagina's zitten, die eruit gehaald hadden moeten worden.
F33	De rapportage bevat een afbeelding per gevonden fout.
F37	De applicatie controleert of de paginavolgorde klopt aan de hand van het paginanummer.
F36	De applicatie controleert of, indien van toepassing, de envelop is gescand.
F38	De applicatie controleert tijdens de controle zonder menselijke handelingen of pagina's mogelijk incompleet zijn op basis van context.
F39	De applicatie kan, indien deze optie is geselecteerd, controleren of enkel gewaarmerkte tekeningen in de scan zitten en geen kopieën.
F41	Het waardeoordeel voor fouten kan worden geïmporteerd vanaf een CSV-bestand.

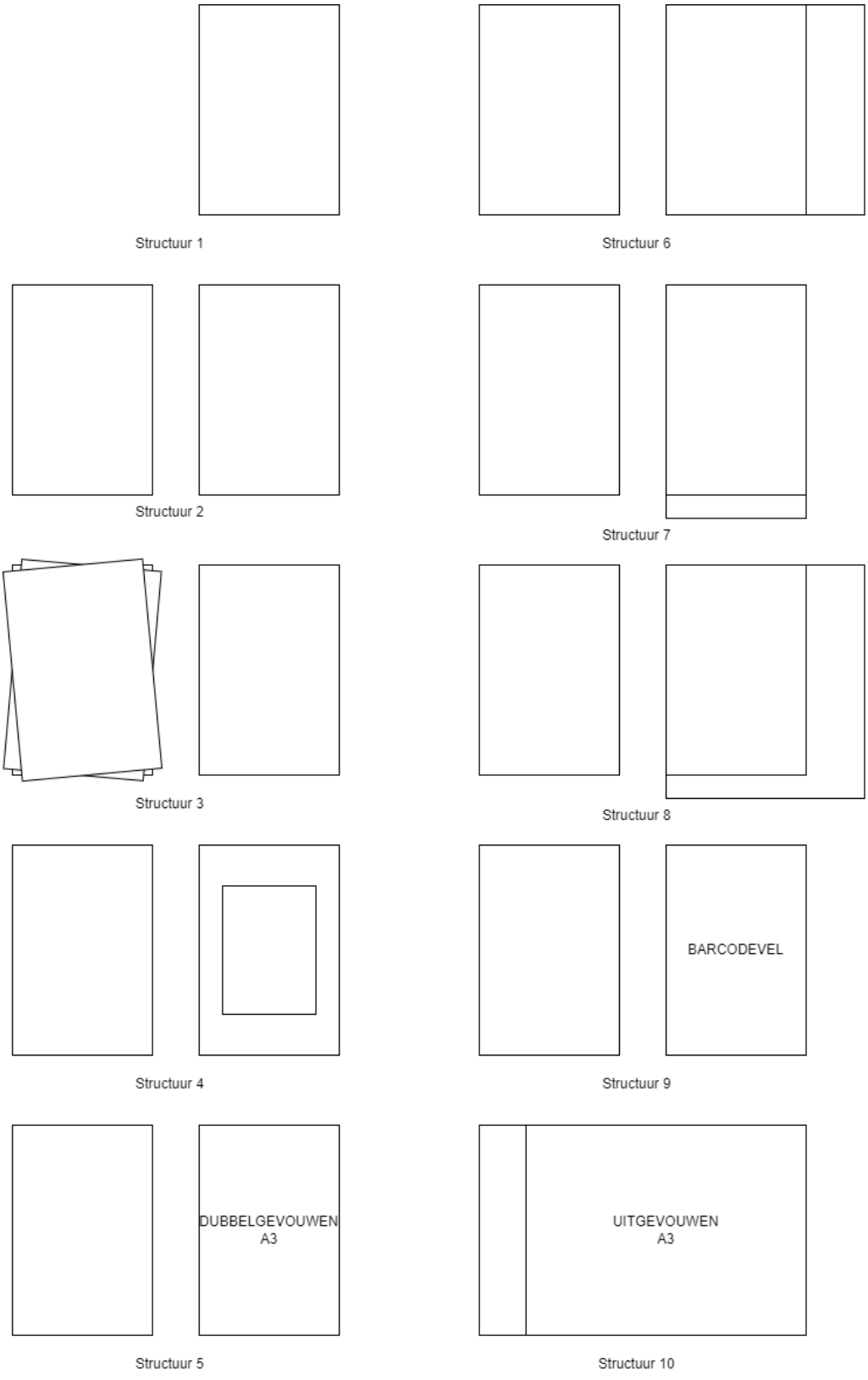
Noot: eisen F12, F13, F22, F27, F28, F29, F30 en F35 zijn vervallen tijdens het valideren en komen daarom niet voor in bovenstaande lijst.

Tabel 2 Niet-functionele requirements

Nummer	Requirement
N15	Alle controles die vooraf kunnen worden gedaan, zonder menselijke handelingen, worden vooraf gedaan.
N20	De applicatie kan de controles zonder menselijke handelingen uitvoeren voor papierformaten tot en met A0+.
N4	De applicatie kan losbladige dossiers controleren.
N5	De applicatie kan pagina's welke 90 graden zijn gedraaid in het fysieke dossier controleren.
N17	De applicatie kan barcodevellen verwerken.
N14	De applicatie kan omgaan met de verschillende structuren van een dossier.
N7	De applicatie heeft een foutmarge van maximaal 2,5% voor type II fouten (false negatives) ten opzichte van het totaal aantal pagina's wat afgekeurd had moeten worden.
N8	De applicatie verwerkt één kant van één pagina binnen maximaal 1 seconde, bij het verwerken met menselijke handelingen.
N16	De gebruiker hoeft gedurende de controle minimaal gebruik te maken van toetsenbord en/of muis, zolang er geen fouten worden gedetecteerd.
N9	De applicatie logt de gehele controle per batch.

N22	De gebruiker kan de logs exporteren naar een bestand.
N10	De logs bevatten de naam van de gebruiker (hoeft niet geauthentiseerd te zijn).
N11	De logs bevatten een log van iedere gecontroleerde pagina, inclusief of deze is goedgekeurd of afgekeurd (en waarom).
N12	Indien een gebruiker een gedetecteerde fout handmatig markeert als zijnde geen fout, wordt dit gelogd.
N18	De gebruiker moet zich authentiseren en autoriseren, voordat deze toegang krijgt tot de scans.
N1	De applicatie voldoet aan de AVG-wetgeving.
N2	De applicatie bewaart geen afbeeldingen, scans of andere representaties van gecontroleerde dossiers na afloop van een controle.
N3	De applicatie deelt geen data met externe partijen.
N13	De applicatie kan papierformaten tot en met A3-formaat controleren.
N21	De applicatie kan met de verschillende API-endpoints voor zowel de klant als voor intern gebruik communiceren voor het ophalen van de scans.
N6	De applicatie heeft een foutmarge van maximaal 4% voor type I fouten (false positives) ten opzichte van het totaal aantal pagina's wat goedgekeurd had moeten worden.
N19	De applicatie verwerkt één kant van één pagina binnen maximaal 2 seconden, bij het verwerken zonder menselijke handelingen.

Bijlage D – Structuren van dossiers



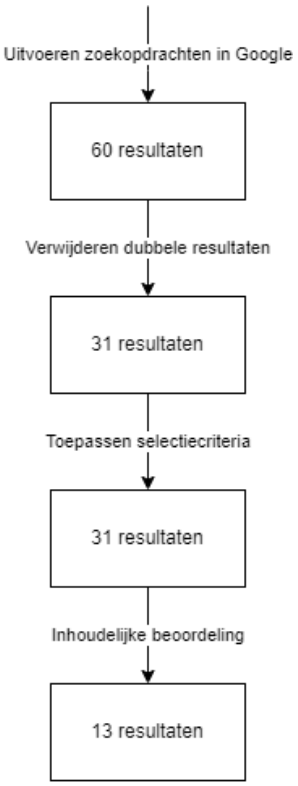
Figuur 3 Structuren

Tabel 3 Beschrijving structuren

Structuur	Toelichting
Structuur 1	Enkele stapel papier. Deze structuur is de startsituatie voorafgaand aan het controleren.
Structuur 2	Twee stapels papier. De linkerstapel ligt ondersteboven. Dit zijn de omgebladerde pagina's.
Structuur 3	De linkerstapel kan met het bladeren scheef gaan liggen.
Structuur 4	Niet alle vellen zijn A4 of dubbelgevouwen A3. Sommige vellen zijn kleiner in breedte, lengte of beiden.
Structuur 5	A3 ligt dubbelgevouwen in de stapel.
Structuur 6	Niet alle vellen zijn A4 of dubbelgevouwen A3. Sommige vellen zijn breder en steken uit.
Structuur 7	Niet alle vellen zijn A4 of dubbelgevouwen A3. Sommige vellen zijn langer en steken uit.
Structuur 8	Niet alle vellen zijn A4 of dubbelgevouwen A3. Sommige vellen zijn breder en langer en steken uit aan twee kanten.
Structuur 9	Het fysieke dossier bevat barcodevellen, welke het begin van een document aangeven, tabbladen aangeven, enzovoorts.
Structuur 10	A3 wordt tijdens de controle uitgevouwen en vervolgens weer dubbelgevouwen.

Iedere pagina kan inhoud bevatten op zowel de voor- als achterkant. Pagina's kunnen ook leeg zijn. Afhankelijk van de situatie zit een lege pagina wel of niet in de scan. Lege pagina's met bijvoorbeeld enkel een paginanummer zitten wel in de scan, lege pagina's met enkel perforatiegaatjes kunnen uit de scan zijn gehaald. Combinaties van structuren zijn mogelijk.

Bijlage E – Totstandkoming bronnen rapid review



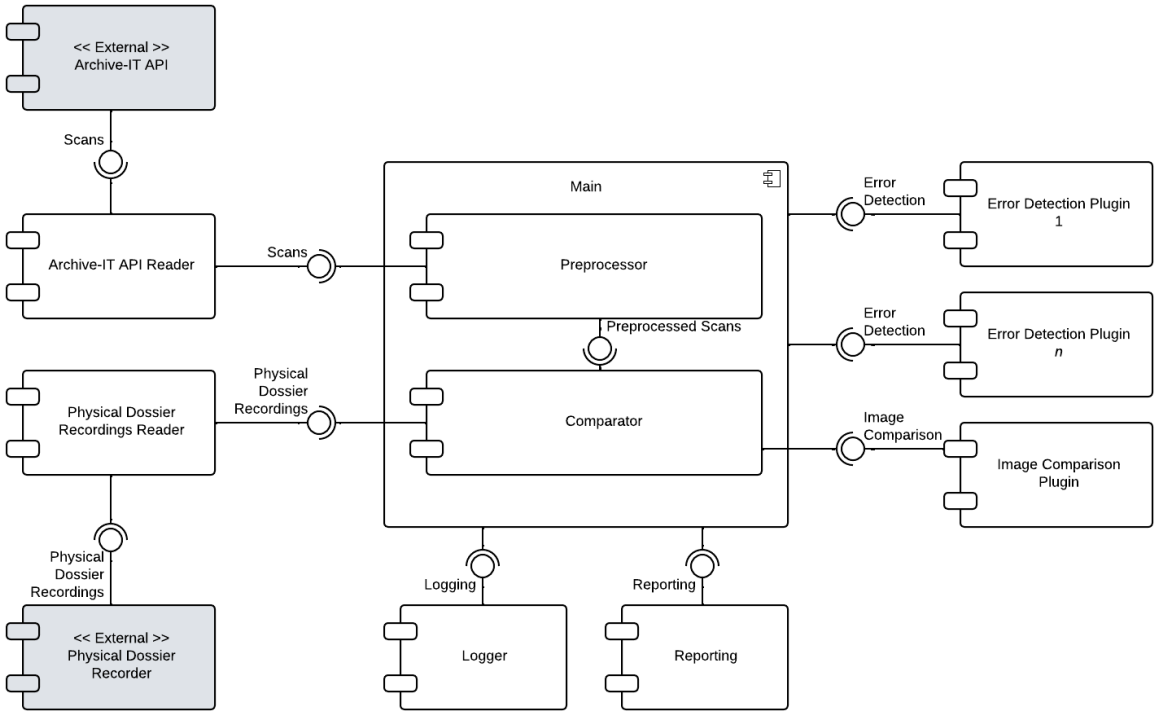
Figuur 4 Totstandkoming geschikte bronnen

De bronnen die geschikt zijn voor het onderzoek zijn:

- Measuring similarity in two images using Python (Raval, 2021)
- image-similarity-measures (Müller, Ekhtiari, Almeida, & Rieke, 2020)
- How To Measure Image Similarities In Python (Yin, 2021)
- A Beginners' Guide to Image Similarity using Python (Agrawal, 2021)
- Measure similarity between images using Python-OpenCV (Sonkar, 2023)
- Image Similarity Measures (Müeller, et al., 2023)
- Comparing ground truth with predictions using image similarity measures (Ekhtiari, 2021)
- 191 - Measuring image similarity in python (DigitalSreeni, 2021)
- How-To: Python Compare Two Images (Rosebrock, 2021a)
- How to compare two images in OpenCV Python (Khan, 2022)
- Detect and visualize differences between two images with OpenCV Python op Stack Overflow (PlsWork & Lam, 2022)
- Image Difference with OpenCV and Python (Rosebrock, 2021b)
- differences-between-two-images (or multiple images) (Benard, 2022)

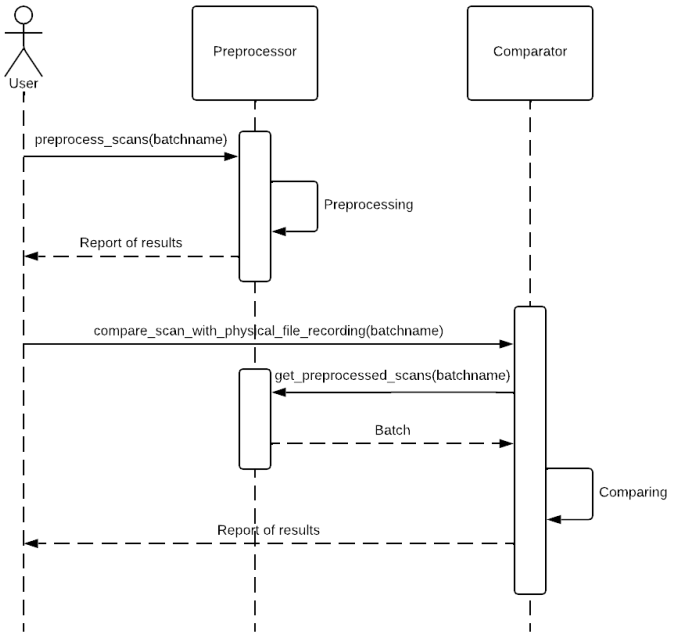
Bijlage F – Ontwerpen op architectuurniveau

Componenten



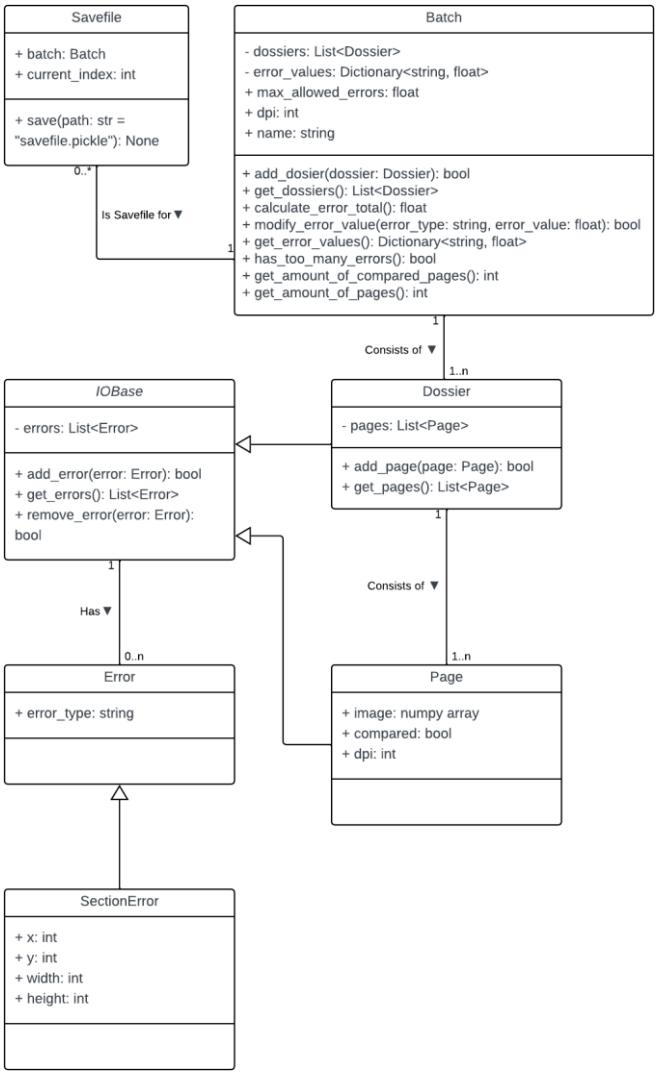
Figuur 5 Componentendiagram op architectuurniveau

Programmaflow



Figuur 6 Main program flow

Datastructuur



Figuur 7 Klassendiagram op architectuurniveau

Bijlage G – Vergelijkingstechnieken

Gevonden vergelijkingstechnieken zijn:

- Structural similarity index (SSIM) (Rosebrock, 2021a)
- Multi-scale structural similarity index (MS-SSIM) (Wang, Simoncelli, & Bovik, 2003)
- Feature similarity measure (FSIM) (Zhang, Zhang, Mou, & Zhang, 2011)
- Information theoretic-based statistic similarity measure (ISSM) (Ekhtiari, 2021)
- Spatial correlation coefficient (SCC) (Zhou, Civco, & Silander, 1998)
- Matchen van keypoints middels een brute force matcher (DigitalSreeni, 2021)
- Relative average spectral error (RASE), werkt enkel op basis van kleurverdeling (Gonzalez-Audicana, Saleta, Catalan, & Garcia, 2004)
- Spectral angle mapper (SAM), werkt enkel op basis van kleurverdeling (Ekhtiari, 2021)

Bijlage H – Accuraatheid per vergelijkingstechniek

Deze test zijn berekend aan de hand van de FinnAlberts/Image-comparison-metrics repository (Alberts, 2023a). Hierbij zijn van een document drie varianten, welke met drie verschillende scanners zijn vastgelegd. Variant 0 is het basisdocument, bij variant 1 mist één zin en bij variant 2 is een zin inhoudelijk veranderd, maar is de lengte hetzelfde gebleven. De drie gebruikte scanners zijn een Codax Scanner, een Inotec Scanner en een CZUR boekenscanner. De opnames van de CZUR boekenscanner kunnen worden vergeleken met de opnames van een 18MP camera.

Bij de vergelijking wordt van één scanner variant 0 genomen, welke vervolgens wordt vergeleken met de drie varianten van een andere scanner. Tevens wordt variant 0 van de scanner met zichzelf vergeleken, om zo duidelijk te maken wat de perfecte waarde is. Volgorde is, zoals te zien in de resultaten, hierbij belangrijk.

Tests zijn uitgevoerd met een Intel Core I5-1145G7, 16.0 GB RAM en geen dedicated GPU.

Tabel 4 Accuraatheid met CZUR boekenscanner als origineel en Codax Scanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	1,00000; 1,00000	0,76673; 0,78129	0,72578; 0,74182	0,74319; 0,75725
MS-SSIM	1,00000	0,80175	0,70400	0,76999
FSIM	1,00000	0,37028	0,31478	0,33983
ISSM	0,00000	0,00000	0,00000	0,00000
SCC	1,00000	0,00086	0,00073	-0,00083
RASE	NaN	inf	inf	inf
SAM	1,49012E-08	0,21838	0,24697	0,23432
Keypoints	1,00000	0,62632	0,53180	0,57895

Tabel 5 Accuraatheid met CZUR boekenscanner als origineel en Inotec Scanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	1,00000; 1,00000	0,77243; 0,78921	0,73698; 0,75525	0,758820; 0,77511
MS-SSIM	1,00000	0,82058	0,72600	0,79387
FSIM	1,00000	0,36271	0,30635	0,32994
ISSM	0,00000	0,00000	0,00000	0,00000
SCC	0,99997	-0,00179	0,00010	0,00256
RASE	NaN	inf	inf	inf
SAM	1,49012E-08	0,21322	0,23904	0,22515

Keypoints	1,00000	0,67925	0,62733	0,67251

Tabel 6 Accuraatheid met Codax als origineel en CZUR Boekenscanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	1,00000; 1,00000	0,78046; 0,79539	0,75668; 0,77096	0,77746; 0,79119
MS-SSIM	1,00000	0,79951	0,73334	0,78974
FSIM	1,00000	0,37299	0,33842	0,37276
ISSM	0,00000	0,00000	0,00000	0,00000
SCC	0,95232	0,00037	0,00098	-0,00213
RASE	0,00000	1621,2	1719,6	1591,2
SAM	0,00000	0,21542	0,23196	0,21224
Keypoints	1,00000	0,59896	0,53030	0,53608

Tabel 7 Accuraatheid met Inotec Scanner als origineel en CZUR boekenscanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	1,00000; 1,00000	0,78803; 0,80515	0,77929; 0,79545	0,77817; 0,79345
MS-SSIM	1,00000	0,81651	0,75827	0,78318
FSIM	1,00000	0,36337	0,33929	0,34195
ISSM	0,00000	0,00000	0,00000	0,00000
SCC	0,25427	-0,00142	0,00553	-4,43894E-05
RASE	0,00000	1515,3	1545,7	1516,5
SAM	0,00000	0,20959	0,21718	0,21303
Keypoints	1,00000	0,64497	0,57471	0,67647

Bijlage I – Performance per vergelijkingstechniek

Deze test zijn berekend aan de hand van de FinnAlberts/Image-comparison-metrics repository (Alberts, 2023a). Hierbij zijn van een document drie varianten, welke met drie verschillende scanners zijn vastgelegd. Variant 0 is het basisdocument, bij variant 1 mist één zin en bij variant 2 is een zin inhoudelijk veranderd, maar is de lengte hetzelfde gebleven. De drie gebruikte scanners zijn een Codax Scanner, een Inotec Scanner en een CZUR boekenscanner. De opnames van de CZUR boekenscanner kunnen worden vergeleken met de opnames van een 18MP camera.

Bij de vergelijking wordt van één scanner variant 0 genomen, welke vervolgens wordt vergeleken met de drie varianten van een andere scanner. Tevens wordt variant 0 van de scanner met zichzelf vergeleken, om zo duidelijk te maken wat de perfecte waarde is. Volgorde is, zoals te zien in de resultaten, hierbij belangrijk.

Tests zijn uitgevoerd met een Intel Core I5-1145G7, 16.0 GB RAM en geen dedicated GPU.

Tabel 8 Performance in seconden met CZUR boekenscanner als origineel en Codax Scanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	6.75092	6.77002	6.76831	6.77959
MS-SSIM	9.51606	9.35471	9.15451	9.21762
FSIM	238.36546	217.02548	219.08128	214.27245
ISSM	1.78487	1.49566	1.48181	1.52674
SCC	4.45053	4.34356	4.26210	4.31293
RASE	0.71514	0.60565	0.61915	0.61989
SAM	0.12348	0.09321	0.08006	0.08114
Keypoints	0.19856	0.20929	0.15564	0.15155

Tabel 9 Performance in seconden met CZUR boekenscanner als origineel en Inotec Scanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	7.08577	6.83373	6.94232	6.06597
MS-SSIM	9.55659	9.39592	10.20180	8.33689
FSIM	270.97050	261.16947	247.88772	229.83737
ISSM	1.89379	1.80873	1.84957	1.56725
SCC	4.42795	4.21818	4.66462	3.96736
RASE	0.72297	0.64228	0.70272	0.61876
SAM	0.06393	0.06385	0.07375	0.07462

Keypoints	0.35182	0.17583	0.14747	0.17342

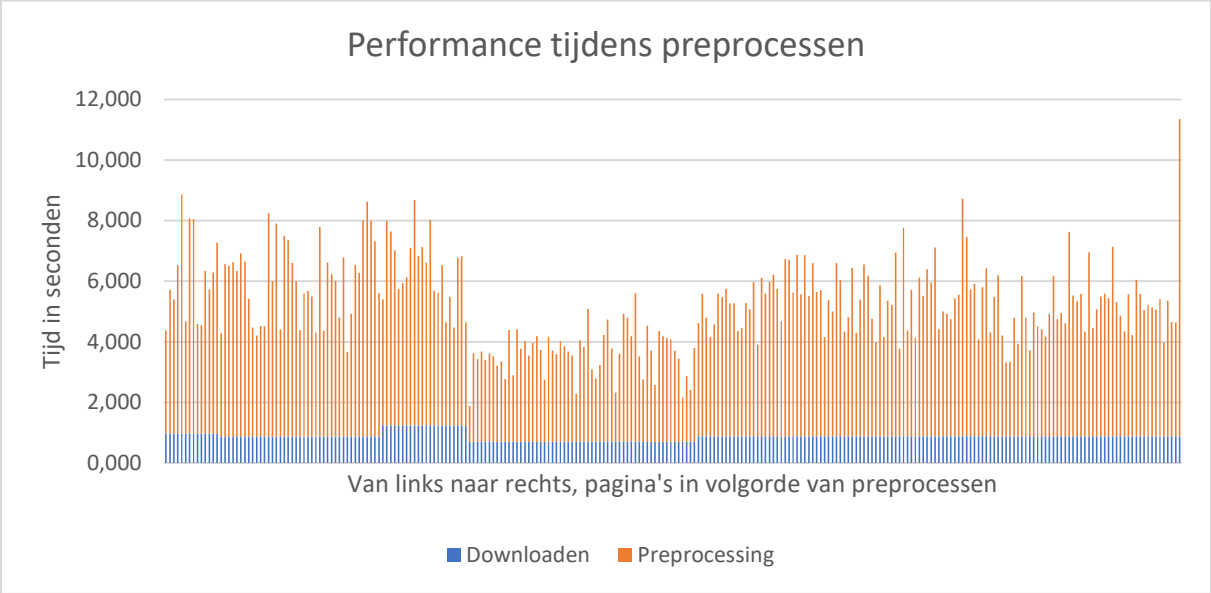
Tabel 10 Performance in seconden met Codax als origineel en CZUR Boekenscanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	11.77348	8.64986	9.08664	9.29123
MS-SSIM	12.80262	12.54456	12.66341	12.70721
FSIM	290.16471	307.99439	279.43203	278.11502
ISSM	2.80461	2.55978	2.57378	2.66859
SCC	5.89357	5.78402	6.03866	5.85264
RASE	1.02401	0.92781	0.93815	1.04292
SAM	0.08940	0.11988	0.10750	0.11615
Keypoints	0.36411	0.22005	0.23222	0.22168

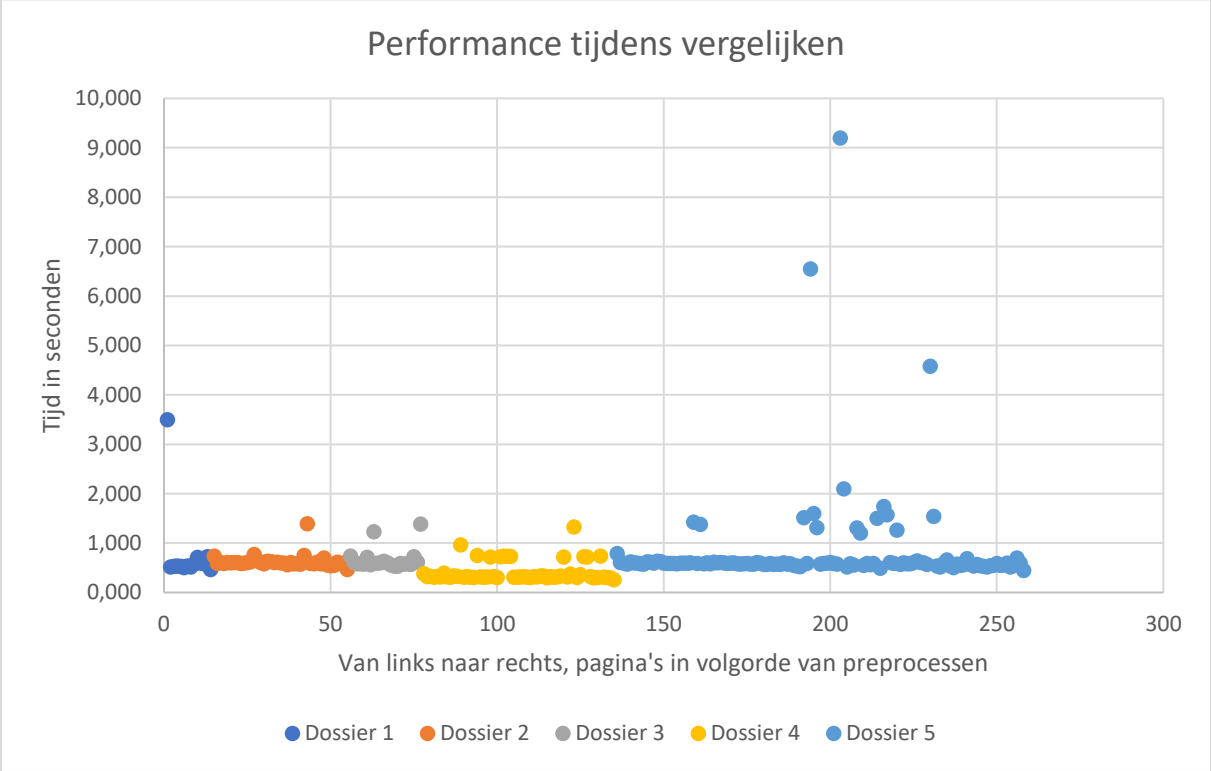
Tabel 11 Performance in seconden met Inotec Scanner als origineel en CZUR boekenscanner ter vergelijking

Metriek	Origineel	Overeenkomstige afbeelding (variant 0)	Variant 1	Variant 2
SSIM	8.96861	8.91187	8.96561	8.72419
MS-SSIM	12.25682	12.41329	12.29273	12.39606
FSIM	280.38374	285.56429	282.58893	281.21722
ISSM	2.74128	2.57793	2.61904	2.67375
SCC	5.83486	5.72651	5.67720	5.63670
RASE	0.95494	0.95235	0.98182	0.94388
SAM	0.10274	0.10863	0.12179	0.12159
Keypoints	0.29301	0.21989	0.22923	0.21540

Bijlage J – Performancetesten



Figuur 8 Tijd per pagina tijdens preprocessen



Figuur 9 Tijd per pagina tijdens vergelijking

Bijlage K – Accuraatheidstesten

Tabel 12 Confusion matrix voor preprocessen

Positive – een fout wordt gedetecteerd Negative – geen fout wordt gedetecteerd		Handmatige controle	
		Positive	Negative
Automatische controle	Positive	4	12
	Negative	0	242

Tabel 13 Confusion matrix voor vergelijken

Positive – een fout wordt gedetecteerd Negative – geen fout wordt gedetecteerd		Handmatige controle	
		Positive	Negative
Automatische controle	Positive	4	28
	Negative	0	224