

HOE WORDT EUS DE BESTE KEUS?

OPTIMALISEREN VAN ENDO-ULTRASONOGRAFISCHE BEOORDELING TER
ONDERSCHEIDING VAN T1 EN T2+ RECTALE CARCINOMEN MET BEHULP
VAN DEEP LEARNING

WETENSCHAPPELIJK VERSLAG
TECHNISCHE GENEESKUNDE ~ TGO

Groep 10
21 Juni 2021

Studenten:

Stan Benjamins
Jesse van der Duin
Heleen van Nie
Luca Plekkenpol

s2098253
s2109433
s2101769
s2108119

Begeleiders:

Dr. F. ter Borg
Dr. J.M. Wolterink
MSc. E.A. van de Graaf
MSc. A.G. Lovink
S. van der Heide

Klinisch
Technologisch
Proces
Proces
Tutor

INHOUDSOPGAVE

Voorwoord	4
Introductie	5
Analyse van rectale carcinomen	6
Anatomie en fysiologie	6
Pathofysiologie	8
Endoscopische ultrasonografie	9
Deep learning	10
Convolutional Neural Network	10
Componenten Convolutional Neural Network	11
Trainen van het model	13
Onderzoeks- en deelvragen	14
Onderzoeksvraag	14
Deelvragen	14
Methode	14
Literatuuronderzoek	15
Classificatie	15
Segmentatie	15
Classificatie architecturen	16
ResNet	16
VGGNet	16
Inception	17
Segmentatie architecturen	17
Fully Convolutional Networks	17
U-Net	18
SegNet	18
Benodigde hoeveelheid data	18
Trainen van kleine datasets	20
Kwaliteit van data	21
Conclusie	21
Interbeoordelaarsvariabiliteit	23
Methode	23
Evaluatie	23
Dice Similarity Coefficient	23
Hausdorff Distance	24
Kappa	24
Resultaten	25

Conclusie	26
Modelvariabiliteit.....	28
Methode.....	28
Evaluatie.....	28
Resultaten	28
Conclusie.....	29
Discussie.....	31
Literatuuronderzoek.....	31
Onderzoek Interbeoordelaarsvariabiliteit.....	32
Onderzoek Modelvariabiliteit.....	33
Conclusie	35
Referenties.....	36
Bijlagen	42
Bijlage A: Convolutie.....	42
Bijlage B: Architecturen Toepassing Classificatie.....	44
Bijlage C: Architecturen Toepassing Segmentatie	46
Bijlage D: Berekening DSC en AHD in Python.....	47
Bijlage E: Verschil Segmentatie TG-studenten en Expert	54
Bijlage F: Boxplots Interbeoordelaarsvariabiliteit.....	57
Bijlage G: Model Python Script	59
Bijlage H: Verschil Model TG-studenten en Expert	66
Bijlage I: Boxplots Modelvariabiliteit	68
Referenties Bijlage.....	70

VOORWOORD

Voor u ligt het wetenschappelijke verslag van het onderzoek naar de mogelijkheden voor verbetering van de beoordeling van endo-ultrasonografische beelden van rectumcarcinomen, ter onderscheiding van de T1 en T2+ stadia. Hierbij willen wij graag een woord van dank uitspreken naar de volgende mensen.

Allereerst willen wij onze dank betuigen aan dr. Frank ter Borg, MDL-arts in Deventer Ziekenhuis, onder andere voor de enthousiaste begeleiding en de introductie van het klinische probleem. Daarnaast heeft dr. Frank ter Borg ons geleerd, door middel van een snelcursus, om rectumcarcinomen in endo-ultrasonografische beelden te segmenteren. Daarbij heeft hij de omvang en de complexiteit van het probleem duidelijk gemaakt. Ook heeft hij zijn tijd beschikbaar gesteld om gesegmenteerde EUS-beelden voor dit onderzoek te leveren. Verder is aan ons als studenten de mogelijkheid gegeven om in het Deventer Ziekenhuis inzicht te krijgen in de klinische context van ons onderzoek.

Ook willen wij dr. Jelmer Wolterink bedanken voor de betrokken begeleiding tijdens dit onderzoek. Zo heeft hij ons geholpen met de technologische aspecten van dit onderzoek, zoals het programmeren in Python, door mee te denken bij technische problemen en suggesties te geven om deze op te lossen. Meetings waarin enkele technische problemen werden opgelost met hulp van dr. Jelmer Wolterink hebben bijgedragen aan ons begrip van deep learning, de toepassingen in de medische wetenschap en het omgaan met Python scripts.

Daarnaast spreken wij onze dank uit naar Stein van der Heide, die als tutor de procesbegeleiding van ons onderzoek voor zijn rekening heeft genomen. Hierbij werden we geholpen met problemen waar we in ons onderzoek tegenaan liepen en heeft hij bijgedragen aan de professionele ontwikkeling van ons als Technisch Geneeskunde studenten.

Verder willen wij Thom Nijboer bedanken voor het beschikbaar stellen van zijn Python script voor automatische beeldanalyse in ons onderzoek en zijn begeleiding bij het begrijpen van en het leren werken met dit model. Daarnaast spreken wij onze dank uit naar Martine Schulten, die heeft bijgedragen aan ons begrip van het deep learning model en ons heeft geholpen met de praktische zaken rondom het segmenteren.

Ten slotte bedanken wij het Deventer Ziekenhuis voor het beschikbaar stellen van de EUS-beelden waarmee in dit onderzoek gewerkt is en voor het mogelijk maken van het meekijken in de kliniek op de MDL-afdeling bij endoscopieën, EUS-onderzoeken, ESD-procedures enzovoorts.

Wij wensen u veel plezier met het lezen van dit onderzoeksverslag.

Stan Benjamins, Jesse van der Duin, Heleen van Nie en Luca Plekkenpol
Enschede, 21 juni 2021

INTRODUCTIE

Rectale carcinomen (RC's) staan op de achtste plek van de meest incidentie vormen van kanker en worden getypeerd als de tiende dodelijkste vorm van kanker.[1] Echter, een vroege diagnose kan de overlevingskansen significant vergroten.[2] Daarnaast is een juiste stadiëring, met behulp van de TNM-classificatie [3], van belang voor het opstellen van de geschikte Therapie.[4] De TNM-classificatie stadiëert RC's volgens een T-, N-, en M-stadium. Allereerst geeft het T-stadium informatie over de infiltratie van de tumor in de darmwand. Hierbij wordt er onderscheid gemaakt tussen de verschillende lagen van de darmwand, die elk een ander T-stadium typeren. Daarnaast geeft het N-stadium informatie over de metastaseringen naar lokale lymfeknopen waarbij onderscheid wordt gemaakt tussen de hoeveelheid lokale lymfeknopen waarin metastase is geconstateerd. Als laatste geeft het M-stadium informatie over metastaseringen naar andere gebieden in het lichaam, zoals de lever of longen. Door te classificeren op basis van het T-, N- en M-stadium, kan het algehele stadium van het RC bepaald worden.[5]

Op dit moment draagt een bevolkingsonderzoek bij aan vroegtijdige diagnose van RC's.[6] Bij het bevolkingsonderzoek wordt de ontlasting van personen van 55 tot 75 jaar onderzocht op hemoglobine. Als hemoglobine aanwezig is in de ontlasting kan dit duiden op een tumor in het colon of het rectum en wordt de persoon uitgenodigd voor een vervolgonderzoek, zoals een coloscopie.[7] De overlevingskansen nemen hierdoor aanzienlijk toe, doordat carcinomen in het rectum zo in een vroeger stadium worden ontdekt dan wanneer de patiënt geen bevolkingsonderzoek zou ondergaan. Zo is de vijfjaarsoverleving voor stadium I carcinomen 94% [5], maar voor stadium IV carcinomen slechts 8%.[6]

Na het bevolkingsonderzoek worden er verschillende stappen ondernomen voor de detectie en stadiëring van de RC's. In de praktijk wordt naast een coloscopie gebruik gemaakt van CT-colografie, X-colon en/of MR colografie voor de detectie van RC's.[8] Met de huidige technieken is juiste stadiëring voor een vroege diagnose van RC's ondermaats. Voor de huidige stadiëring van RC's wordt onder andere gebruikt gemaakt van MRI. Echter, met MRI kan geen betrouwbaar onderscheid worden gemaakt tussen de afzonderlijke lagen van de darmwand, waardoor de determinatie van het T-stadium slechts voor 77% accuraat is.[4] Endoscopische ultrasonografie (EUS) slaagt er wel in om de verschillende lagen in beeld te brengen en verhoogt hiermee de accuraatheid van de determinatie van het T-stadium tot 87%. Het nadeel van deze techniek is dat de beoordeling van deze beelden onderhevig is aan grote interbeoordelaarsvariabiliteit.[4] Een consequentie hiervan is dat de patiënt mogelijk foutief gediagnosticeerd wordt, met alle gevolgen van dien.

Er is een groot verschil tussen de behandeling van een T1 of T2+ gestadieerd RC. Bij een T1 stadium kan met een endoscopische submucosale dissectie (ESD) de oppervlakkige tumor worden weggehaald. Bij een T2+ stadium is een resectie van het betreffende deel van het rectum nodig.[4,8] Een resectie is een zeer invasieve ingreep waarbij de kwaliteit van leven van patiënten aanzienlijk vermindert door veranderingen zoals een stoma.

Kortom, door foutieve stadiëring van T1 en T2+ RC's moet er een extra behandeling ingezet worden of is er sprake van overbehandeling. Om dit te voorkomen moet er een duidelijker onderscheid tussen T1 en T2+ gestadieerde RC's gemaakt kunnen worden. Een veelbelovend hulpmiddel hiervoor is het automatische analyseren van de EUS-beelden met behulp van *deep learning*. In dit onderzoek wordt geëvalueerd hoe een dergelijk model geoptimaliseerd kan worden en in hoeverre Technisch Geneeskunde (TG) studenten een bijdrage kunnen leveren aan de training van het model.

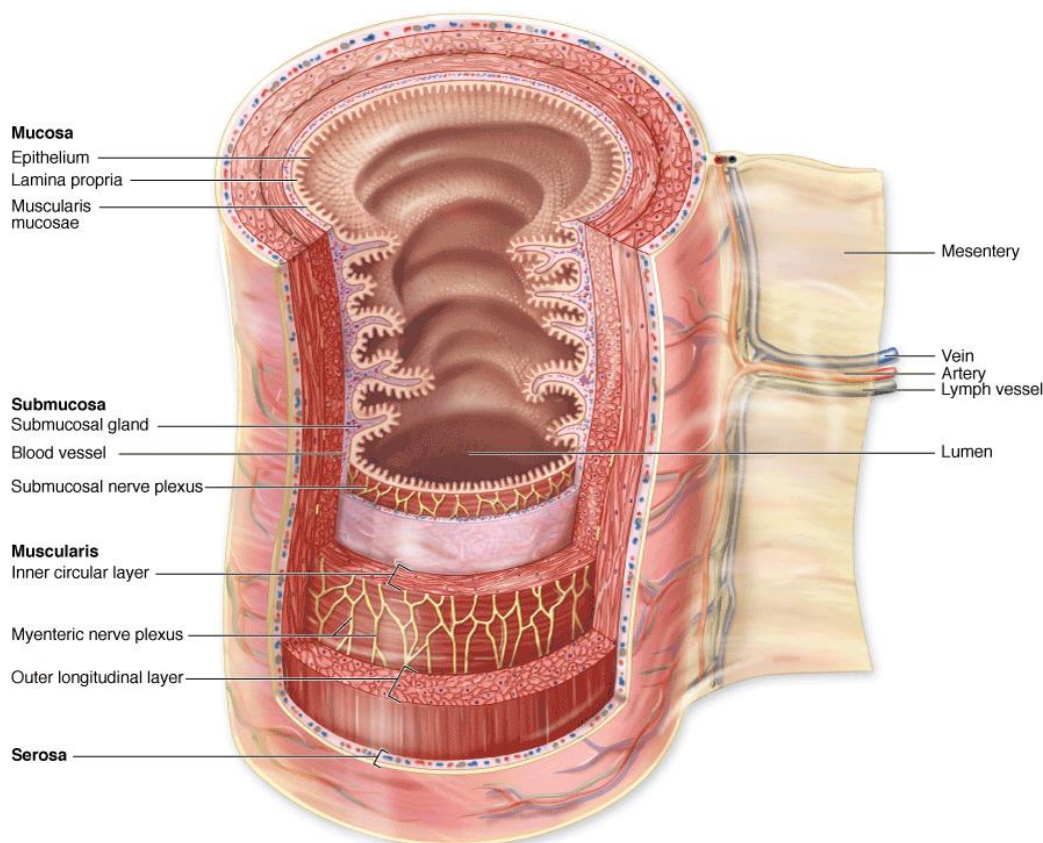
ANALYSE VAN RECTALE CARCINOMEN

Vóór de start van het onderzoek is er een uitvoerige analyse gedaan van de belangrijke onderwerpen die betrekking hebben op de stadiëring van RC's en automatische beeldanalyse. In deze sectie wordt dit besproken op het gebied van *Anatomie en Fysiologie*, *Pathofysiologie*, *Endoscopische ultrasonografie*, *Deep learning* en *Convolutional neural networks*.

ANATOMIE EN FYSIOLOGIE

De dikke darm, of ook wel het intestinum crassum, voert vitale functies uit, zoals het omzetten van onverteerbare etensresten in faeces en het opnemen van essentiële vitaminen en water uit de faeces. Verschillende klieren dragen bij aan de smering van de chymus door het afscheiden van mucus, of leveren een aandeel in de vertering door uitscheiding van spijsverteringsenzymen. Daarnaast beschermt het lymfoïde apparaat in de gastro-intestinale tractus tegen het binnendringen van infectieuze organismen en antigenen.[9]

De gastro-intestinale tractus bezit een algemeen bouwplan, gevisualiseerd in figuur 1, dat bestaat uit vier lagen: mucosa, submucosa, muscularis propria en serosa. Dit geldt dus ook voor het intestinum crassum, dat bestaat uit het caecum, het colon en het rectum.[10]

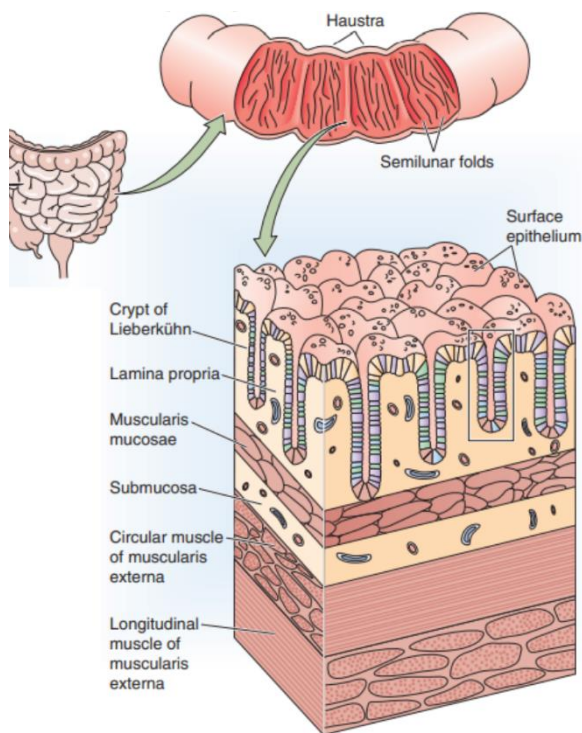


Source: Mescher AL; Junqueira's Basic Histology: Text and Atlas, 12th Edition: <http://www.accessmedicine.com>
Copyright © The McGraw-Hill Companies, Inc. All rights reserved.

Figuur 1: Opbouw van de wand van de gastro-intestinale tractus.[10]

De mucosa laag van het colon wordt tot de anus bekleed met cilindrisch epitheelweefsel en is georganiseerd in dicht opeengepakte crypten van Lieberkühn (figuur 2). In het anale kanaal gaat het epitheel over in meerlagig plaveiselepitheel van de epidermis.[10] Daarnaast bevat de mucosa twee verschillende lagen. Enerzijds een lamina propria van losmazig bindweefsel met daarin bloed- en lymfevaten en soms gladde spiervezels,

klieren en lymfefollikels. Anderzijds een lamina muscularis mucosae, wat een dun laagje glad spierweefsel is dat op de grens tussen de mucosa en de submucosa ligt.[10] De muscularis mucosae geeft de mucosa een eigen beweeglijkheid, waardoor het contact tussen het darmepitheel en de darminhoud bevorderd wordt. De mucosa in het colon is glad en bevat geen villi in tegenstelling tot het intestinum tenue, de dunne darm, maar wel grote hoeveelheden simpele tubulaire mucosa klieren die veel slijmbekercellen bevatten.[10] Mucoseuze klieren scheiden mucus uit in het lumen van het colon om het oppervlak glad te houden en om het te beschermen tegen grove delen chymus.[9] Rondom de mucosa ligt een laag met bloed- en lymfevaten en bindweefsel, genaamd de submucosa. Deze bevat een zenuwplexus, onderverdeeld in een binnenste (Meissner) en een buitenste (Henle/Schabadasch) plexus submucosus, zie figuur 2.[10] Deze laag dient als ondersteuning voor de andere lagen.[11]



Figuur 2: Schematische weergave van opbouw darmwand met crypten van Lieberkühn en haustra structuren.[11]

De muscularis propria omringt de mucosa en submucosa en bestaat uit een dikke circulaire laag van gladde spieren met daaromheen een dunnere longitudinale laag. Tussen de beide spierlagen ligt een tweede zenuwplexus, de plexus myentericus of ook wel de plexus van Auerbach, welke de contracties van beide spierlagen coördineert. De muscularis kan contraheren en zorgt voor peristaltiek in het intestinum crassum.[10] Continue contractie van het gladde spierweefsel in de muscularis propria zorgen voor de haustra structuren in het colon (figuur 2).[11]

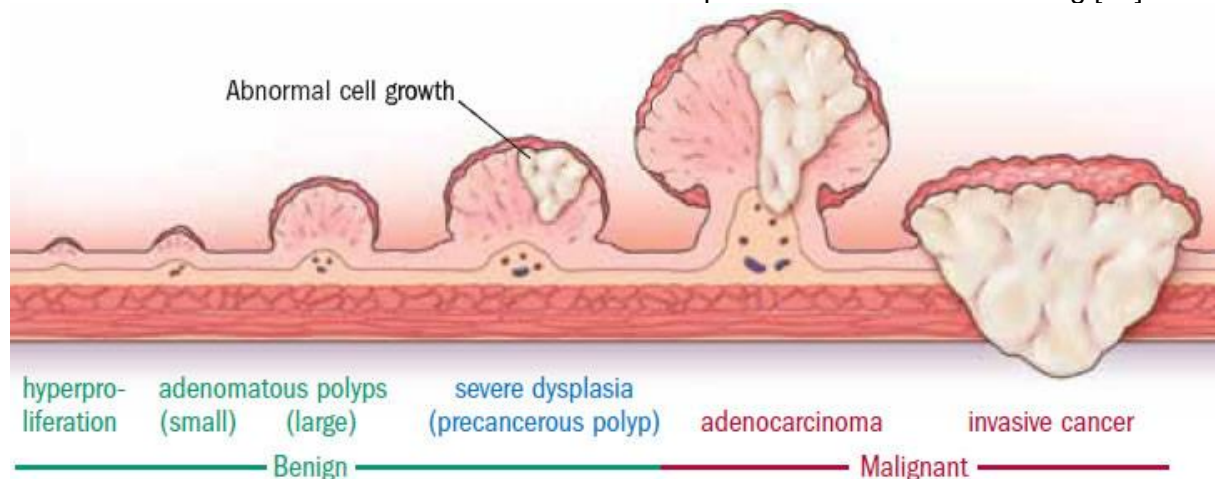
Ten slotte vormt de serosa de buitenste laag van het intestinum crassum. Dit is een dunne laag bestaande uit bindweefsel met veel bloed- en lymfevaten, afgedekt met een laag mesothelium.[10] Het scheidt waterige sereuze vloeistof uit om het buitenoppervlak van het colon te smeren. Hierdoor wordt het beschermd tegen frictie met andere buikorganen en omliggende spieren en botten.[11]

PATHOFYSIOLOGIE

In dit onderzoek is er sprake van een patiëntenpopulatie gediagnosticeerd met een rectaal carcinoom. Vaak worden tumoren in het colorectale gebied genoemd onder één noemer, wegens de biologische overeenkomsten tussen de twee soorten weefsels. Echter, er bestaat een verschil in de behandeling van de twee typen tumoren.[12]

Een poliep is een voorloper van een RC en is te definiëren als een abnormale uitgroeiing van weefsel in het lumen van de darm. Een indeling van verschillende poliepen kan worden gemaakt op het gebied van aanhechting aan de darmwand (sessiel of gesteeld), de histologie (hyperplastisch of adenomateus) en de potentie tot het vormen van neoplastisch weefsel (benigne of maligne).[12] De meeste RC's, zo'n 70-90%, ontstaan uit adenomateuze poliepen waardoor enkel wordt ingegaan op de kenmerken van dit type poliep.[13]

Een adenomateuze poliep is te definiëren als een poliep die bestaat uit dysplastische epitheelcellen. Deze poliepen variëren erg in vorm en grootte en kunnen onderverdeeld worden volgens hun groeipatroon: tubulair (buisvormig), villus (vlokvormig) of tubulovillus (een combinatie van de twee).[14] Doorgaans zijn kleine adenomen (< 2cm) tubulair en grotere adenomen tubulovillus of villus.[14] Adenomateuze poliepen ontstaan door enkele neoplastische veranderingen in de cellen in de crypten. De cellen verliezen namelijk de mogelijkheid tot apoptose en slagen er niet in om te differentiëren tot volwassen epitheelcellen gedurende de migratie naar het oppervlakkige deel van de crypt. Daarnaast zal de celdeling gedurende de migratie van de cel langs de hele lengte van de crypt persisteren.[12] In een normale situatie stopt dit al na een derde van de lengte van de crypten zullen volwassen cellen aan het einde worden afgestoten naar het darmlumen. Echter, door de verstoringen zullen er meer cellen worden aangemaakt dan dat er cellen worden afgestoten, waardoor er een uitgroeiing ontstaat aan het oppervlak richting het darmlumen (figuur 3). In 85% van de gevallen zullen er nog 8-10 mutaties plaats moeten vinden voordat het adenoom zich ontwikkelt tot een invasief carcinoom met potentie tot metastasering.[12]



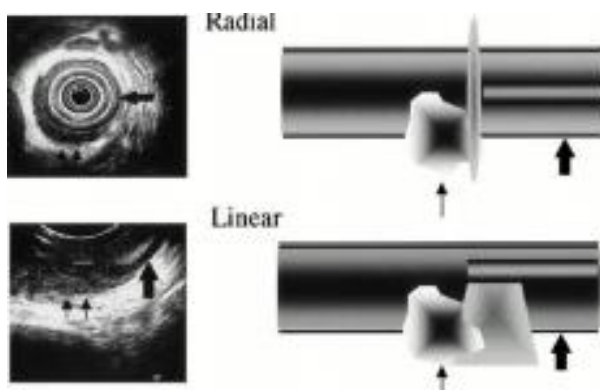
Figuur 3: Schematische weergave van groei van een poliep tot een carcinoom.[15]

In vroege stadia zullen de (kleine) RC's zorgen voor geen tot weinig klachten. Naarmate de tumor groeit is een veelvoorkomend symptoom bloed in de ontlasting, wat ook wel fecaal occult bloed wordt genoemd. Hierop wordt getest tijdens het bevolkingsonderzoek. Verdere symptomen kunnen variëren door verschil in grootte en locatie van het RC.[16]

ENDOSCOPISCHE ULTRASONOGRAFIE

Eén van de technieken waarmee RC's zichtbaar gemaakt kunnen worden is endoscopische ultrasonografie.[17] EUS is een techniek die bestaat uit een combinatie van echografie en endoscopie.[18] Met behulp van een transducer op de tip van de endoscoop kan een real-time afbeelding gemaakt worden.[17] Op basis van de piëzo-elektrische materialen waaruit de transducers bestaan, worden geluidsgolven uitgezonden. Deze piëzo-elektrische elementen zorgen namelijk voor omzetting van elektrische signalen naar longitudinale geluidsgolven en andersom.[18] De uitgezonden longitudinale geluidsgolven gaan een interactie aan met het weefsel, waarbij terugkerende golven opgevangen worden door de piëzo-elektrische elementen. Deze interactie ontstaat door het verschil in akoestische impedantie tussen twee weefsels, door verstrooiing of door absorptie.[18] De akoestische impedantie is een eigenschap van elk soort weefsel die de weerstand tegen golfpropagatie beschrijft. Door een verschil in weerstand tussen twee weefseltypen wordt bij de overgang een deel van de geluidsgolf gereflecteerd.[18] Verstrooiing ontstaat doordat structuren kleiner zijn dan de golflengte van de geluidsgolf of een onregelmatig oppervlak hebben. Hierdoor worden de geluidsgolven verstrooid in het weefsel. Deze geluidsgolven komen lang niet allemaal terug bij de piëzo-elektrische elementen en zorgen dus voor een verzwakking van het signaal.[19] Absorptie is ook een eigenschap van elk soort weefsel. De absorptiecoëfficiënt geeft aan hoeveel van het invallende geluidsvermogen wordt geabsorbeerd.[20] De geluidsgolf wordt hierbij omgezet in warmte waarbij verzwakking van het signaal ook het gevolg is.[19] De intensiteit van het terugkerende signaal zorgt ervoor dat er een real-time afbeelding gereconstrueerd wordt.[18]

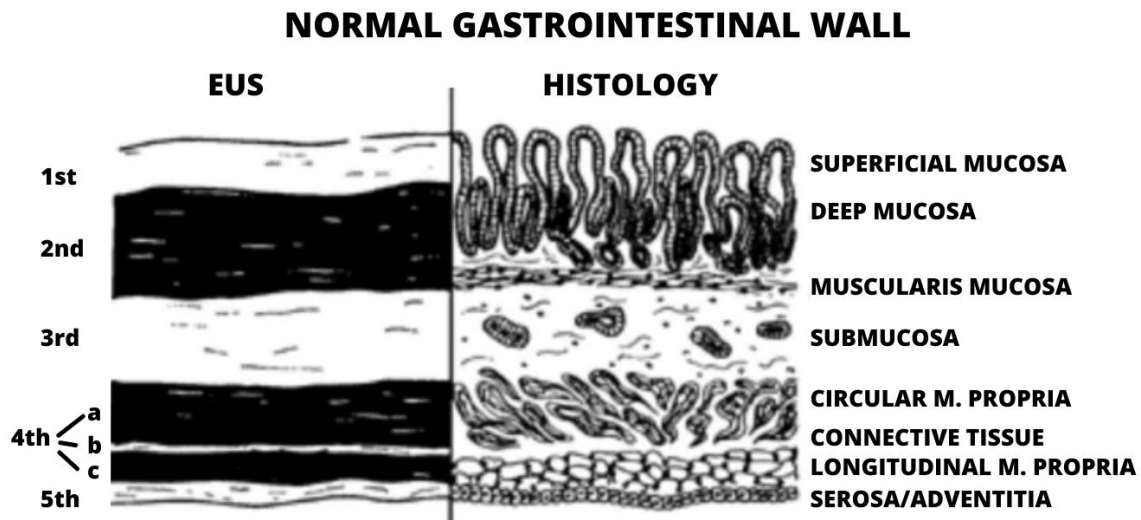
Er wordt gebruik gemaakt van twee typen EUS in de klinische praktijk, genaamd lineaire EUS en radiale EUS.[18,21,22] De lineaire EUS zendt parallel aan de endoscoop geluidsgolven uit, waarbij er een longitudinale doorsnede van het weefsel wordt afgebeeld. De radiale EUS zendt loodrecht op de as van de endoscoop geluidsgolven uit waarbij een 360 graden afbeelding gerealiseerd wordt.[21] De werking van beide typen EUS verschillen niet van elkaar, maar de real-time weergave van het weefsel wel (figuur 4). Het Deventer Ziekenhuis maakt gebruik van lineaire EUS, zodat ook getest kan worden of de spierwand mee verschuift bij verschuiving van de tumor of poliep. Dit is om te bekijken of de tumor of poliep aan de spierwand gehecht is en dus T1 of T2+ is.



Figuur 4: Verschieden tussen lineaire- en radiale EUS weergave van het weefsel in real-time.[22]

De verschillende lagen van de rectale wand zullen een ander beeld geven op het EUS-beeld (figuur 5). De mucosa zal over het algemeen donker zijn en wordt opgevolgd door een wat lichtere submucosa. Vervolgens geeft de muscularis propria een donker signaal af en sluit de serosa af met een wat licht signaal.[21] Wanneer een poliep aanwezig is op een EUS-beeld zijn daarin twee delen te onderscheiden. Het goedaardige deel van de poliep zal iets donkerder zijn dan de mucosa en het kwaadaardige deel zal nog wat donkerder zijn, maar niet zo donker als de muscularis propria. Bij een T1 RC is het donkere, kwaadaardige deel van de poliep niet door de lichte submucosa heen gegroeid. Bij een T2+ RC is dit wel het

geval en maakt het donkere kwaadaardige deel van de poliep dus contact met de donkere muscularis propria.[23]



Figuur 5: Correlatie tussen de vijf zichtbare lagen met behulp van EUS en de histologische lagen van de darmwand.[21]

DEEP LEARNING

Het grootste probleem met betrekking tot de stadiëring van RC's is de interbeoordelaarsvariabiliteit.[4] Een mogelijke oplossing hiervoor is het creëren van een objectief model dat de EUS-beelden automatisch beoordeelt, zoals een deep learning model. Deep learning is een vorm van *artificial intelligence* en is gebaseerd op biologische neurale netwerken.[24] Deze neurale netwerken bestaan uit neuronen die met elkaar verbonden zijn. De neuronen krijgen een input en produceren daarbij een output die ze weer doorgeven aan de volgende neuron via verbindingen. Deze verbindingen heten synapsen en bij iedere synaps hoort een bepaalde waarde, een *weight*. De *weight* bepaalt hoe de output van de ene neuron wordt doorgegeven als input van de volgende neuron.

Een veelgebruikte vorm van deep learning die ontworpen is om gegevens in een matrix zoals een afbeelding te verwerken, is een *convolutional neural network* (CNN). Het model bestaat uit verschillende lagen die elk een ander aspect van de afbeelding herkennen.[25] In de eerste lagen zijn dit simpele vormen, zoals lijnen, en hogere lagen kunnen de uitkomsten van eerdere lagen combineren om complexere structuren te herkennen.[26] Bij de verwerking van de data worden spatiële relaties behouden, wat benodigd is om afbeelding juist te verwerken.[27,28] Door deze componenten is het mogelijk om CNN's voor automatische beeldanalyse in te zetten, zo ook voor het maken van het onderscheid tussen T1 en T2+ gestadieerde RC's.

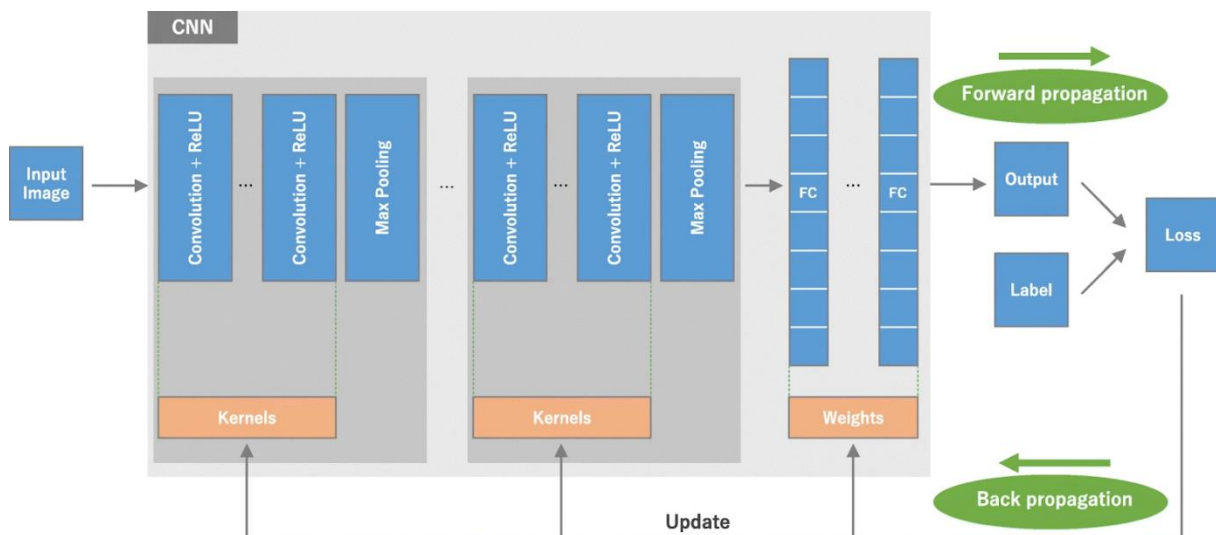
CONVOLUTIONAL NEURAL NETWORK

In de literatuur komt naar voren dat CNN's de meest gebruikte en zeer succesvolle vorm van deep learning zijn bij medische beeldanalyse.[29] Googlenet, een vorm van CNN, had bijvoorbeeld een significant hogere accuraatheid (89%) bij het detecteren van kanker dan pathologen (70%).[30] Een belangrijke reden hiervoor is dat een CNN in staat is om zelf te bepalen welke eigenschappen, oftewel *features*, belangrijk zijn door gebruik te maken van een trainingsset. Bij eerdere modellen was dit bijvoorbeeld niet mogelijk en ging er tijd verloren aan het handmatig selecteren van deze eigenschappen.[28] Door de toenemende complexiteit van de features per laag in de CNN is het ook mogelijk om patronen te herkennen. Dit is erg gunstig voor medische beeldanalyse, waar veel ingewikkelde anatomische beelden in voorkomen.[31] Ten slotte kan een CNN gebruik maken van

parallele *Graphics Processing Units*. Dit zorgt voor een snellere verwerking van beelden door het model, doordat meerdere taken op hetzelfde moment gedaan kunnen worden.[32,33]

COMPONENTEN CONVOLUTIONAL NEURAL NETWORK

Er zijn veel verschillende architecturen van CNN's, die over het algemeen uit dezelfde componenten zijn opgebouwd. In dit onderdeel zal worden besproken hoe het model de afbeeldingen interpreteert en welke componenten worden gebruikt voor een CNN. In figuur 6 is een algemene opbouw van een CNN te zien.



Figuur 6: Schematische weergave van de algemene structuur en de training van een CNN.[27]

AANGELEVERDE DATA

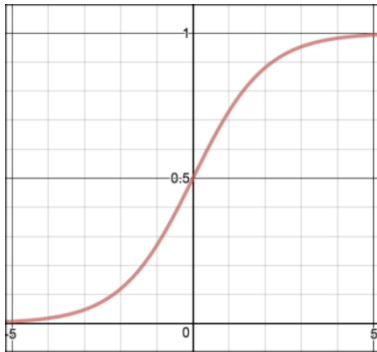
Een CNN krijgt data aangeleverd in de vorm van afbeeldingen. Een afbeelding is eigenlijk een matrix bestaande uit pixels welke de kleur, het contrast, de helderheid en de scherpte bepalen.[27] Zo kan een afbeelding bijvoorbeeld zwart-wit of in kleur zijn. De waarden van de pixels in een zwart-wit afbeelding worden bepaald aan de hand van grijs tinten. Hierbij krijgen de verschillende tinten een waarde van 0 tot 255, waarbij de waarde 0 de tint zwart aangeeft en de waarde 255 wit.[27] Daarnaast is er ook een RGB-kleurenschaal voor afbeeldingen in kleur. Een kleurenfoto bestaat fundamenteel uit drie lagen, rood, groen en blauw, waarmee elke bestaande kleur gemaakt kan worden. Hierbij geeft een pixelwaarde van 0 de tint zwart aan, en de maximale waarde van 255 de kleur rood, groen of blauw.[27] Als input voor de *convolutional layer*, die hierna besproken zal worden, worden de pixelwaarden van alle pixels in de afbeelding gebruikt.

CONVOLUTIONAL LAYER

De eerste laag van een CNN is vaak een convolutional layer. In deze laag wordt er een filter, bestaande uit één of meerdere *kernels*, over de afbeelding bewogen.[25,27] Kernels zijn matrices die verschillende weights bevatten, waarmee eigenschappen van de afbeelding uitgelicht worden. Er wordt een convolutie van de kernel met het betreffende gebied van de afbeelding gemaakt. Dit gebeurt door middel van een *dot-product* van de matrix met pixelwaarden uit het betreffende gebied van de afbeelding met het filter. De convolutie wordt in bijlage A besproken. Door telkens andere kernels te gebruiken, kunnen verschillende features van de afbeelding gedetecteerd worden. Deze features worden tezamen doorgegeven aan de daaropvolgende laag van het model.[25,27]

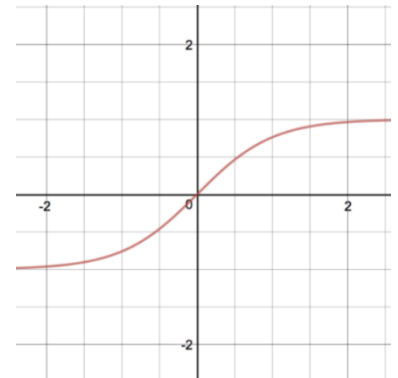
ACTIVATIEFUNCTIE

De activatiefunctie is een niet-lineaire functie die wordt toegepast om de data om te zetten van een lineair input signaal tot een non-lineair output signaal.[34] Daarbij controleert de activatiefunctie of en in welke mate de output van de ene laag door wordt gegeven aan de volgende laag. Een ander voordeel van de activatiefuncties is dat deze af te leiden zijn, wat belangrijk is voor de training van het model.[34] Er zijn veel verschillende activatiefuncties, maar vier varianten worden het meest gebruikt: de sigmoïd functie, de hyperbole tangent functie (Tanh), de softmax en de rectified linear unit (ReLU) functie.[35,36]

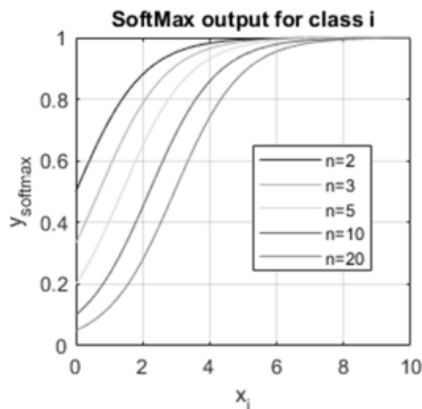


Figuur 7: Sigmoid functie.

Daarnaast is de tanh een veelgebruikte activatiefunctie. Deze lijkt op de sigmoïd functie, maar heeft twee grote verschillen: de output neemt een waarde tussen de -1 en 1 aan en de functie is gecentreerd rond de 0 (figuur 8). Dit laatste heeft vooral als voordeel dat het helpt bij het trainingsproces van meerlagige netwerken, waardoor deze functie wordt verkozen boven de sigmoïd functie. Deze functie wordt het meest gebruikt bij netwerken die ontworpen zijn voor het verwerken van taal.[34]

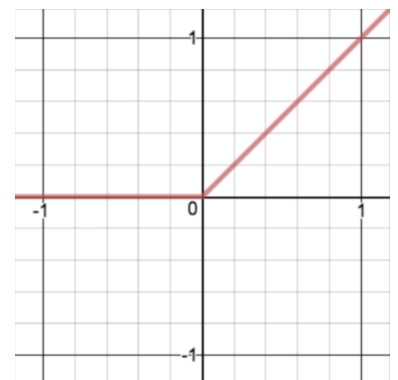


Figuur 8: Tanh functie.



Figuur 9: Softmax functie.

Als laatste is de ReLU activatiefunctie de meest gebruikte activatie functie voor een deep learning model. Deze functie geeft bij een negatieve input 0 als output en bij een positieve input is de output gelijk aan de input (figuur 10). Het voordeel van deze functie is dat de berekeningen snel verlopen, vanwege de relatief eenvoudige vergelijking.[34]



Figuur 10: ReLU functie.

POOLING LAYER

De *pooling layer* vermindert de hoeveelheid data en daarbij ook de rekenkracht die nodig is om de data te verwerken. Dit gebeurt door per blok van pixels bijvoorbeeld de hoogste ('*max pooling*'), gemiddelde ('*average pooling*') of laagste ('*min pooling*') waarde te nemen.[25,27,28] Zo worden meerdere pixelwaardes vervangen door 1 pixelwaarde in de output van de pooling layer. De pooling layer zorgt ervoor dat structuren goed worden herkend, ongeacht de plek in de afbeelding waar de structuur zich bevindt. Dit kan door het verkleinen van de hoeveelheid data in de afbeelding en tegelijkertijd het behouden van de relevante informatie en spatiële relaties.[25,27,28]

FULLY CONNECTED LAYER

Na meerdere convolutional en pooling layers worden de resultaten verwerkt tot een uitkomst in de *fully connected layer*. De convolutional en pooling layers zijn namelijk slechts in staat tot het verkrijgen van features uit de afbeelding en het verminderen van het aantal parameters. De fully connected layer deelt de features in groepen in, waarbij lijnen en andere vormen worden gecombineerd, om zo structuren, zoals een RC, als geheel te herkennen.[25,27,28]

TRAINEN VAN HET MODEL

Een CNN moet worden getraind om het verschil tussen de voorspelde uitkomst en de daadwerkelijke uitkomst te verkleinen. Een manier om dit tot stand te brengen is door veranderingen aan te brengen in de weights en kernels van het model. Aan het begin van de training heeft iedere synaps een bepaalde weight. Deze weight wordt bepaald aan de hand van een bepaalde *weight initialisation*, zoals *zero initialisation* of *random initialisation*. [25] De data die aan het model wordt geleverd, wordt vermenigvuldigd met de weight en vormt de input voor het volgende neuron. Op deze manier is er sprake van *forward propagation*, de data loopt van begin tot eind door het model.[27] Elke keer wanneer de data door het model loopt, genaamd een *epoch*, wordt het verschil tussen de daadwerkelijk uitkomst en de voorspelde uitkomst van de epoch berekend aan de hand van een *loss function*. [27] Deze loss function wordt daarbij weer meegenomen tijdens de volgende stap in het trainen van het model, de backpropagation. Bij backpropagation is het doel om de weights en kernels te updaten om uiteindelijk te zorgen dat de *loss* tussen de voorspelde uitkomst en de daadwerkelijke uitkomst verkleint. Deze waarde kan worden gevonden met behulp van verschillende loss functions zoals *gradient descent* en *batch size*. [27]

Een andere belangrijke waarde die hierbij meegenomen kan worden is de *learning rate*. De learning rate is een waarde tussen de 0 en de 1 die weergeeft hoe groot de stappen zijn waarmee het model de weights updatet na elke epoch.[27] Een grote waarde zal minder epochs nodig hebben, dus sneller tot een benadering te komen. Dit heeft wel als nadeel dat hierdoor een mogelijk betere benadering wordt overgeslagen. Een kleinere waarde zal daarom een betere benadering kunnen maken, maar dat duurt langer vanwege meerdere epochs.[37] Idealiter is het mogelijk dat de learning rate zich vanzelf kan aanpassen. Hierbij wordt er gekeken naar het verschil tussen de huidige loss en de loss van de volgende epoch en wordt de grootte van de learning rate aangepast aan de grootte van dit verschil.[38] Dit verkleint de kans dat de loss bij een lokaal minimum blijft hangen.

Telkens als het model de laagste loss bereikt slaat hij de waarden van de weights en biases op en vervolgt het model zijn training. Op die manier wordt de afwijking van de CNN tussen de voorspelde uitkomst en de daadwerkelijke uitkomst zoveel mogelijk verkleind.[38] Dit proces voor meerdere epochs doorlopen. Op deze manier wordt het model getraind tot een goed werkend model, waarna het model getest kan worden.

ONDERZOEKS- EN DEELVRAGEN

Voor dit onderzoek zijn een onderzoeksvraag en deelvragen opgesteld. Aan de hand van de methode is beantwoording van deze vragen mogelijk.

ONDERZOEKSVRAAG

Hoe kan geautomatiseerde beeldanalyse van endo-ultrasonografie beelden geoptimaliseerd worden ter onderscheiding van T1 en T2+ rectumcarcinomen?

DEELVRAGEN

- Hoe kunnen classificatie en segmentatie worden toegepast voor medische beeldanalyse van EUS-beelden?
- Aan welke eisen dienen de data te voldoen op het gebied van hoeveelheid en kwaliteit voor het trainen van een CNN model?
- Wat is de interbeoordelaarsvariabiliteit tussen studenten onderling en tussen de studenten en de expert bij het segmenteren van de EUS-beelden?
- Wat is het verschil in prestatie van een CNN model getraind met gesegmenteerde EUS-beelden door Technisch Geneeskunde studenten in vergelijking met gesegmenteerde EUS-beelden door een expert?

METHODE

Het onderzoek is opgebouwd uit drie verschillende subonderzoeken met ieder een eigen analyse en conclusie. De drie conclusies worden uiteindelijk samengevoegd tot één conclusie. Het eerste subonderzoek is een literatuuronderzoek waarbij wordt gekeken naar het verschil tussen classificatie en segmentatie en de eisen waaraan de data voor het trainen van een CNN model dient te voldoen. De kennis die hierbij wordt opgedaan, wordt gebruikt als achtergrondinformatie bij de andere subonderzoeken. Bij het literatuuronderzoek wordt gebruik gemaakt van databases zoals *Scopus*, *Google Scholar* en *Pubmed*.

Het tweede subonderzoek is opgesteld om te bepalen wat de interbeoordelaarsvariabiliteit is tussen TG-studenten onderling ten opzichte van de expert, Frank ter Borg, bij het segmenteren van EUS-beelden. TG-studenten hebben, na een korte instructie, individueel 10 EUS-beelden gesegmenteerd en deze vervolgens vergeleken met de segmentaties van de expert. Hiermee wordt bepaald wat de kwaliteit is van de segmentaties van de studenten.

Het derde subonderzoek focust zich op het verschil in prestatie van een CNN model getraind met gesegmenteerde EUS-beelden door TG-studenten in vergelijking met gesegmenteerde EUS-beelden door een expert. Hierbij wordt het model twee keer getraind met dezelfde EUS-beelden, maar de ene keer gesegmenteerd door de studenten en de andere keer gesegmenteerd door de expert. Dit model is gemaakt in de programmeertaal *Python* en wordt bewerkt in *Google Colab*.

Om een volledig beeld te krijgen van de procedure zal er ook in de praktijk mee worden gekeken met het klinische traject, de beoordeling en endoscopische behandeling van een T1 gestadieerd RC.

LITERATUURONDERZOEK

Voor het gebruik van automatische beeldanalyse met een deep learning model voor stadiëring van RC's zijn twee toepassingen interessant, classificatie en segmentatie. In dit literatuuronderzoek wordt er uitgebreid ingegaan op de verschillen en overeenkomsten van classificatie en segmentatie in architecturen, benodigde hoeveelheid data, trainen van kleine datasets en kwaliteit van de data. Hieruit zullen vervolgens enkele conclusies volgen over hoe automatische beeldanalyse met een deep learning model ter onderscheiding van T1 en T2+ RC's geoptimaliseerd kan worden.

CLASSIFICATIE

Classificatie is een toepassing van automatische beeldanalyse bij gebruik van een deep learning model en kan over het algemeen ingedeeld worden in twee stappen. De eerste stap is het verkrijgen van features uit een afbeelding. In een medische toepassing kan het beschreven worden als het maken van onderscheid tussen anatomische en pathologische features [28], zoals de darmwand en poliep bij de EUS-beelden. Door het model met vele verschillende EUS-beelden met een bekende diagnose als input te trainen leert het model steeds beter deze features te herkennen.[39] Op deze manier kan het model de EUS-beelden met een bepaalde waarschijnlijkheid toewijzen aan een klasse.[27] Een voordeel van classificatie is dat de features niet handmatig geselecteerd hoeven te worden, maar dat het model dit zelf herkent. Daarbij dient er een database aangemaakt te worden van honderden tot duizenden beelden om het model te trainen.[39] Als tweede stap worden de verkregen features gebruikt om niet eerder gebruikte EUS-beelden te classificeren.[40] Het model probeert in de dataset met test EUS-beelden de getrainde features te herkennen waarna het model een bepaalde klasse aan het EUS-beeld toebedeelt met bijbehorende waarschijnlijkheid.[27] In het geval van dit onderzoek geeft het model dan een beoordeling van een EUS-beeld in de vorm van een onderscheid tussen een T1 en T2+ gestadieërd RC. Met zo'n model zou een medisch professional geholpen kunnen worden bij de beoordeling van EUS-beelden en daarmee een beter onderscheid tussen de T1 en T2+ stadia RC's kunnen maken. Een nadeel hiervan is dat er bij een model op basis van classificatie geen visualisatie van de keuze van het model is: het is een *black box*. Een medisch professional krijgt dus niet te zien waarop het model de beoordeling heeft gebaseerd en moet kritisch blijven kijken naar de EUS-beelden. In deze toepassing is het wenselijk voor de medisch professional om hier wél inzicht in te krijgen, aangezien het verschil in behandeling tussen T1 en T2+ RC's een grote impact heeft op de kwaliteit van leven van de patiënt.

SEGMENTATIE

Naast classificatie is segmentatie ook een toepassing van automatische beeldanalyse bij gebruik van een deep learning model.[28,39] Dit houdt in dat een model getraind wordt om bepaalde structuren in een afbeelding, in dit geval de EUS-beelden, te herkennen en in segmenten te verdelen. Hierbij verschillen zowel de input als output van het model in vergelijking met classificatie. Als input voor automatische beeldanalyse worden zowel de EUS-beelden en de gesegmenteerde EUS-beelden aan het model geleverd, waarmee het model gaat trainen om features te herkennen en structuren te detecteren. Er zal aanvankelijk een dataset gecreëerd moeten worden met EUS-beelden waarin de structuren, oftewel de lagen van de darmwand, gesegmenteerd zijn. Een probleem hierbij is dat handmatige segmentatie tijdrovend is voor experts en bovendien onderhevig is aan grote interbeoordelaarsvariabiliteit.[4,28] In een model op basis van segmentatie wordt als output een segmentatie bestand gegeven met EUS-beeld gegeven waarin de structuren gesegmenteerd zijn. Op deze manier is voor de medisch professional de keuze van het model gevisualiseerd waarna een beoordeling kan worden gegeven over het stadium van het RC, T1 of T2+. Zo wordt de beoordeling van EUS-beelden meer geobjectiveerd en kan op basis van de uitkomst een meer geïnformeerde beslissing gemaakt worden wat betreft de behandeling.

CLASSIFICATIE ARCHITECTUREN

In de toepassing van classificatie bij automatische beeldanalyse worden CNN architecturen gebruikt op basis van een encoder structuur. Een encoder structuur zorgt ervoor dat de input, zoals een EUS-beeld, wordt omgezet in een gecodeerde output, waarin features zitten die de input representeren.[41,42] Deze gecodeerde output geeft daarbij de classificatie van RC's aan, T1 of T2+. Voor het gebruik van classificatie als toepassing bij EUS-beelden, zijn er verschillende deep-learning architecturen ontworpen op basis van een encoder structuur. De drie meest gebruikte CNN architecturen op het gebied van classificatie van RC's zijn ResNet, VGGNet en Inception.[28] Deze drie architecturen zullen in deze sectie nader worden toegelicht, waarvan de structuren te zien zijn in bijlage B.

RESNET

In de architectuur van een ResNet CNN worden met name 3x3 convolutional layers met een ReLu activatiefunctie en *batch normalization* gebruikt, met daarnaast enkele pooling layers. Een ResNet wordt opgebouwd uit meerdere *blocks* van lagen.[43] Een bijzonder element in zo'n block is de *skip connection*. Hierbij wordt de output van de ene laag niet slechts doorgegeven als input voor de volgende laag, maar worden ook één of meerdere lagen overgeslagen.[43]

Over het algemeen wordt aangenomen dat een deep learning model met meerdere lagen betere prestaties oplevert.[44] Dit komt doordat zulke modellen meer parameters evalueren tijdens de analyse van afbeeldingen. Echter, na een bepaalde hoeveelheid lagen, oftewel een bepaalde 'diepte' van het netwerk, nemen de prestaties af. Dit komt doordat de gradiënten van de loss functie van het model snel naar 0 dalen, waardoor de weights vervolgens niet meer aangepast worden en het model dus niet verder leert. Dit verschijnsel wordt *vanishing gradient* genoemd.[25] Door middel van de skip connections in de architectuur van een ResNet model wordt dit probleem voorkomen en blijft het model leren.

De ResNet architectuur wordt vooral gebruikt voor gezichtsherkenning en in medische wereld voor classificatie van medische beelden.[45] Zo is een ResNet model onder andere gebruikt voor de stadiëring van maagtumoren, en bij het classificeren van het type gliomen in de hersenen.[46,47]

VGGNET

Het *Visual Geometry Group Network* (VGGNet) is een encoder architectuur die gebruik maakt van relatief kleine 3x3 kernels tijdens de convoluties en een stride length van 1 heeft, om het aantal parameters te verminderen.[48,49] Daarnaast bevat het model pooling layers en een softmax layer aan het eind.[49]

Van deze architectuur zijn er verschillende typen ontworpen. De typen VGG16 en VGG19 hebben de beste werking, waarbij het getal staat voor het aantal lagen dat het model bezit.[28] Zo bestaat het VGG16 model uit 13 convolutional layers en 3 fully connected layers.[50] Met deze architecturen hebben de onderzoekers laten zien dat het gebruik van 16 tot 19 lagen met weights, zoals convolutional en fully connected layers, een positieve invloed heeft op de prestaties van het model.[48] Door het gebruik van meerdere kleine kernels achter elkaar kan de werking van grotere kernels zoals in andere architecturen worden nagebootst. Daarbij zorgen meerdere kleine kernels voor minder te trainen variabelen dan één grote kernel.[51] Hierdoor leert het model sneller nieuwe features en is het minder gevoelig voor overfitting. Echter, door de grotere hoeveelheid lagen en kernels zijn er meer parameters waardoor de training van het model uiteindelijk toch langzamer gaat dan andere systemen.[48,52]

De architecturen van VGGNet worden vaak gebruikt als basis van CNN's voor detectie en classificatie van objecten. Het is bovendien één van de meest gebruikte vormen van CNN's voor classificatie van beelden en de analyse van colonkanker.[28]

INCEPTION

De classificatie architectuur Inception bestaat uit zogenoemde *Inception modules*. De Inception modules maken gebruik van drie typen convoluties: 1x1, 3x3, en 5x5. De 1x1 convolutie blocks bevatten een ReLu activatiefunctie en zorgen ervoor dat de hoeveelheid data voor verwerking wordt verminderd. Deze laag dient als tussenstap voordat de data wordt doorgegeven aan de blocks met 3x3 en 5x5 convoluties. Naast het gebruik van deze drie typen convoluties, maakt de Inception module ook gebruik van een *global average pooling block*. Dit zorgt ervoor dat de resolutie van de afbeelding wordt verkleind.[53] Inception verschilt veel van het klassieke classificatie model door het gebruik van de eerdergenoemde Inception modules. Dit zorgt er namelijk voor dat in plaats van het verwerken van een afbeelding in een directe lijn van input naar output, ook sprake is van aftakkingen.[25,53,54] Op deze manier vinden er parallelle convoluties plaats waardoor verschillende feature maps worden gecreëerd. Deze feature maps worden vervolgens samengevoegd en gebundeld tot een input voor de volgende laag van het model.[25] In de medische wereld wordt de architectuur Inception onder andere gebruikt bij het classificeren van borstkanker en soorten bloedcellen.[55,56]

SEGMENTATIE ARCHITECTUREN

Als bij automatische beeldanalyse de toepassing segmentatie wordt gebruikt, maken CNN architecturen gebruik van een encoder-decoder structuur. De encoder structuur werkt met hetzelfde principe als bij automatische beeldanalyse op basis van de toepassing classificatie. De input van de decoder structuur is gelijk aan de output van de encoder structuur. Dit is gecodeerde data waarin features het EUS-beeld representeren.[57] Met deze input maakt de decoder structuur er weer een afbeelding van met dezelfde grootte als de input [42], maar dan opgedeeld in segmenten. De output geeft daarbij de gesegmenteerde lagen van de darmwand weer waarna een medisch professional uitspraak kan doen over de stadiëring.

Bij het creëren van een CNN voor automatische beeldanalyse met de toepassing segmentatie bij RC's worden met name drie architecturen veel gebruikt, namelijk de *fully convolutional network* (FCN), de *U-Net* en de *SegNet*. [28] In deze sectie zullen deze drie architecturen worden toegelicht, waarvan de structuren te zien zijn in bijlage C.

FULLY CONVOLUTIONAL NETWORKS

Een fully convolutional network (FCN) is een zeer succesvolle en veelgebruikte architectuur voor segmentatie.[28] Een FCN bestaat enkel uit convolutional en pooling layers. Hierbij worden de fully connected layers vervangen door een pooling layer en een deconvolutional layer.[58] De laatste drie lagen van een FCN model worden convolutional layers met kernelgroottes van 7x7, 1x1 en 1x1.[59]

In andere architecturen, zoals VGGNet en ResNet, wordt door middel van een fully connected layer aan het eind een afbeelding geclassificeerd. De informatie over de classificatie wordt na een softmax layer verkregen, maar deze informatie is één-dimensionaal. Hierdoor kan alleen de klasse van de hele afbeelding worden geïdentificeerd, niet slechts van één pixel. Om een afbeelding goed te kunnen segmenteren, moeten ook de afzonderlijke pixels kunnen worden gecategoriseerd. Doordat een FCN in plaats van een fully connected layer meerdere convolutional layers heeft, wordt aan het eind van het model een twee-dimensionale feature map verkregen. Hierdoor kunnen de pixels door middel van een softmax layer wel afzonderlijk geclassificeerd worden.[59]

Daarnaast kan een FCN beelden verwerken die in afmetingen afwijken van de beelden waarmee het model is getraind. De deconvolutional layer wordt namelijk gebruikt om de verkregen feature map uit te vergroten. Hiermee worden de afmetingen van de output van het model gelijk gemaakt aan de afmetingen van de input.[59] Het nadeel is dat de pooling layers mogelijk zorgen voor een output met een verlaagde resolutie.[39]

FCN modellen worden met name gebruikt voor semantische segmentatie in toepassingen zoals zelfrijdende auto's en segmentaties van bijvoorbeeld laesies in de lever en poliepen bij coloscopieën.[60,61]

U-NET

De U-Net is een architectuur in een U-vorm bestaande uit een *contracting path* en een *expansive path*. [62] Het doel van de contracting path is om de context van de afbeelding vast te leggen en van de expansive path om te helpen bij de nauwkeurige locatie van de features in de afbeelding. Bij elke stap in de contracting path wordt een 3x3 convolution, ReLU en 2x2 max-pooling layer toegepast waarna bij elke stap in de expansive path een 2x2 upconvolution, 3x3 convolution en ReLU layer wordt toegepast. De reden waarom een architectuur zoals U-Net geschikt is voor medische beeldanalyse, is dat eenvoudige en complexere features gecombineerd worden. Hierdoor kunnen de eenvoudige features de accuraatheid verhogen en de complexe features complexe functies extraheren. Daarnaast kan de output van de U-Net, door skip connections tussen de contracting en expansive path, spatiële lokalisatie behouden.[62,63] De segmentatie architectuur U-Net wordt veel gebruikt voor het segmenteren van medische beelden. De U-Net architectuur wordt veelvuldig gebruikt bij het segmenteren van tumoren uit CT-, MRI- of US-beelden.[64–66]

SEGNET

De SegNet architectuur maakt gebruik van dezelfde encoder structuur als de VGGNet architectuur bij de toepassing classificatie.[57] Een verschil is dat er een decoder structuur bijkomt met dezelfde lagen, maar dan in omgekeerde richting.[57] De laatste laag van de SegNet is een softmax layer, waarbij elke pixel wordt gebonden aan een label.[25] De decoder structuur van een SegNet maakt gebruik van een grotere pooling index waardoor een kleinere feature map ontstaat. Dit wordt hersteld naar de oorspronkelijke resolutie door middel van convoluties met een te trainen kernel. Het gebruik van een grotere pooling index heeft als voordeel dat de integriteit van high-frequency informatie en de rand karakterisatie wordt behouden maar het aantal leerbare parameters verlaagd wordt. Daarentegen zal het poolen van feature maps met een lage resolutie ervoor zorgen dat aangrenzende informatie genegeerd wordt.[59] SegNet is al in enkele medische toepassingen gebruikt, zoals het segmenteren van spieren, kraakbeen, botten en ter segmentatie van rode en witte bloedcellen onder de microscoop.[59]

BENODIGDE HOEVEELHEID DATA

Alvorens een model te trainen is het van belang dat er voldoende data aanwezig is, zodat het model adequate voorspellingen kan doen. De vraag die hierbij naar boven komt is dan ook: hoeveel data is er nodig om het model succesvol te trainen?

Bij classificatie is de prestatie van het model onder andere afhankelijk van de grootte van de trainingsset.[67] De trainingset dient genoeg data per klasse, dus EUS-beelden van zowel T1 als T2+ gestadieerde RC's, te bevatten, zodat het model de verschillende klassen kan leren onderscheiden. De vuistregel hierbij is dat een grotere trainingsset zorgt voor betere prestaties.[67] Ook is het belangrijk om rekening te houden met de mate van verschil tussen klassen. Het verschil maken tussen een zwarte of witte afbeelding is bijvoorbeeld relatief makkelijk en zal daarom weinig data nodig hebben.

Een studie uit 2016 van Cho et al. heeft bijvoorbeeld onderzoek gedaan naar het creëren van een learning curve van hun model door de trainingsset per klasse herhaaldelijk te vergroten.[68] Hierbij maakte zij gebruik van de architectuur GoogleNet en bestond de trainingsset uit verschillende groottes van CT-afbeeldingen. Met deze resultaten is vervolgens een learning curve opgesteld. Een learning curve is een plot van de prestaties van een model over de tijd, waaruit dus geconcludeerd kan worden hoe een model leert van de data die erin gestopt wordt. Met zo'n learning curve kan berekend worden hoeveel data benodigd is voor

een gewenste toepassing.[68] Daarnaast kunnen learning curves gebruikt worden om problemen te diagnosticeren wat betreft *underfitting* of *overfitting*. [69]

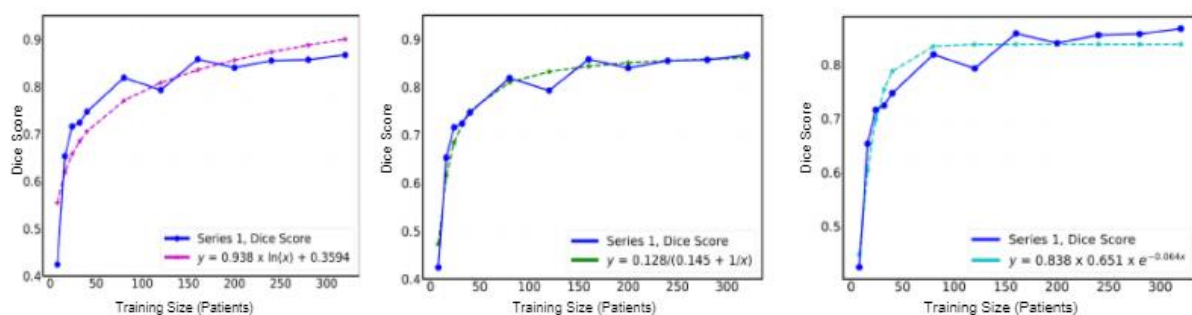
De berekende grootte van de dataset voor het model van Cho et al. is niet representatief voor een classificatiemodel voor het probleem bij stadiëring van RC's, maar de learning curve zou in dit geval wel toegepast kunnen worden. De learning curve wordt als volgt gedefinieerd:

$y = f(x; \mathbf{b}) = 100 + b_1 \times x^{b_2}$, waarbij b_1 en b_2 de richtingscoëfficiënt en vervalcoëfficiënt zijn, y gegeven wordt als de classificatie accuraatheid en x staat voor de grootte van de trainingsset.[68]

De vervalcoëfficiënt bepaalt hier in hoeverre de richtingscoëfficiënt afneemt.[70] De parameters b_1 en b_2 zijn te verkrijgen met behulp van *nonlinear regression*. Uiteindelijk zal het model bij een toenemende grootte van de trainingsset minimaal verbeteren, wat in de learning curve terug te zien is als een asymptotisch verband. De onderzoekers van Cho et al. stelden drie eisen op voor het creëren van een optimale learning curve: de trainingsset moet van hoge kwaliteit zijn, de sampling points moeten systematisch worden vergroot en elk sampling point moet meerdere keren worden herhaald voor het verkrijgen van een geschat, statistisch gemiddelde.[68]

Bij een segmentatie model hebben de onderzoekers Bardis et al. op eenzelfde manier gekeken naar de benodigde grootte van hun dataset.[71] Hierbij gebruikten zij 400 plaatjes voor het trainen van een U-Net model waarbij deze 12 keer getraind werd met telkens een verschillende verhouding van de groottes van de trainings- en validatieset.[71] Voor elke set werden de *Dice Similarity Coefficient* en de standaarddeviatie berekend om vervolgens te voorspellen wat de Dice Similarity Coefficient zal zijn bij een grotere trainingsset. Met deze gegevens werd er een curve gemaakt, waarbij de grootte van de trainingsset op de x-as staat en de Dice Similarity Coefficient op de y-as. Deze curve was benaderd met drie typen functies: een logaritmische, een asymptotische en een exponentiële. De exponentiële functie, zie de middelste grafiek in figuur 11, beschreef het best de curve van het model.[71] De *Hausdorff Distance* kan op een gelijke manier worden meegenomen in de bepaling van de optimale grootte van de trainingsset.

Mean Dice Score vs. Number of Cases



Figuur 11: In het onderzoek van Bardis et al. is de gemiddelde Dice Similarity Coefficient benaderd met meerdere functies. A) Logaritmische functie. B) Asymptotische functie. C) Exponentiële functie. Vanwege de laagste mean squared error is de asymptotische functie (B) de beste benadering.[71]

Uit dit onderzoek bleek, net als bij classificatie, dat er een optimale grootte is voor de trainingsset. Zo is in figuur 11 te zien dat er vanaf een grootte van 160 afbeeldingen in de trainingsset een plateau ontstaat. Vanaf dit punt zal bij een grotere trainingsset de Dice Similarity Score minimaal verbeteren en zal het model slechts stabiel worden.[71]

Een soortgelijk onderzoek kan opgesteld worden voor het segmentatiemodel voor geautomatiseerde analyse van EUS-beelden van RC's. Naar verwachting zal de optimale grootte van de trainingsset groter zijn dan in het onderzoek van Bardis et al. Dit komt doordat in dit segmentatiemodel meer labels geëvalueerd moeten worden en er dus ook meer parameters meespelen in het model. De grotere complexiteit van het model resulteert in een grotere benodigde trainingsset.

TRAINEN VAN KLEINE DATASETS

Voor het model dat gebruikt wordt in het onderzoek naar de modelvariabiliteit, wordt gebruik gemaakt van 40 gesegmenteerde EUS-beelden. Dit wordt gezien als een kleine dataset voor het trainen van een deep learning model.[72] Een veelvoorkomend probleem hierbij is overfitting, wat betekent dat het model te complex is voor het probleem. Het model heeft in een dergelijke situatie te veel geleerd van de afbeeldingen, doordat de weights te specifiek worden aangepast aan de kleine dataset. Een gevolg hiervan is dat bij nieuwe data de patronen minder goed herkend worden. Hierdoor is de accuraatheid van de segmentatie bij de trainingsset vele malen groter dan de accuraatheid bij de testset.[73] Er zijn verschillende manieren om overfitting te voorkomen bij het trainen van kleine datasets. Ten eerste is het belangrijk om de data te verdelen in drie groepen waarbij je een trainingsset, een validatieset en een testset hebt, in een verhouding van 60/20/20 (figuur 12).[74] In de literatuur wordt veelal aanbevolen om deze verdeling binnen de dataset aan te houden, in het geval van een kleine hoeveelheid beschikbare data.[75]



Figuur 12: Aanbevolen verdeling van een dataset bij het trainen van een Neural Network in het geval van een kleine dataset.[74]

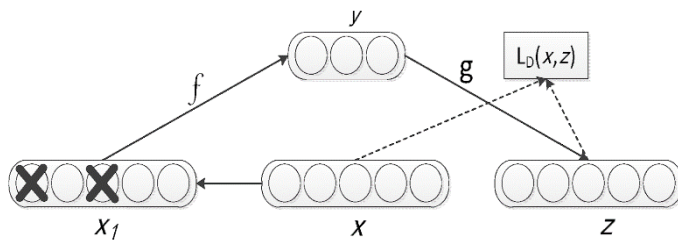
Ten tweede kan gebruik gemaakt worden van regularisatie. Regularisatie refereert naar een set van verschillende technieken die de complexiteit van het neurale netwerk verlaagt tijdens training en hiermee overfitting tegengaat. Er zijn verschillende vormen van regularisatie namelijk L1-regularisatie, L2-regularisatie, *dropout* en *batch normalisation*. [25]

L1- en L2-regularisatie berusten op het verkleinen van het aantal weights. Hierdoor zullen bepaalde features genegeerd worden, waardoor de resterende matrices minder complex zijn.[76] In het geval van L1-regularisatie kan de waarde 0 worden en in het geval van L2-regularisatie zal deze weight richting de 0 gaan zonder daadwerkelijk 0 te worden.[76]

Dropout gaat nog een stapje verder dan L1- en L2- regularisatie en is een van de meest gebruikte toepassingen van regularisatie.[25] Hierbij wordt bij elke epoch willekeurig een neuron uitgeschakeld tijdens training. Deze neuron zal daardoor niet deelnemen in de forward en backward propagation van de desbetreffende epoch. Dit zorgt ervoor dat het model geforceerd wordt om te leren van onafhankelijke eigenschappen en verkleint hiermee de kans op overfitting.[25]

Ook kan batch normalization helpen om overfitting tegen te gaan bij een kleine trainingsset. In principe is het een voorbewerkingslaag, waarin de outputs van elke laag worden genormaliseerd. Vanwege het continu updaten van de weights kan het namelijk zijn dat er een groot verschil ontstaat in de activatie van verschillende lagen. Naast regularisatie zorgt batch normalisation er ook voor dat de trainingstijd korter wordt en kan het vanishing gradient probleem vermeden worden.[25]

Ten derde kan er gebruik worden gemaakt van een soort architectuur genaamd *stacked denoising auto-encoder* (SDAE). SDAE is een structuur die bestaat uit denoising auto-encoders.[77] Dit zijn lagen van een model waarbij aan de input ruis wordt toegevoegd vooraf aan de training. Door ruis toe te voegen, wordt het model gedwongen om te leren het ruis weg te werken zodat 'schone' input wordt verkregen. Een visualisatie van de werking is te zien in figuur 13. Een voordeel hiervan is dat het model complexe features kan leren ontwikkelen met minder data.[78] Hierdoor kunnen meer complexe relaties tussen de input en output lagen herkend worden.[79] Deze laag wordt meerdere keren doorlopen om het model veel features te laten detecteren.



Figuur 13: Schematische weergave van een denoising auto-encoder laag waarbij x de input, x_1 de input met ruis, y de ontwikkelde feature door encoding van x_1 , z de output door decoding van y , en $L_D(x, y)$ de error.[77]

Als laatst kan het gebruik van data augmentatie wenselijk zijn bij een training met een kleine dataset. Bij data augmentatie wordt de dataset vergroot door de beschikbare beelden te transformeren. Onder deze transformaties vallen allerlei aanpassingen die aan een afbeelding gemaakt kunnen worden, zoals rotaties, flippen, bijsnijden, contrast aanbrengen, transleren etc. Op deze manier wordt er vanuit bestaande afbeeldingen 'nieuwe' data gecreëerd om het model beter te trainen.[25]

KWALITEIT VAN DATA

De kwaliteit van de aangeleverde data speelt ook een rol in de prestatie van het model. Tijdens het creëren van een dataset voor een segmentatiemodel dienen de vijf lagen van de rectumwand en RC in een grote hoeveelheid EUS-beelden te worden gesegmenteerd. Dit is een lastige taak aangezien de verschillende lagen niet altijd duidelijk te onderscheiden zijn. Bij het gebruik van *high definition* data zullen de vijf lagen van het rectum en de poliep beter te onderscheiden zijn.[80] Hierdoor zal het minder tijd kosten om een segmentatie te maken en zal de segmentatie accurater worden, wat leidt tot een verbetering van het model. Om deze reden wordt aangeraden om gebruik te maken van high definition EUS-beelden (1920x1080 pixels) in plaats van *standard definition* EUS-beelden bij het trainen van een segmentatiemodel.[81]

Bij classificatie is het ook van belang om afbeeldingen van een hoge kwaliteit te includeren in de trainingsset. De kwaliteit van de beelden heeft namelijk directe invloed op de accuraatheid van het model.[68] In de praktijk betekent dit dat een model, getraind met een lagere kwaliteit EUS-beelden, ofwel een grotere dataset nodig heeft om dezelfde accuraatheid te behalen als een model dat getraind is met een hogere kwaliteit EUS-beelden, ofwel deze accuraatheid nooit zou kunnen evenaren. Om deze reden wordt bij dit type model ook aangeraden om gebruik te maken van high definition EUS-beelden.

CONCLUSIE

In dit literatuuronderzoek is de afweging gemaakt welke toepassing van automatische beeldanalyse kan worden gebruikt voor het deep learning model: classificatie of segmentatie. Het voordeel van een classifierend model is dat het creëren van een trainingsset minder tijd kost, aangezien de beelden direct aan het model geleverd kunnen worden. Echter, het voordeel weegt niet op tegen het inzicht dat de medisch professional krijgt bij een segmenterend model. Bij de toepassing segmentatie zijn de verschillende lagen van de darm en poliep de output van het model, waarop een medisch professional zijn beoordeling van het RC met betrekking tot de stadiëring kan baseren. Bij classificatie zal de medisch professional onvoldoende inzichten krijgen in de beoordeling van het deep learning model, omdat de output slechts de waarschijnlijkheid van de diagnose T1 of T2+ is. Daarbij geeft de toepassing van segmenteren de mogelijkheid om nauwkeurig te bekijken hoe het model leert, waardoor er gemakkelijker aanpassingen aan het model gemaakt kunnen worden voor de juiste resultaten. Om deze redenen heeft een model dat gebaseerd is op het principe van segmentatie de voorkeur.

Daarnaast is er gekozen voor een U-Net model. Deze architectuur wordt binnen de medische wetenschappen veel gebruikt als vorm van CNN in soortgelijke toepassingen. Ook is een U-Net model, beschikbaar gesteld door Thom Nijboer. Verder zijn veelvoorkomende architecturen van modellen op basis van zowel classificatie als segmentatie besproken. Hieruit is geen concrete aanbeveling te halen over welke architectuur betere prestaties genereert.

De benodigde hoeveelheid data om een deep learning model te trainen bij de toepassing segmentatie is vast te stellen met een curve van de Dice Similarity Coefficient. Het aankomend onderzoek naar modelvariabiliteit maakt slechts gebruik van 40 gesegmenteerde EUS-beelden van TG-studenten en de expert. Dit komt door het gebrek aan tijd in het onderzoek. Voor een vervolgonderzoek zou onderzocht kunnen worden hoeveel EUS-beelden minimaal nodig zijn voor een representatieve uitkomst.

Bij het gebruik van een kleine hoeveelheid data is overfitting een veelvoorkomend probleem. Daarom is het belangrijk om daar in het aankomend onderzoek naar de modelvariabiliteit rekening mee te houden. Als eerste wordt dit gedaan door een goede verdeling van trainingsset, validatieset en testset te maken, en is er een mogelijkheid tot regularisatie ingebouwd in het model. Daarnaast zou ter optimalisatie van automatische beeldanalyse ook gebruik kunnen worden gemaakt van SDAE en data augmentatie in een vervolgonderzoek.

Voor het deep learning model is het als laatste beter om data te gebruiken met een high definition (1920x1080) kwaliteit. Vanwege de mogelijkheden in het Deventer Ziekenhuis kunnen high definition EUS-beelden niet opgeslagen worden. In de aankomende onderzoeken over de interbeoordelaarsvariabiliteit en modelvariabiliteit worden daardoor standard definition (640x480) EUS-beelden gebruikt.

INTERBEOORDELAARSVARIABILITEIT

Een probleem bij de beoordeling van de stadiëring van een RC is de interbeoordelaarsvariabiliteit. Door de subjectieve beoordeling van EUS-beelden is de betrouwbaarheid van de stadiëring van RC's op zijn minst twijfelachtig te noemen.[82] Een oplossing voor de hoge interbeoordelaarsvariabiliteit zou automatische beeldanalyse met een deep learning model zijn. Echter, voordat een dergelijk model gebruikt kan worden voor deze toepassing, zal het getraind moeten worden met veel gesegmenteerde EUS-beelden van RC's. Tot op heden is er nog geen grote database van deze segmentaties beschikbaar. Aanvankelijk zal dus een flink aantal EUS-beelden gesegmenteerd moeten worden.

Mogelijk zouden TG-studenten hier aan bij kunnen dragen, waarbij de vraag is of TG-studenten, met minder ervaring op dit gebied, kwalitatief goede segmentaties kunnen leveren. Om dit vast te stellen wordt in dit onderzoek geanalyseerd wat de interbeoordelaarsvariabiliteit en de kwaliteit van segmentaties van TG-studenten ten opzichte van de expert, MDL-arts Frank ter Borg, zijn. Hieruit kan geconcludeerd worden in hoeverre TG-studenten in staat zijn om EUS-beelden van RC's te segmenteren, om zo betrouwbare data te leveren aan een deep learning model.

METHODE

In dit onderzoek worden 10 dezelfde EUS-beelden door elke TG-student en de expert gesegmenteerd, waarbij er een onderscheid gemaakt wordt in de volgende lagen van het RC en de darmwand: *Muscularis propria*, *Submucosa*, *Mucosa*, *Poliep (goedaardig)* en *Poliep (kwaadaardig)*. Voor dit onderzoek zijn de EUS-beelden geleverd door het Deventer Ziekenhuis en zijn deze data door de expert beschikbaar gesteld in het programma *Castor EDC*. De EUS-beelden zijn voor dit onderzoek geselecteerd op basis van kwaliteit. De studenten konden de segmentaties van de expert niet inzien en mochten niet overleggen tijdens het maken van een segmentatie. De segmentaties zijn gemaakt in het programma *3D Slicer*. Tijdens de evaluatie van de segmentaties zijn de lengtes van de lagen gelijk gemaakt, zodat in de vergelijking van de segmentaties van de studenten met die van de expert, de resultaten niet beïnvloed zouden worden door afwijkende lengtes van de gesegmenteerde lagen.

Naast de vergelijking van de onderlinge beelden vindt er ook een vergelijking plaats met de gemiddelde resultaten en de gecombineerde segmentaties ten opzichte van de expert. De gemiddelde resultaten zijn verworven door de gemiddelde waarde van de vergelijking van de studenten ten opzichte van de expert per laag te berekenen. De gecombineerde segmentaties zijn ontstaan door de segmentaties van de studenten bij elkaar op te tellen met behulp van een Python script, te zien in bijlage D.

EVALUATIE

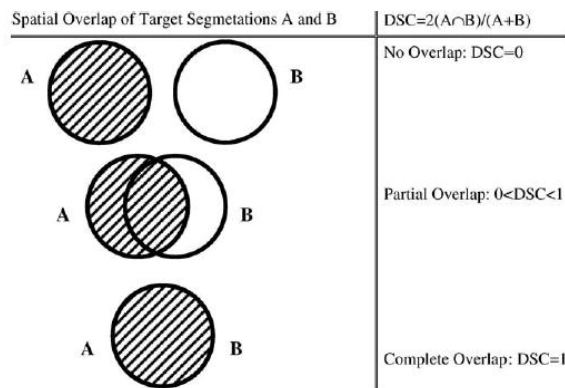
Bij het beoordelen van rectale EUS-beelden is de algehele accuraatheid van belang om te beoordelen of de tumor een T1 of een T2+ is. Om deze reden dient er gebruikt gemaakt te worden van een waarde die de overeenkomst tussen de segmentaties beoordeelt en een waarde die de spatiële verhouding tussen de segmentaties berekent. Hierbij wordt gebruik gemaakt van de Dice Similarity Coëfficiënt (DSC) en de Average Hausdorff Distance (AHD) voor elke gesegmenteerde laag. Daarnaast wordt de Kappa-waarde gebruikt voor de interbeoordelaarsbetrouwbaarheid tussen TG-studenten.

DICE SIMILARITY COEFFICIENT

De DSC is een grootheid voor de statistische analyse waarbij twee datasets met elkaar worden vergeleken. De DSC wordt in de medische wereld veel gebruikt op het gebied van het vergelijken van handmatige segmentaties en segmentaties gemaakt op basis van deep learning.[83] Bij deze vergelijking wordt de mate van overlap tussen de twee datasets berekend, aan de hand van de volgende formule:

$DSC(A, B) = \frac{2(A \cap B)}{A + B}$, waarbij A en B twee datasets representeren en $\cap$ de intersectie tussen beide datasets is.

De DSC kent een waarde tussen de 0 en de 1, waarbij een DSC van 0 betekent dat er sprake is van geen overlap en een DSC van 1 betekent dat er sprake is van totale overlap tussen de datasets (figuur 14).[84] In dit onderzoek wordt de DSC bepaald door de tien gesegmenteerde EUS-beelden per TG-student te vergelijken met de tien gesegmenteerde EUS-beelden van de expert. Dit wordt per gesegmenteerde laag met behulp van een script in Python berekend (bijlage D) en vervolgens gevisualiseerd in een boxplot.



Figuur 14: Visualisatie van de Dice Similarity Coefficient, waarmee spatiële overlap en reproduceerbaarheid kan worden weergegeven.[84]

HAUSDORFF DISTANCE

Naast de DSC wordt bij segmentatie van medische beelden de Hausdorff Distance (HD) gebruikt om segmentaties met elkaar te vergelijken.[85] Met behulp van de HD is namelijk de afstand tussen twee sets van punten, in dit geval twee segmentaties, te meten.[85] De berekende afstand in pixels geeft de minimale grootte van uitbreiding aan die nodig is voor de ene segmentatie om de andere segmentatie te bevatten.[85] Over het algemeen is de HD sensitief voor uitschieters. Om deze reden wordt er gebruik gemaakt van de Average Hausdorff Distance (AHD), die stabiel is.[83] De AHD is te berekenen met de volgende formule:

$AHD(X, Y) = (\frac{1}{X} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{Y} \sum_{y \in Y} \min_{x \in X} d(x, y)) / 2$, waarbij X en Y de sets met punten zijn van de segmentatie.

Hierbij worden de minimale afstanden bij elkaar opgeteld en gedeeld door het aantal punten in de dataset. Ook voor de evaluatie van de ADH wordt een Python script (bijlage D) gebruikt, waarna de resultaten worden gevisualiseerd met behulp van een boxplot.

KAPPA

Voor de interbeoordelaarsbetrouwbaarheid wordt de Kappa berekend.[86] De Kappa neemt een waarde van -1 tot 1 aan, waarmee een uitspraak gedaan kan worden in hoeverre de overeenstemming tussen verschillende beoordelaars op kans berust. Hierbij geeft een waarde van 0 bijvoorbeeld aan dat de overeenstemming volledig op kans berust. In tabel 1 is weergegeven hoe de overige waardes van Kappa te interpreteren zijn en wat dat zegt over de betrouwbaarheid van de data.[86]

Tabel 1: Interpretatie van Kappa.[86]

KAPPA WAARDE	MATE VAN OVEREENSTEMMING	BETROUWBAARHEIDSPERCNTAGE
0 - 0.20	Geen	0 - 4%
0.21 - 0.39	Minimaal	4 - 15%

0.40 - 0.59	Zwak	16 - 35%
0.60 - 0.79	Gemiddeld	36 - 63%
0.80 - 0.90	Sterk	64 - 81%
0.90 <	Bijna perfect	82 - 100%

De kappa is te berekenen met behulp van de volgende formule:

$k = \frac{\text{Pr}(a) - \text{Pr}(e)}{1 - \text{Pr}(e)}$, waarbij $\text{Pr}(a)$ de waargenomen overeenkomst en $\text{Pr}(e)$ de toevalsovereenkomst is.[86]

In dit onderzoek wordt de Kappa tussen de TG-studenten onderling berekend aan de hand van de gegeven stadiëring per EUS-beeld. De expert wordt hierbij niet vergeleken, aangezien deze informatie bezit over de uitkomst van het pathologisch onderzoek (PA). Op deze manier geeft de uitkomst informatie over de mate van overeenstemming van de gegeven stadiëring tussen TG-studenten.

RESULTATEN

De resultaten van dit onderzoek zijn weergegeven in bijlage F en G. In bijlage F zijn de segmentaties te vinden en in bijlage G de boxplots. Op het gebied van de DSC en AHD zijn de mediaan, de *Interquartile Range* (IQR) en de gemiddelde waarde weergegeven in tabel 2 en 3. voor de *Muscularis propria*, *Submucosa*, *Mucosa*, *Poliep (goedaardig)* en *Poliep (kwaadaardig)*. In tabel 2 zijn de resultaten te zien tussen de TG-studenten ten opzichte van de expert. In tabel 3 zijn de resultaten te zien van de gecombineerde segmentatie ten opzichte van de expert.

De *Muscularis propria* heeft de hoogste gemiddelde DSC met een waarde van 0,6171. Daarnaast heeft het een lage gemiddelde AHD van 4,4069. De IQR van beide scores is gemiddeld, respectievelijk 0,2347 en 1,6276. Bij de combinatie van de studenten t.o.v. de expert is de gemiddelde DSC score hoger, 0,7009, en de gemiddelde AHD score lager, 4,0711. De *Submucosa* heeft bij zowel de DSC als de AHD een hoge IQR, 0,2867 en 2,4513. Bij de combi zijn de IQR waardes al lager, maar nog steeds hoger dan de andere lagen, met respectievelijk 0,1743 en 1,8040.

De *Mucosa* heeft de laagste gemiddelde DSC met een waarde van 0,4541 en de laagste ADH met een waarde van 3,8657. De IQR is in beide gevallen laag, 0,2867 en 2,4513.

De *Poliep (goedaardig)* heeft lage IQR's met 0,1486 voor DSC en 0,8933 voor AHD.

Tabel 2: Gemiddelde segmentaties tussen TG-studenten

	Muscularis propria	Submucosa	Mucosa	Poliep (goedaardig)	Poliep (kwaadaardig)
	(Mediaan ± IQR)				
DSC	0,7171 ± 0,2347	0,5413 ± 0,2867	0,4634 ± 0,2199	0,5379 ± 0,1486	0,5209 ± 0,2771
HD	4,3275 ± 1,6276	5,1917 ± 2,4513	3,8151 ± 0,8288	5,6407 ± 0,8933	5,1157 ± 1,0623
	(Gemiddelde)				
DSC	0,6171	0,4849	0,4541	0,4921	0,5816
HD	4,4069	5,2290	3,8657	5,6461	5,0450

De Kappa-waarden van elke TG-student afzonderlijk ten opzichte van de andere TG-studenten zijn respectievelijk 0,3740, 0,2563, 0,2255 en 0,3281. De gemiddelde Kappa-waarde tussen TG-studenten is daarmee gelijk aan 0,2960.

Uit de resultaten van de gecombineerde segmentaties ten opzichte van de expert blijkt dat bijna alle scores per laag, zowel de DSC als de AHD, verbeterd zijn vergeleken met de

resultaten van de gemiddelde segmentaties per laag. Alleen de mediaan van de AHD en het gemiddelde van de DSC in *Poliep (goedaardig)* zijn minder goed dan die van de resultaten van de gemiddelde segmentaties. Bovendien is de IQR van de mediaan van de DSC in de lagen *Muscularis propria*, *Submucosa*, *Mucosa* en *Poliep (kwaadaardig)* significant verkleind.

Tabel 3: Gecombineerde segmentaties van TG-studenten.

	Muscularis propria	Submucosa	Mucosa	Poliep (goedaardig)	Poliep (kwaadaardig)
	(Mediaan $\pm$ IQR)				
DSC	0,7783 $\pm$ 0,1422	0,6328 $\pm$ 0,1743	0,5314 $\pm$ 0,0972	0,5830 $\pm$ 0,1492	0,6315 $\pm$ 0,1669
HD	3,7357 $\pm$ 1,0046	5,0156 $\pm$ 1,8040	3,5434 $\pm$ 0,8029	5,6896 $\pm$ 0,5497	5,0518 $\pm$ 0,9476
	(Gemiddelde)				
DSC	0,7009	0,5613	0,5186	0,4723	0,5951
HD	4,0711	5,1878	3,4255	5,5220	4,9261

CONCLUSIE

Op basis van de DSC is de laag *Muscularis propria* het beste gesegmenteerd door de TG-studenten ten opzichte van de expert. Deze is namelijk het hoogst qua gemiddelde en mediaan in de resultaten van zowel de gemiddelde segmentaties als die van de gecombineerde segmentaties. Verder is in de boxplot het verschil tussen de TG-studenten bij de DSC duidelijk te zien op het gebied van spreiding. De boxplots variëren per student namelijk in lengte, wat terug te zien is in de verschillen in IQR per TG-student per laag. Dit duidt op een inconsistentie van segmenteren tussen TG-studenten. Bij de laag *Submucosa* is deze inconsistentie het hoogst. Bij de *Poliep (goedaardig)* in de gemiddelde segmentaties en *Mucosa* in de gecombineerde segmentaties zijn de waarden voor IQR het laagst, wat betekent dat deze laag het meest consistent werd getekend door de TG-studenten t.o.v. de expert. Doordat in de gecombineerde segmentaties uitschieters werden gefilterd in het Python script, werden waarden van IQR in alle lagen behalve *Poliep (goedaardig)* kleiner, wat duidt op een verbeterde consistentie van de segmentaties. Op het gebied van overeenkomst met de expert scoort de *Mucosa* het laagst vanwege de lage medianen en het laagste gemiddelde van de DSC.

Bij AHD liggen de gemiddelden in een range van 1,7804 pixels, wat betekent dat de spreiding tussen de TG-studenten niet groot is. Toch ligt het gemiddelde van alle lagen op 4,8385 pixels, wat aangeeft dat het verschil tussen TG-studenten en de expert zeker aanwezig is. De medianen van de AHD hebben de hoogste waarden in de laag *Poliep (goedaardig)*. Dit betekent dat deze laag het meest moet worden uitgebreid om de segmentatie van de expert te bevatten. Daar staat tegenover dat de laag *Mucosa* het minst moet worden uitgebreid. De IQR van de AHD is over het algemeen het hoogst in de laag *Submucosa*. Hieruit is te concluderen dat ook bij de ADH in deze laag de grootste inconsistentie te zien is tussen TG-studenten. De IQR van de AHD is bij de gemiddelde segmentaties het laagst in de laag *Mucosa*, wat daarentegen een lage inconsistentie laat zien tussen TG-studenten. Daarnaast is deze waarde bij de gecombineerde segmentaties het laagst in de laag *Poliep (goedaardig)*, wat in dit geval dus duidt op een relatief hoge consistentie.

Wat betreft de Kappa is de gemiddelde Kappa tussen TG-studenten gelijk aan 0,2960. Deze waarde geeft dus een indicatie over de mate waarin de stadiëringen van RC's in EUS-beelden van verschillende TG-studenten meer overeenkomst hebben dan wat puur op basis van toeval verwacht zou worden. Deze Kappa wordt volgens de nomenclatuur van Landis et al. geclassificeerd als een matige overeenkomst tussen verschillende beoordelaars.[87]

Kortom, op basis van de IQR van de DSC geeft de laag *Poliep (goedaardig)* de kleinste interbeoordelaarsvariabiliteit tussen TG-studenten ten opzichte van de expert. Daarnaast is

de laag *Muscularis propria* het best gesegmenteerd op basis van de DSC, maar heeft deze laag ook een hoge IQR. De laag *Mucosa* is volgens de DSC het slechtst gesegmenteerd tussen TG-studenten qua mediaan en gemiddelde. De laag *Submucosa* scoort voor de DSC en de AHD het laagst qua IQR, wat betekent dat deze laag de grootste interbeoordelaarsvariabiliteit aangeeft tussen TG-studenten ten opzichte van de expert. Op basis van de IQR van de AHD geeft de laag *Mucosa* de kleinste interbeoordelaarsvariabiliteit. De laag *Mucosa* moet ook het minst uitgebreid worden om de segmentatie van de expert te bevatten op het gebied van mediaan en gemiddelde. Daarbij is de laag, die het meest uitgebreid moet worden, *Poliep (goedaardig)*. Volgens de Kappa is er een matige overeenkomst tussen de stadiëringen van RC's in EUS-beelden van TG-studenten dan wat puur op basis van toeval verwacht zou worden.

MODELVARIABILITEIT

In het vorige onderzoek zijn de interbeoordelaarsvariabiliteit en de kwaliteit van de segmentaties door TG-studenten vergeleken ten opzichte van de expert. In dit onderzoek wordt geëvalueerd wat het verschil in prestatie is van het CNN model dat enerzijds getraind is met segmentaties van de TG-studenten en anderzijds getraind is met segmentaties van een expert. Dit heeft als doel om vast te stellen wat de invloed van de interbeoordelaarsvariabiliteit is op de kwaliteit van de segmentaties gemaakt door het model.

METHODE

In dit onderzoek wordt er gebruik gemaakt van een U-net model in *Google Colab* gemaakt door Thom Nijboer. Voor het gebruik van het model zijn er een aantal aanpassingen gemaakt in het script, deze zijn in bijlage G groen gemarkeerd. Bij het onderzoek wordt dit model vervolgens viermaal getraind: twee keer met de dataset van dr. Frank ter Borg en twee keer met de dataset van TG-studenten. Elke dataset bestaat uit 40 gesegmenteerde EUS-beelden, waarvan er 30 worden gebruikt voor training en 10 voor validatie. Dit is volgens de 60/20/20 verdeling om overfitting tegen te gaan. De trainings- en validatiebeelden worden willekeurig door het model geselecteerd. Na de training wordt de output van de modellen geëvalueerd door 10 gekozen EUS-beelden als input te leveren. Deze 10 beelden zijn ook gesegmenteerd door de expert en deze segmentaties dienen als de *ground truth* in de vergelijking met de uitkomst van het model. Na elke training worden alle variabelen gereset. Op deze manier kan worden geëvalueerd in hoeverre de training met beelden uit verschillende databases (TG-studenten t.o.v. expert) de kwaliteit van de segmentaties van het model beïnvloedt.

EVALUATIE

Na het trainen van het model met segmentaties van de TG-studenten en van de expert, wordt de uitkomst geëvalueerd met behulp van tien testbeelden. De tien testbeelden worden aan het model geleverd en daarbij gesegmenteerd. Net als in het vorige onderzoek over de interbeoordelaarsvariabiliteit, worden deze segmentaties vergeleken met de segmentaties die de expert heeft gemaakt van de testbeelden met behulp van de DSC en AHD. De DSC en AHD zullen per laag geëvalueerd worden aan de hand van een boxplot. Zo kan er een uitspraak gedaan worden over de prestatie van het model op basis van de training met verschillende inputs, segmentaties van TG-studenten of van de expert.

RESULTATEN

Een overzicht van de algehele uitkomsten is gegeven in bijlage H en I, waarbij in H de uitkomsten van het model zijn gevisualiseerd en in I de vergelijking wordt weergegeven in boxplots. In tabel 4 en 5 zijn de (gemiddelde) DSC & AHD en de bijbehorende mediaan en IQR weergegeven van expert en TG-studenten. De scores zijn gegeven per segmentatie laag, waarbij de resultaten van de afzonderlijke trainingen op de horizontale as staan.

Voor de *Muscularis Propria* zijn de waarden van de studenten licht verhoogd t.o.v. van de expert. Zo hebben de studenten een mediaan bij DSC op 0,2963 met een IQR van 0,1408 en bij AHD een mediaan op 5,6387 met een IQR van 1,5694. Daartegenover heeft de expert bij de DSC een mediaan op 0,2423 met een IQR van 0,1863 en bij AHD een mediaan op 5,8151 met een IQR van 1,261.

Hiernaast laten de segmentaties van de *Submucosa* voor beide systemen de beste waarden zien. Zo is de mediaan van de DSC bij de studenten 0,3587 en bij de expert 0,3404 met respectievelijk een IQR van 0,1622 en 0,1748. Daarbij is de mediaan van de AHD bij de studenten gelijk aan 5,6965 pixels en bij de expert 5,9032 pixels, met een respectievelijke IQR van 1,2672 en 1,5232 pixels.

Bij de *Mucosa* is voor zowel de uitkomst van de studenten als die van de expert de mediaan van de DSC laag, respectievelijk 0,1757 en 0,1692. Daarbij is de IQR in beide gevallen ook

laag, 0,1301 en 0,1412. Bij de AHD is de mediaan laag met een hoge IQR, $4,2106 \pm 1,8968$ en $3,9910 \pm 2,0986$.

Het deel *Poliep (kwaadaardig)* heeft zowel bij de studenten als bij de expert voor de DSC een relatief hoge mediaan, maar is bij de studenten iets hoger, 0,2983 t.o.v. 0,2699. De IQR is bij beide ook relatief hoog, met voor de studenten 0,2637 en voor de expert 0,2538. De mediaan bij de AHD ligt ook relatief hoog, met 6,2844 voor de studenten en 6,4192 voor de expert. Bij de IQR is een groter verschil te zien tussen de studenten, 1,0099, en de expert, 1,7418.

Tabel 4: Resultaten model getraind door segmentaties van expert.

Model Expert	Muscularis propria	Submucosa	Mucosa	Poliep (goedaardig)	Poliep (kwaadaardig)
(Mediaan $\pm$ IQR)					
DSC	$0,2423 \pm 0,1863$	$0,3404 \pm 0,1748$	$0,1692 \pm 0,1412$	$0,1536 \pm 0,2965$	$0,2699 \pm 0,2538$
AHD	$5,8151 \pm 1,261$	$5,9032 \pm 1,5232$	$3,9910 \pm 2,0986$	$6,0837 \pm 1,3389$	$6,4192 \pm 1,7418$
(Gemiddelde)					
DSC	0,2516	0,3097	0,1756	0,2310	0,2401
AHD	5,9016	5,8768	4,2498	6,0964	6,1590

Tabel 5: Resultaten model getraind door segmentaties van TG-studenten

Model TG'ers	Muscularis propria	Submucosa	Mucosa	Poliep (goedaardig)	Poliep (kwaadaardig)
(Mediaan $\pm$ IQR)					
DSC	$0,2963 \pm 0,1408$	$0,3587 \pm 0,1622$	$0,1757 \pm 0,1301$	$0,2186 \pm 0,2126$	$0,2983 \pm 0,2637$
AHD	$5,6387 \pm 1,5694$	$5,6965 \pm 1,2672$	$4,2106 \pm 1,8968$	$6,0837 \pm 1,5797$	$6,2844 \pm 1,0099$
(Gemiddelde)					
DSC	0,3038	0,3280	0,1774	0,2041	0,2937
AHD	5,6387	5,6709	4,4433	6,0537	6,3490

CONCLUSIE

De *Submucosa* is de best gesegmenteerde laag van beide modellen op basis van de gemiddelde DSC. Daarbij heeft deze laag in beide gevallen een lage IQR wat betekent dat de variabiliteit van de verschillende resultaten klein is. De modellen hebben de *Submucosa* dus telkens gesegmenteerd met een gelijke precisie, waardoor van een nieuwe segmentatie goed te zeggen is hoe betrouwbaar de gesegmenteerde *Submucosa* is.

De *Muscularis propria* is na de *Submucosa* de best gesegmenteerde laag op basis van de gemiddelde DSC. Ook de gemiddelde ADH score is hierbij relatief goed. Vermoedelijk wordt deze laag zo gesegmenteerd, omdat het een duidelijke donkere kleur heeft omringd door een lichte laag van de *Submucosa* en de serosa.

De *Mucosa* heeft bij beide modellen een lage gemiddelde DSC, maar ook een lage gemiddelde ADH. Dit komt waarschijnlijk doordat de *Mucosa* altijd maar een klein gebied omvat in de afbeelding. Dit betekent dat een segmentatie van het model al snel niet goed overlapt met de ground truth en dus een slechte DSC oplevert. Echter, door de segmentatie van het model vaak maar een klein stukje uit te breiden, overlapt deze wel met de ground truth, waardoor de ADH laag is.

De laag *Poliep (kwaadaardig)* heeft in beide gevallen een relatief hoge gemiddelde DSC, maar daarbij ook een hoge IQR. Dit betekent dat de DSC's van verschillende segmentaties door hetzelfde model heel erg van elkaar verschillen. Hierdoor is niet te zeggen hoe betrouwbaar de gesegmenteerde *Poliep (kwaadaardig)* is wanneer het model een segmentatie creëert. Daarbij heeft deze laag ook de hoogste ADH van alle lagen.

Er is dus een duidelijk verschil te zien in de gemiddelde DSC's en AHD's van de verschillende lagen bij beide systemen. In de resultaten zijn de scores van de *Poliep (goedaardig)* niet uitgelicht. De reden hiervoor is dat de scores zeer gemiddeld zijn en deze laag daarnaast minder interessant is. Bovendien is het belangrijk om te benoemen dat de DSC en AHD scores van de lagen niet optimaal zijn. De modellen presteren nog niet goed genoeg om daadwerkelijk in de kliniek gebruikt te worden.

DISCUSSIE

In de volgende sectie zullen zowel verbeterpunten voor een vervolgonderzoek als discussiepunten van het huidige onderzoek worden uiteengezet voor de drie verschillende onderzoeken.

LITERATUURONDERZOEK

Zoals vermeld is er bij classificatie sprake van een black box: medisch professionals kunnen niet zien waarop het systeem de classificering van EUS-beelden baseert. Dit maakt dat er veel vertrouwen in het model gesteld moet worden, wanneer een medisch professional zijn/haar beoordeling (deels) baseert op de classificering van de beelden. Voordat dit gedaan wordt, zal het model dermate verbeterd moeten worden dat de betrouwbaarheid ervan nagenoeg perfect is. Ook zou geprobeerd kunnen worden om de black box op te heffen, om zo de keuzes waarop de stadiëring door het systeem gebaseerd wordt, beter te visualiseren. Een mogelijkheid om dit te doen, zou *feature visualization* kunnen zijn.

Bij feature visualization wordt er gekeken naar welke features een model, of een bepaald onderdeel van het model, detecteert.[88] Dit werkt zowel op een kleine schaal, waarbij individuele features van een neuron zichtbaar gemaakt kunnen worden, als op een grote schaal, waarbij er gekeken wordt naar de groeperingen van features die het model gebruikt om een klasse te verbinden aan het object.[88] Vooral het laatste is interessant voor de automatische beeldanalyse van EUS-beelden. Op deze manier kan namelijk zichtbaar worden gemaakt op welke features het model selecteert voordat deze een beslissing maakt. Hierbij wordt aangeraden om de visualisatie te nemen van de laag vóór de softmax functie, aangezien dit een betere kwaliteit geeft.[88]

Echter, de feature visualisatie geeft enkel inzicht over de algemene beslissingen van het model op basis van features en niet over de individuele features van de EUS-afbeeldingen. Om deze reden is dit niet de meest geschikte aanpak om de medisch professional inzicht te geven in het model. Het kan daarentegen wel helpen bij het optimaliseren van het model. De afzonderlijke feature visualizations vertellen namelijk waar een model echt naar op zoek is. Zoals in figuur 15 te zien is, kan bij deze modellen de vraag gesteld worden of ze op zoek zijn naar het echte object, zoals een honkbal of een gebouw, of dat ze slechts selecteren op de features die ermee samenhangen, zoals strepen of de lucht. Met dit inzicht kan het leerproces van het model geëvalueerd en verbeterd worden, waardoor er duidelijk wordt wat de best mogelijke input is voor het model.[88,89]



Baseball—or stripes?

Animal faces—or snouts?

Clouds—or fluffiness?

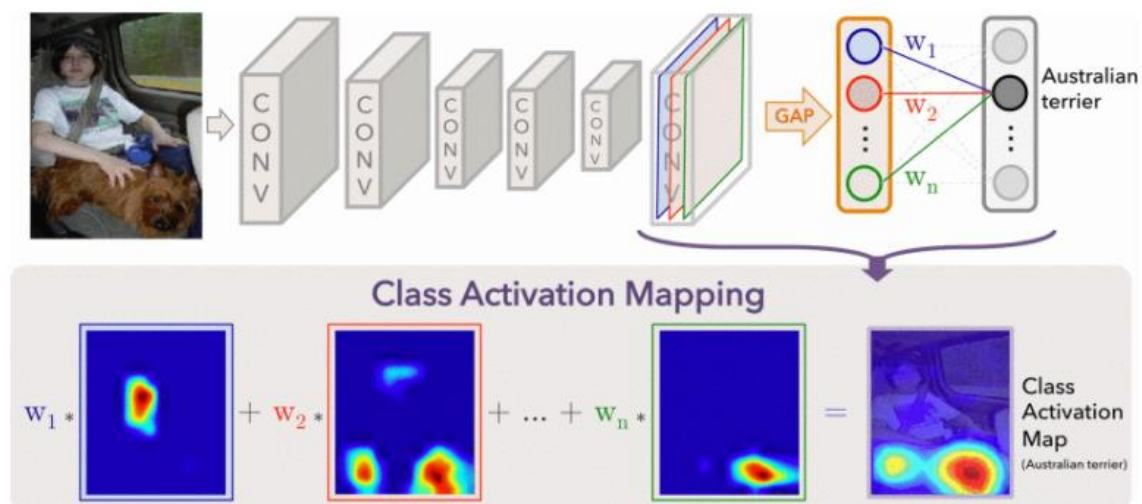
Buildings—or sky?

Figuur 15: Weergave van hoe neuronen in een model de input verwerken en hoe dit door middel van feature visualization in beeld kan worden gebracht.[88]

Een andere mogelijkheid om de medisch professional meer inzicht te geven in de black box van classificatie, zou *attribution* kunnen zijn. Attribution is een methode waarbij de belangrijkste regio's voor het maken van een beslissing op een afbeelding worden weergegeven bij een classificerend model.[89] In tegenstelling tot feature visualization zou de medisch professional op deze manier wel kunnen controleren of het model de keuze terecht heeft gemaakt.

Attribution kan op verschillende manieren toegepast worden, waarbij *Perturbation* en *Class Activation Maps* (CAM) de meest aantrekkelijke opties zijn voor het probleem. Beide methodes creëren namelijk heatmaps. Op de heatmap is niet alleen te zien welke regio belangrijk zijn voor de keuze, maar ook in welke mate zij bijdragen aan de keuze. In figuur 16 is het proces van CAM weergegeven en in de output is te zien dat het hoofd van de terriër, het rood gekleurde deel van de map, het meeste bijdraagt aan de classificatie tot *Australian Terrier*. [89,90]

Het verschil tussen perturbation en CAM is de manier waarop de heatmap wordt gegenereerd. Bij perturbation ligt *occlusion sensitivity analysis* ten grondslag aan de techniek. Hierbij wordt er steeds een deel van de oorspronkelijke afbeelding afgedekt met een grijs vierkant en wordt er vervolgens bekeken wat het effect hiervan is op de uitslag van het model. Op deze manier valideert het model waar de beslissing vandaan komt, wat vervolgens wordt weergegeven in een heatmap.[91] Bij CAM wordt de heatmap gegenereerd door een *global average pooling* layer over de laatste convolutie laag van elke feature map te halen. Van de resulterende features wordt de gewogen som berekend door middel van een fully connected layer, waaruit vervolgens de heatmap ontstaat.[89,90]



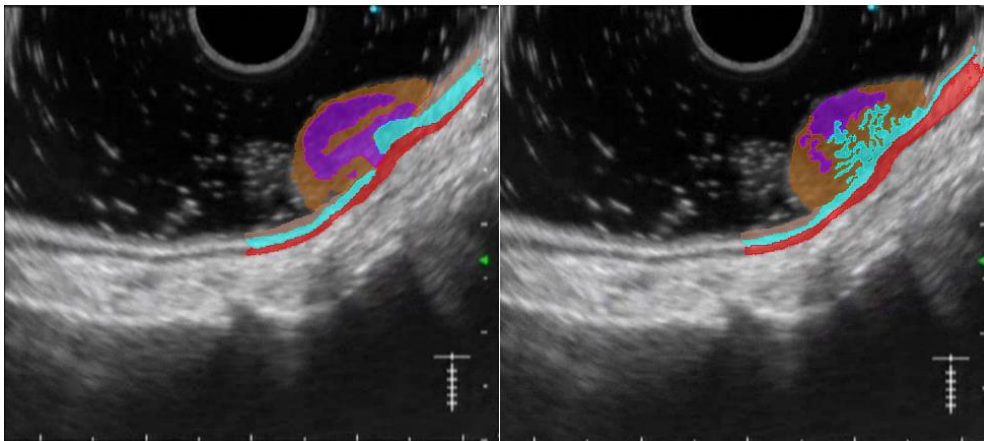
Figuur 16: Visualisatie van Class Activation Mapping.[90]

ONDERZOEK INTERBEOORDELAARSVARIABILITEIT

De uitslagen van het interbeoordelaarsvariabiliteit onderzoek zijn niet volledig reproduceerbaar. Ten eerste maakt het onderzoek gebruik van relatief weinig beelden. Wegens de tijdsduur van dit onderzoek is hiervoor gekozen, maar indien er een realistischer beeld geschetst moet worden zal het onderzoek herhaald moeten worden met meer beelden. Ten tweede bespreken de studenten de uitslag van de segmentatie bij enkele EUS-beelden. Dit kan voor een vertekend beeld van interbeoordelaarsvariabiliteit zorgen, aangezien de bespreking mogelijk wordt meegenomen in de daaropvolgende segmentaties. Ten derde is het voor een vervolgonderzoek naar de interbeoordelaarsvariabiliteit bevorderlijk om de segmentaties van de studenten te vergelijken met meerdere experts. De experts verschillen onderling ook in het maken van een segmentatie en dit kan vervolgens meegenomen worden in de interbeoordelaarsvariabiliteit van de studenten.

De kwaliteit en de consistentie van de segmentaties gemaakt door de studenten zouden verbeterd kunnen worden door meer informatie beschikbaar te stellen voor de studenten. Tijdens het onderzoek had de expert meer informatie over de RC's in de EUS-beelden, zoals de uitslag van de PA, en meer ervaring vanuit het vakgebied. Het delen van de informatie in de vorm van annotaties zou de overeenkomst tussen de segmentaties van de studenten en de expert verbeteren.

Hiernaast zal een langere training van de studenten mogelijk zorgen voor een verbeterde kwaliteit van de segmentaties. In dit onderzoek kregen de studenten een snelcursus van 30 min waarin globaal werd uitgelegd hoe het programma 3D Slicer werkt en wat de belangrijkste kenmerken zijn van de verschillende lagen. Hierbij is ook één EUS-beeld gesegmenteerd als voorbeeld. De cursus zelf stond tevens online om terug te kijken voor de studenten. Door de cursus te verlengen en meer voorbeelden te laten zien is het mogelijk om enkele standaardfouten te verhelpen. Een voorbeeld hiervan is het segmenteren van de submucosa, een van de slechtst gesegmenteerde lagen door de studenten. Een mogelijke verklaring hiervoor is dat de expert uitstulpingen van de submucosa in de poliep tekent, waar de studenten dit niet doen (figuur 17). Extra voorbeeld segmentaties en een langere cursus hadden dit kunnen voorkomen.



Figuur 17: Verschil tussen links segmentatie TG-student zonder bloedvaten uit Submucosa in poliep en rechts segmentatie expert met bloedvaten uit Submucosa in poliep.

Ook is het interessant om te kijken of de kwaliteit van de segmentaties daadwerkelijk beter wordt indien de kwaliteit van de EUS-beelden omhoog gaat (van SD- naar HD-kwaliteit). Dan zijn namelijk de lagen beter te onderscheiden.

ONDERZOEK MODELVARIABILITEIT

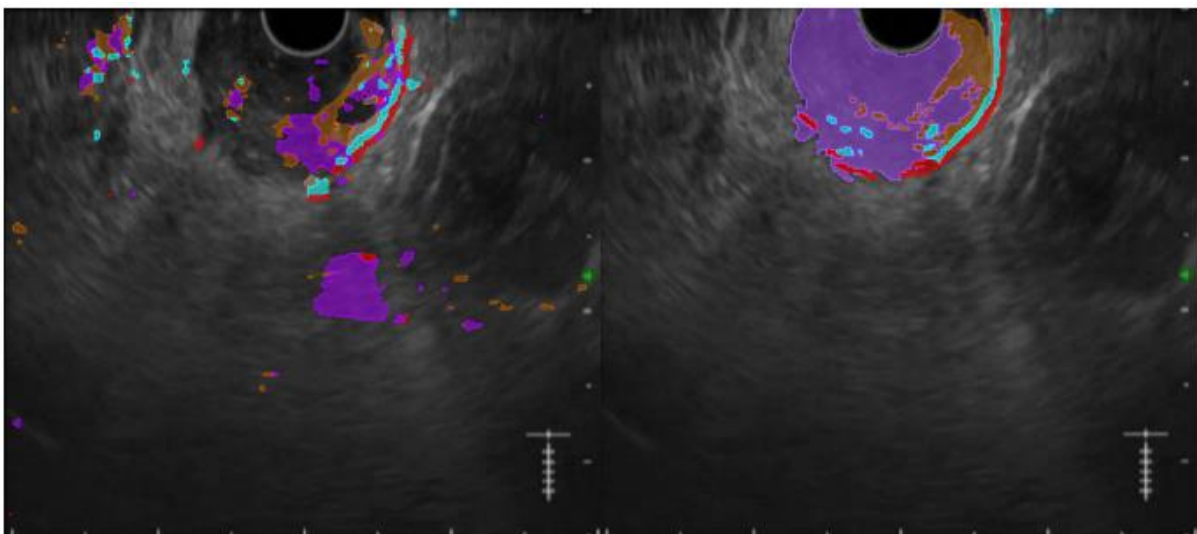
Allereerst is er verschil in resultaten van verschillende trainingen van het systeem aanwezig. Dit is te verklaren aangezien de weights bij elke training op een andere willekeurige waarde beginnen volgens de normaalverdeling. Dit zorgt ervoor dat de uitkomst van het model nooit hetzelfde is en dat de uiteindelijke best loss ook telkens verschillend is. Er bestaat dus een kans dat het model een betere best loss had kunnen bereiken.

Naast het verschil tussen de trainingen is er ook een verschil in de mate van een segmentatie binnen dezelfde training. Zo zijn er beelden waarin er wel enkele lagen zijn gesegmenteerd, zoals de *Muscularis propria*, de *Submucosa* en de *Mucosa*, maar waar het model de poliep niet herkent. Het gebeurt niet alleen bij de complexe afbeeldingen, maar ook bij sommige eenvoudige afbeeldingen. Dit is een tegenvallend beeld aangezien de poliep de meest belangrijke regio is van de afbeelding. Een mogelijke verklaring hiervoor kan zijn dat de afbeeldingen waar het model mee is getraind niet volledig representatief zijn voor de testbeelden. Zo komt het probleem bijvoorbeeld voor bij poliepen die meer links in het beeld zijn gelokaliseerd of bij beelden waar veel ruis overheen zit.

Data augmentatie, zoals besproken is In het literatuuronderzoek, kan van toegevoegde waarde zijn voor dit probleem. Door dezelfde beelden een aantal transformaties te laten ondergaan en het model hiermee te trainen, worden bijvoorbeeld ook de poliepen aan de linkerkant mogelijk beter herkend dan nu het geval is. Specifiek kan *random cropping*, een vorm van data augmentatie, van toegevoegde waarde zijn bij deze toepassing. Hierbij wordt er op een willekeurige manier een afbeelding onderverdeeld in kleine afbeeldingen, waardoor poliepen die op een andere plek of met afwijkende grootte in het EUS-beeld staan, beter worden herkend.[92]

Daarbij is het interessant om te kijken naar de gewenste accuraatheid van het systeem die bereikt dient te worden. Uit literatuur is de huidige interbeoordelaarsvariabiliteit gesteld op 61-77% en is er een breed spectrum voor de algehele accuraatheid op het gebied van T- en N-stadiëring (63%-95%).[4] Voor het systeem is het dus gewenst om hierboven te zitten. Hierbij valt te beargumenteren dat niet alle lagen deze accuraatheid hoeven te behalen. Voor het verschil in T1 en T2+ zijn de *Muscularis propria*, *Submucosa* en *Poliep (kwaadaardig)* de belangrijkste lagen. De *Mucosa* is namelijk de binnenste laag en een T1 RC zal hier altijd doorheen gaan, aangezien het tumorweefsel dikwijls uit de *Mucosa* ontstaat. De *Poliep (goedaardig)* zal daarbij nooit door de *Submucosa* gaan en draagt niet bij aan het onderscheiden van een T1 of T2+ RC.

De analyses voor de modelvariabiliteit zijn gemaakt op basis van de DSC en de AHD. Hierbij is het interessant om te bekijken wat tegenstrijdige waarden van DSC en AHD betekenen voor de segmentatie. Als voorbeeld zijn in figuur 18 twee segmentaties weergegeven van hetzelfde beeld, waarvan het eerste beeld gemaakt is door het model getraind met de beelden door de studenten en het tweede beeld handmatig is gesegmenteerd door de expert. In de resultaten is hierbij te zien dat de segmentatie van de studenten qua DSC redelijk hoog scoort. Echter, het model van de studenten laat op meerdere plekken laesies zien, waardoor deze segmentatie niet erg accuraat is. Dit is terug te zien in de hoge waarde van de AHD voor de lagen van de *Poliep (goedaardig)* en *Poliep (kwaadaardig)* bij het model van de studenten. Indien bij de vergelijking van verschillende modellen de waarden van DSC en AHD tegenstrijdig zijn, is het daarom verstandig om de gesegmenteerde beelden nader te bekijken. Hieruit kan worden bekeken wat de onderliggende oorzaak is, zodat het model hieraan kan worden aangepast. In het onderstaande beeld is er bijvoorbeeld mogelijk sprake van overfitting. Enkele aanpassingen die beschreven staan in het literatuuronderzoek zouden dit tegen kunnen gaan.



Figuur 18: Links: segmentatie van RC door het U-Net model getraind met de beelden van de TG-studenten. Rechts: handmatige segmentatie door de expert van zelfde RC.

CONCLUSIE

Voor het automatisch analyseren van EUS-beelden ter onderscheiding van T1 en T2+ rectumcarinomen is er gekozen voor deep learning model met segmentatie als toepassing. Met deze toepassing wordt voor de medisch professional de beoordeling van de EUS-beelden gevisualiseerd, waarna een stadiëring van het RC volgt. Voor het deep learning model is er gekozen voor een U-Net architectuur, vanwege veelvuldig gebruik van deze architectuur in soortgelijke medische toepassingen.

Tijdens het onderzoek naar de interbeoordelaarsvariabiliteit tussen de segmentaties van TG-studenten ten opzichte van de expert is gebleken dat er zekere verschillen in kwaliteit en consistentie van het segmenteren bestaan tussen TG-studenten. In de resultaten kwam naar voren dat de lagen van de darmwand, die essentieel zijn voor stadiëring zoals *Muscularis propria*, *Submucosa* en *Poliep (kwaadaardig)*, relatief laag scoorden qua overeenkomst met daarbij een hoge interbeoordelaarsvariabiliteit. Ook was de overeenkomst tussen stadiëringen van segmentaties tussen TG-studenten matig, wat betekent dat de overeenstemming veelal op toeval berustte.

Doorgaand op de invloed van de interbeoordelaarsvariabiliteit op de modelvariabiliteit, kan er geconcludeerd worden dat de interbeoordelaarsvariabiliteit geen tot weinig invloed heeft op de prestatie van het U-Net model. De resultaten tussen een U-net model getraind door TG-studenten ten opzichte van een expert, laten geen slechtere resultaten zien. De resultaten lijken daarentegen juist beter, maar dit verschil is niet significant.

Om de automatische beeldanalyse van EUS-beelden te verbeteren ter onderscheiding van T1 en T2+ RC's, is de aanbeveling om segmentaties van TG-studenten en het deep learning model te optimaliseren. Uit de analyse van de gemaakte segmentaties volgt de aanbeveling dat TG-studenten uitgebreider getraind dienen te worden, zodat mogelijk meer consistente en kwalitatief betere segmentaties gerealiseerd kunnen worden. Ook het combineren van segmentaties van verschillende TG-studenten zorgt voor een kleinere interbeoordelaarsvariabiliteit ten opzichte van de expert en betere overeenkomst met de expert.

Enkele aanpassingen aan het U-Net model resulteren mogelijk ook in verbeterde resultaten. Voor de training van het deep learning model zou de optimale hoeveelheid EUS-beelden met een curve van de DSC bepaald kunnen worden. Daarbij is het gebruik van high definition EUS-beelden een vereiste voor een beter resultaat. Naast het optimaliseren van de training, kan het U-Net model geoptimaliseerd worden door het tegengaan van overfitting met behulp van data augmentatie, regularisatie en stack denoising auto encoding.

Kortom, door verbeteringen van het segmenteren van EUS-beelden en het deep learning model, kan automatische beeldanalyse worden geoptimaliseerd ter onderscheiding van T1 en T2+ RC's. Met deze objectieve beoordeling worden bij de stadiëring van RC's onnodige behandelingen en overbehandeling tegengegaan. De beste keus voor de stadiëring van RC's met behulp van automatische beeldanalyse, is EUS!

REFERENTIES

- [1] Rawla P, Sunkara T, Barsouk A. Epidemiology of colorectal cancer: Incidence, mortality, survival, and risk factors. *Prz Gastroenterol* 2019;14:89–103. <https://doi.org/10.5114/pg.2018.81072>.
- [2] Masud M, Sikder N, Nahid A-A, Bairagi AK, AlZain MA. A Machine Learning Approach to Diagnosing Lung and Colon Cancer Using a Deep Learning-Based Classification Framework. *Sensors* 2021;21:748. <https://doi.org/10.3390/s21030748>.
- [3] Rosen RD, Sapra A. TNM Classification. StatPearls Publishing; 2021.
- [4] Valero M, Robles-Medrand C. Endoscopic ultrasound in oncology: An update of clinical applications in the gastrointestinal tract. *World J Gastrointest Endosc* 2017;9:243. <https://doi.org/10.4253/wjge.v9.i6.243>.
- [5] Glynne-Jones R, Wyrwicz L, Tiret E, Brown G, Rödel C, Cervantes A, et al. Rectal cancer: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up. *Ann Oncol* 2017;28:iv22–40. <https://doi.org/10.1093/annonc/mdx224>.
- [6] NHG-Standpunt Bevolkingsonderzoek darmkanker n.d. https://www.nhg.org/sites/default/files/content/nhg_org/uploads/nhg-standpunt darmkankerscreening hwdc2013_0.pdf (accessed May 3, 2021).
- [7] Bevolkingsonderzoek darmkanker n.d. <https://www.bevolkingsonderzoeknederland.nl/darmkanker/> (accessed May 3, 2021).
- [8] Startpagina - CRC - Richtlijn - Richtlijndatabase n.d. https://richtlijndatabase.nl/richtlijn/colorectaal_carcinoom_crc/startpagina_-_crc.html (accessed May 3, 2021).
- [9] Large Intestine - Anatomy and Physiology n.d. <https://www.innerbody.com/anatomy/digestive/large-intestine> (accessed May 2, 2021).
- [10] Mescher AL. Spijsverteringskanaal. In: Wisse E, Vreuls CPH, Hillebrands J-L, editors. *Junqueira's Funct. Histol.* 15th ed., Houten: Bohn Stafleu van Loghum; 2016, p. 398–431.
- [11] Binder HJ. Organization of the gastrointestinal system. *Med. Physiol.* 3rd ed., Philadelphia: Elsevier; 2017, p. 11.
- [12] Guerin L, Mitros F. The Gastrointestinal tract. In: Strayer DS, Rubin E, Saffitz JE, Sciler AL, editors. *Rubin's Pathol. Clin. Found. Med.*, Lippincott Williams and Wilkins; 2015, p. 810–8.
- [13] Rudy DR, Zdon MJ. Update on Colorectal Cancer. *Am Fam Physician* 2000;61:1759–70.
- [14] ter Borg S, Neefjes A, van Boven H, van Marion A. Uw pathologie rapport beter begrijpen 2013. https://pathology.nl/wp-content/uploads/PDF/publicaties/voorlichting/nvvp1304FAQ_colon_adenoom.pdf (accessed May 3, 2021).
- [15] They found colon polyps: Now what? - Harvard Health n.d. <https://www.health.harvard.edu/diseases-and-conditions/they-found-colon-polyps-now-what> (accessed May 3, 2021).
- [16] Colon cancer - Symptoms and causes - Mayo Clinic n.d. <https://www.mayoclinic.org/diseases-conditions/colon-cancer/symptoms-causes/syc-20353669> (accessed May 3, 2021).
- [17] Ang TL, Kwek ABE, Wang LM. Diagnostic endoscopic ultrasound: Technique, current status and future directions. *Gut Liver* 2018;12:483–96. <https://doi.org/10.5009/gnl17348>.
- [18] Shaw SM, Kimmey MB. General principles of endoscopic ultrasonographic imaging. *Tech Gastrointest Endosc* 2000;2:50–5. <https://doi.org/10.1053/TG.2000.5430>.
- [19] O'Brien WD. Ultrasound-biophysics mechanisms. *Prog Biophys Mol Biol* 2007;93:212–55. <https://doi.org/10.1016/j.pbiomolbio.2006.07.010>.

- [20] Wells PNT. Absorption and dispersion of ultrasound in biological tissue. *Ultrasound Med Biol* 1975;1:369–76. [https://doi.org/10.1016/0301-5629\(75\)90124-6](https://doi.org/10.1016/0301-5629(75)90124-6).
- [21] Siddiqui AA, Fayiga Y, Huerta S. The role of endoscopic ultrasound in the evaluation of rectal cancer. *Int Semin Surg Oncol* 2006;3:36. <https://doi.org/10.1186/1477-7800-3-36>.
- [22] Brugge WR. Endoscopic ultrasonography: The current status. *Gastroenterology* 1998;115:1577–83. [https://doi.org/10.1016/S0016-5085\(98\)70039-3](https://doi.org/10.1016/S0016-5085(98)70039-3).
- [23] Ter Borg F. Persoonlijke communicatie 2021.
- [24] Razzak MI, Naz S, Zaib A. Deep learning for medical image processing: Overview, challenges and the future. *Lect. Notes Comput. Vis. Biomech.*, vol. 26, Springer Netherlands; 2018, p. 323–50. https://doi.org/10.1007/978-3-319-65981-7_12.
- [25] Ghosh A, Sufian A, Sultana F, Chakrabarti A, De D. Fundamental concepts of convolutional neural network. *Intell. Syst. Ref. Libr.*, vol. 172, Springer; 2019, p. 519–67. https://doi.org/10.1007/978-3-030-32644-9_36.
- [26] Sahiner B, Pezeshk A, Hadjiiski LM, Wang X, Drukker K, Cha KH, et al. Deep learning in medical imaging and radiation therapy. *Med Phys* 2019;46:e1–36. <https://doi.org/10.1002/mp.13264>.
- [27] Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 2018;9:611–29. <https://doi.org/10.1007/s13244-018-0639-9>.
- [28] Pacal I, Karaboga D, Basturk A, Akay B, Nalbantoglu U. A comprehensive review of deep learning in colon cancer. *Comput Biol Med* 2020;126:104003. <https://doi.org/10.1016/j.combiomed.2020.104003>.
- [29] Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK. Medical Image Analysis using Convolutional Neural Networks: A Review. *J Med Syst* 2018;42:226. <https://doi.org/10.1007/s10916-018-1088-1>.
- [30] Sarvamangala DR, Kulkarni R V. Convolutional neural networks in medical image understanding: a survey. *Evol Intell* 2021;1:3. <https://doi.org/10.1007/s12065-020-00540-3>.
- [31] Hussain S, Anwar SM, Majid M. Segmentation of glioma tumors in brain using deep convolutional neural network. *Neurocomputing* 2018;282:248–61. <https://doi.org/10.1016/j.neucom.2017.12.032>.
- [32] Yu H, Yang LT, Zhang Q, Armstrong D, Deen MJ. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing* 2021;444:92–110. <https://doi.org/10.1016/j.neucom.2020.04.157>.
- [33] Oh KS, Jung K. GPU implementation of neural networks. *Pattern Recognit* 2004;37:1311–4. <https://doi.org/10.1016/j.patcog.2004.01.013>.
- [34] Enyinna Nwankpa C, Ijomah W, Gachagan A, Marshall S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. n.d.
- [35] Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax | by Mukesh Chaudhary | Medium n.d. <https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5> (accessed June 19, 2021).
- [36] da Costa RF, Hayashi Yelisetty SM, Marques JC, Tasinaffo PM. A Brief Didactic Theoretical Review on Convolutional Neural Networks, Deep Belief Networks and Stacked Auto-Encoders. *Int J Eng Tech Res* 2019;9. <https://doi.org/10.31873/ijetr.9.12.35>.
- [37] Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436–44. <https://doi.org/10.1038/nature14539>.
- [38] Takase T, Oyama S, Kurihara M. Effective neural network training with adaptive learning rate based on training loss. *Neural Networks* 2018;101:68–78. <https://doi.org/10.1016/j.neunet.2018.01.016>.

- [39] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. *Med Image Anal* 2017;42:60–88. <https://doi.org/10.1016/j.media.2017.07.005>.
- [40] Lai Z, Deng H. Medical image classification based on deep features extracted by deep model and statistic feature fusion with multilayer perceptron. *Comput Intell Neurosci* 2018;2018. <https://doi.org/10.1155/2018/2061516>.
- [41] Nguyen Q, Lee SW. Colorectal Segmentation Using Multiple Encoder-Decoder Network in Colonoscopy Images. *Proc. - 2018 1st IEEE Int. Conf. Artif. Intell. Knowl. Eng. AIKE 2018*, Institute of Electrical and Electronics Engineers Inc.; 2018, p. 208–11. <https://doi.org/10.1109/AIKE.2018.00048>.
- [42] Ye JC, Sung WK. Understanding Geometry of Encoder-Decoder CNNs. *36th Int Conf Mach Learn ICML 2019* 2019;2019-June:12245–54.
- [43] Wu Z, Shen C, van den Hengel A. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Pattern Recognit* 2019;90:119–33. <https://doi.org/10.1016/j.patcog.2019.01.006>.
- [44] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016- December, IEEE Computer Society; 2016, p. 770–8. <https://doi.org/10.1109/CVPR.2016.90>.
- [45] Gong T, Niu H. An implementation of resnet on the classification of rgb-d images. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12093 LNCS, Springer; 2020, p. 149–55. https://doi.org/10.1007/978-3-030-49556-5_15.
- [46] Zhu Y, Wang QC, Xu MD, Zhang Z, Cheng J, Zhong YS, et al. Application of convolutional neural network in the diagnosis of the invasion depth of gastric cancer based on conventional endoscopy. *Gastrointest Endosc* 2019;89:806-815.e1. <https://doi.org/10.1016/j.gie.2018.11.011>.
- [47] Lu Z, Bai Y, Chen Y, Su C, Lu S, Zhan T, et al. The classification of gliomas based on a Pyramid dilated convolution resnet model. *Pattern Recognit Lett* 2020;133:173–9. <https://doi.org/10.1016/j.patrec.2020.03.007>.
- [48] Wang W, Yang Y. Development of convolutional neural network and its application in image classification: a survey. *Opt Eng* 2019;58:1. <https://doi.org/10.1117/1.OE.58.4.040901>.
- [49] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., International Conference on Learning Representations, ICLR*; 2015.
- [50] Muhammad U, Wang W, Chattha SP, Ali S. Pre-trained VGGNet Architecture for Remote-Sensing Image Scene Classification. *Proc. - Int. Conf. Pattern Recognit.*, vol. 2018- August, Institute of Electrical and Electronics Engineers Inc.; 2018, p. 1622–7. <https://doi.org/10.1109/ICPR.2018.8545591>.
- [51] Jun H, Shuai L, Jinming S, Yue L, Jingwei W, Peng J. Facial Expression Recognition Based on VGGNet Convolutional Neural Network. *Proc. 2018 Chinese Autom. Congr. CAC 2018*, Institute of Electrical and Electronics Engineers Inc.; 2019, p. 4146–51. <https://doi.org/10.1109/CAC.2018.8623238>.
- [52] Zhiqi Y. Face recognition based on improved VGGNET convolutional neural network. *IAEAC 2021 - IEEE 5th Adv. Inf. Technol. Electron. Autom. Control Conf.*, Institute of Electrical and Electronics Engineers Inc.; 2021, p. 2530–3. <https://doi.org/10.1109/IAEAC50856.2021.9390856>.
- [53] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going Deeper with Convolutions. n.d.
- [54] Guo Z, Chen Q, Wu G, Xu Y, Shibasaki R, Shao X. Village building identification based on Ensemble Convolutional Neural Networks. *Sensors (Switzerland)* 2017;17. <https://doi.org/10.3390/s17112487>.

- [55] Alom MZ, Yakopcic C, Taha TM, Asari VK. Microscopic Blood Cell Classification Using Inception Recurrent Residual Convolutional Neural Networks. *Proc. IEEE Natl. Aerosp. Electron. Conf. NAECON*, vol. 2018- July, Institute of Electrical and Electronics Engineers Inc.; 2018, p. 222–7. <https://doi.org/10.1109/NAECON.2018.8556737>.
- [56] Alom MZ, Yakopcic C, Nasrin MS, Taha TM, Asari VK. Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. *J Digit Imaging* 2019;32:605–17. <https://doi.org/10.1007/s10278-019-00182-7>.
- [57] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017;39:2481–95. <https://doi.org/10.1109/TPAMI.2016.2644615>.
- [58] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12- June-2015, IEEE Computer Society; 2015, p. 431–40. <https://doi.org/10.1109/CVPR.2015.7298965>.
- [59] Liu X, Song L, Liu S, Zhang Y. A review of deep-learning-based medical image segmentation methods. *Sustain* 2021;13:1–29. <https://doi.org/10.3390/su13031224>.
- [60] Li B, Zhang T, Xia T. Vehicle Detection from 3D Lidar Using Fully Convolutional Network. *Robot Sci Syst* 2016;12.
- [61] Ben-Cohen A, Diamant I, Klang E, Amitai M, Greenspan H. Fully convolutional network for liver segmentation and lesions detection. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10008 LNCS, Springer Verlag; 2016, p. 77–85. https://doi.org/10.1007/978-3-319-46976-8_9.
- [62] Siddique N, Sidike P, Elkin C, Devabhaktuni V. U-Net and its variants for medical image segmentation: theory and applications. *IEEE Access* 2020;9:82031–57. <https://doi.org/10.1109/ACCESS.2021.3086020>.
- [63] Ibtehaz N, Rahman MS. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Networks* 2020;121:74–87. <https://doi.org/10.1016/j.neunet.2019.08.025>.
- [64] Beulah AS, Kaul K, Chauhan D. BRAIN MRI ANALYSIS AND SEGMENTATION USING 2D-UNET ARCHITECTURE. *Eur J Mol Clin Med* n.d.;08:2021.
- [65] Jin Q, Meng Z, Sun C, Cui H, Su R. RA-UNet: A Hybrid Deep Attention-Aware Network to Extract Liver and Tumor in CT Scans. *Front Bioeng Biotechnol* 2020;8:605132. <https://doi.org/10.3389/fbioe.2020.605132>.
- [66] Cui S, Chen M, Liu C. DsUnet: A New Network Structure for Detection and Segmentation of Ultrasound Breast Lesions. *J Med Imaging Heal Informatics* 2019;10:661–6. <https://doi.org/10.1166/jmihi.2020.2914>.
- [67] Luo C, Li X, Wang L, He J, Li D, Zhou J. How Does the Data set Affect CNN-based Image Classification Performance? 2018 5th Int. Conf. Syst. Informatics, ICSAI 2018, Institute of Electrical and Electronics Engineers Inc.; 2019, p. 361–6. <https://doi.org/10.1109/ICSAI.2018.8599448>.
- [68] Cho J, Lee K, Shin E, Choy G, Do S. HOW MUCH DATA IS NEEDED TO TRAIN A MEDICAL IMAGE DEEP LEARNING SYSTEM TO ACHIEVE NECESSARY HIGH ACCURACY? 2016.
- [69] Poggio T, Kawaguchi K, Liao Q, Miranda B, Rosasco L, Boix X, et al. Theory of Deep Learning III: explaining the non-overfitting puzzle. 2018.
- [70] You K, Long M, Wang J, Jordan MI. How Does Learning Rate Decay Help Modern Neural Networks? 2019.
- [71] Bardis M, Houshyar R, Chantaduly C, Ushinsky A, Glavis-Bloom J, Shaver M, et al. Deep learning with limited data: Organ segmentation performance by U-net. *Electron* 2020;9:1–12. <https://doi.org/10.3390/electronics9081199>.
- [72] Fujisawa Y, Otomo Y, Ogata Y, Nakamura Y, Fujita R, Ishitsuka Y, et al. Deep-learning-based,

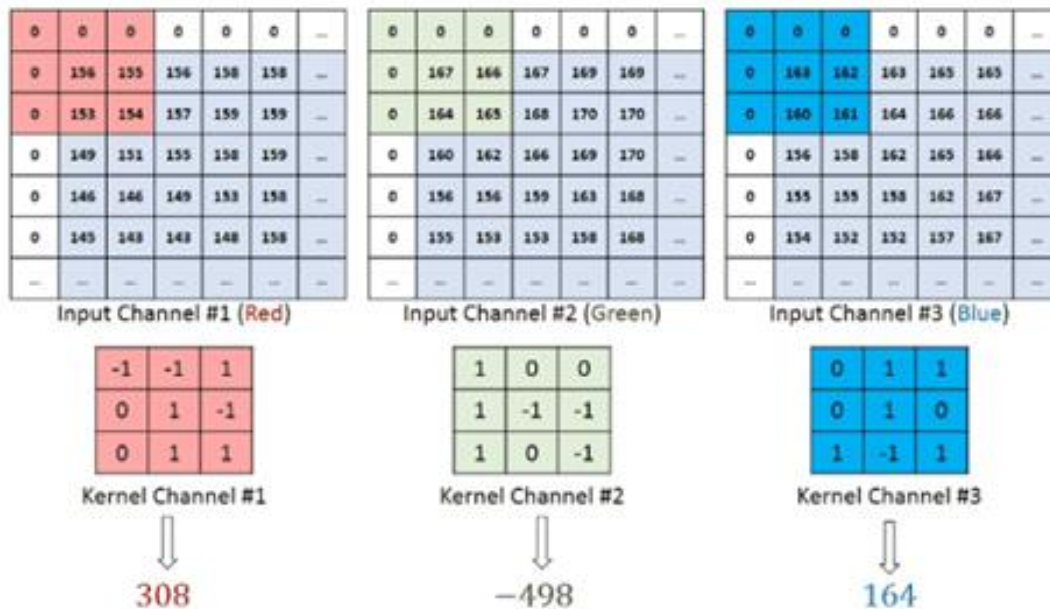
- computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. *Br J Dermatol* 2019;180:373–81. <https://doi.org/10.1111/bjd.16924>.
- [73] Santos MS, Soares JP, Abreu PH, Araujo H, Santos J. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [Research Frontier]. *IEEE Comput Intell Mag* 2018;13:59–76. <https://doi.org/10.1109/MCI.2018.2866730>.
 - [74] Piotr Skalski. Preventing Deep Neural Network from Overfitting | by Piotr Skalski | Towards Data Science n.d. <https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a> (accessed June 18, 2021).
 - [75] Raheja S, Chopra R. Training data on recursive parallel processors for deep learning. *Lect. Notes Electr. Eng.*, vol. 612, Springer; 2020, p. 63–71. https://doi.org/10.1007/978-981-15-0372-6_5.
 - [76] Ng AY. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. n.d.
 - [77] Liang J, Liu R. Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. *Proc. - 2015 8th Int. Congr. Image Signal Process. CISP 2015, Institute of Electrical and Electronics Engineers Inc.*; 2016, p. 697–701. <https://doi.org/10.1109/CISP.2015.7407967>.
 - [78] Liang P, Shi W, Zhang X. Remote sensing image classification based on Stacked Denoising Autoencoder. *Remote Sens* 2018;10:16. <https://doi.org/10.3390/rs10010016>.
 - [79] Liu P, Zheng P, Chen Z. Deep learning with stacked denoising auto-encoder for short-term electric load forecasting. *Energies* 2019;12:2445. <https://doi.org/10.3390/en12122445>.
 - [80] Sailer M, Allgayer H, Dietrich CF. 24 Endoanal and Endorectal Sonography. In: Dietrich CF, editor. *Endosc. Ultrasound*. 2006th ed., Stuttgart: Georg Thieme Verlag; 2006. <https://doi.org/10.1055/b-0034-52454>.
 - [81] Nijboer TS, Gijsbers KM, Wolterink JM, Ter Borg F. Segmentation of Bowel Wall Layers in Endoscopic Ultrasound Using a Machine Learning Model in Patients with T1 or T2 Colorectal Cancer. 2020.
 - [82] Yang X, Wang H, Dong Q, Xu Y, Liu H, Ma X, et al. An Artificial Intelligence System for Distinguishing Between Gastrointestinal Stromal Tumors and Leiomyomas Using Endoscopic Ultrasonography (with video). *Endoscopy* 2021. <https://doi.org/10.1055/a-1476-8931>.
 - [83] Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med Imaging* 2015;15:29. <https://doi.org/10.1186/s12880-015-0068-x>.
 - [84] Zou KH, Warfield SK, Bharatha A, Tempany CMC, Kaus MR, Haker SJ, et al. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index. *Acad Radiol* 2004;11:178–89. [https://doi.org/10.1016/S1076-6332\(03\)00671-8](https://doi.org/10.1016/S1076-6332(03)00671-8).
 - [85] Aydin OU, Taha AA, Hilbert A, Khalil AA, Galinovic I, Fiebach JB, et al. On the usage of average Hausdorff distance for segmentation performance assessment: hidden error when used for ranking. *Eur Radiol Exp* 2021;5:1–7. <https://doi.org/10.1186/s41747-020-00200-2>.
 - [86] McHugh ML. Interrater reliability: The kappa statistic. *Biochem Medica* 2012;22:276–82. <https://doi.org/10.11613/bm.2012.031>.
 - [87] Landis JR, Koch GG. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 1977;33:159. <https://doi.org/10.2307/2529310>.
 - [88] Olah C, Mordvintsev A, Schubert L. Feature Visualization. *Distill* 2017;2:e7. <https://doi.org/10.23915/distill.00007>.
 - [89] Buhrmester V, Münch D, Arens M. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. n.d.
 - [90] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-

- December, IEEE Computer Society; 2016, p. 2921–9. <https://doi.org/10.1109/CVPR.2016.319>.
- [91] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8689 LNCS, Springer Verlag; 2014, p. 818–33. https://doi.org/10.1007/978-3-319-10590-1_53.
 - [92] Takahashi R, Matsubara T, Uehara K. Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. IEEE Trans Circuits Syst Video Technol 2020;30:2917–31. <https://doi.org/10.1109/TCSVT.2019.2935128>.
 - [93] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science n.d. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (accessed June 18, 2021).
 - [94] Ghosh A, Sufian A, Sultana F, Chakrabarti A, De D. Fundamental concepts of convolutional neural network. Intell. Syst. Ref. Libr., vol. 172, Springer; 2019, p. 519–67. https://doi.org/10.1007/978-3-030-32644-9_36.
 - [95] A Comprehensive Introduction to Different Types of Convolutions in Deep Learning | by Kunlun Bai | Towards Data Science n.d. <https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215> (accessed June 18, 2021).
 - [96] Christopher J, Messou E. Handling Invalid Pixels in Convolutional Neural Networks. 2020.
 - [97] Choi H, Ryu S, Kim H. Short-Term Load Forecasting based on ResNet and LSTM. 2018 IEEE Int. Conf. Commun. Control. Comput. Technol. Smart Grids, SmartGridComm 2018, Institute of Electrical and Electronics Engineers Inc.; 2018. <https://doi.org/10.1109/SmartGridComm.2018.8587554>.
 - [98] Kataoka H, Iwata K, Satoh Y. Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection 2015.
 - [99] Khvostikov A, Aderghal K, Krylov A, Catheline G, Benois-Pineau J. 3D Inception-based CNN with sMRI and MD-DTI data fusion for Alzheimer’s Disease diagnostics 2018.
 - [100] Semantic Segmentation using Fully Convolutional Neural Networks | by Wilbur de Souza | Medium n.d. <https://medium.com/@wilburdes/semantic-segmentation-using-fully-convolutional-neural-networks-86e45336f99b> (accessed June 18, 2021).
 - [101] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans Pattern Anal Mach Intell 2017;39:2481–95. <https://doi.org/10.1109/TPAMI.2016.2644615>.

BIJLAGEN

BIJLAGE A: CONVOLUTIE

In de convolutional layer van een CNN wordt de data uit een afbeelding verwerkt door middel van een wiskundige operatie genaamd het dot-product. In deze bijlage zal de convolutie met wiskundige onderbouwing uitgelegd worden.



Figuur 1: Convolutie van een matrix met pixelwaarden met bijbehorende drie kernels voor de RGB-kleuren. [93]

In de figuur 1 is een convolutie van een afbeelding, oftewel de matrix met pixelwaarden,[27] met verschillende kernels te zien. Er zijn drie verschillende kernels, voor de drie RGB-kleuren (rood, blauw en groen). Deze kernels zijn matrices met waarden van -1, 0 en 1, waarmee een dot-product met het betreffende deel van de afbeelding wordt uitgevoerd. In de bovenstaande figuur wordt er een convolutie uitgevoerd van de rode kernel met het rode gebied van de afbeelding.

0 x -1	0 x -1	0 x 1
0 x 0	156 x 1	155 x -1
0 x 0	153 x 1	154 x 1

$0 + 0 + 0 + 0 + 156 - 155 + 0 + 153 + 154 = 308$

Figuur 2: Berekening van het dot-product bij een convolutie.

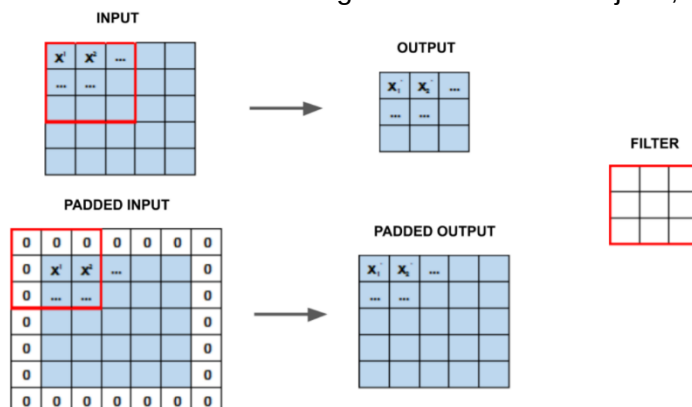
Het dot-product dat vervolgens berekend wordt, is aangegeven in figuur 2. Hierbij worden alle elementen uit het betreffende gebied van de afbeelding met de corresponderende elementen in de rode kernel vermenigvuldigd. Daarna worden alle uitkomsten van de vermenigvuldigingen bij elkaar opgeteld. De convolutie van het betreffende gebied van de afbeelding met de rode kernel levert dus een waarde van 308 op. Vervolgens worden ook de convoluties van dat deel van de afbeelding met de groene en de blauwe kernel uitgevoerd, in het geval van een RGB-afbeelding. In het geval van een zwart-wit afbeelding volstaat ook een enkele kernel, omdat de afbeelding dan niet uit drie lagen (rood, groen en blauw) is opgebouwd, maar slechts uit één laag bestaande uit pixelwaarden van grijstinten.

Na de dot-producten van de rode, groene en blauwe kernel met het deel van de afbeelding in figuur 1, ontstaat er drie verschillende matrices. Door de kernels telkens met een bepaalde stapgrootte oftewel het aantal pixels te verschuiven, kan uiteindelijk met de hele afbeelding een convolutie uitgevoerd worden. Deze stapgrootte, waarmee de kernel telkens verschoven wordt, is de *stride length*. [94] Als output levert dit een nieuwe matrix op, waarin de waarden van de pixels een representatie vormen van pixels in de originele afbeelding. Op deze manier wordt de matrix steeds verder verkleind, waardoor ook de hoeveelheid te verwerken data door het CNN kleiner wordt.



Figuur 19: RGB-plaatje met daarnaast het resultaat van de convolutie per kleur. [95]

Door telkens verschillende kernels te gebruiken, kunnen ook telkens andere eigenschappen van de pixels in een deel van de afbeelding worden gedetecteerd. Dit is in bijvoorbeeld figuur 3 te zien, waarin in het eerste plaatje een afbeelding in kleur is gegeven. Vervolgens zijn in de resterende drie plaatjes de outputs gegeven van de convoluties van de afbeelding met verschillende kernels, namelijk voor de kleuren rood, groen en blauw. Het is te zien dat door verschillende kernels, ook verschillende eigenschappen van de afbeelding worden uitgelicht. Zo kunnen er ook kernels gebruikt worden om lijnen, vormen of structuren uit te lichten.



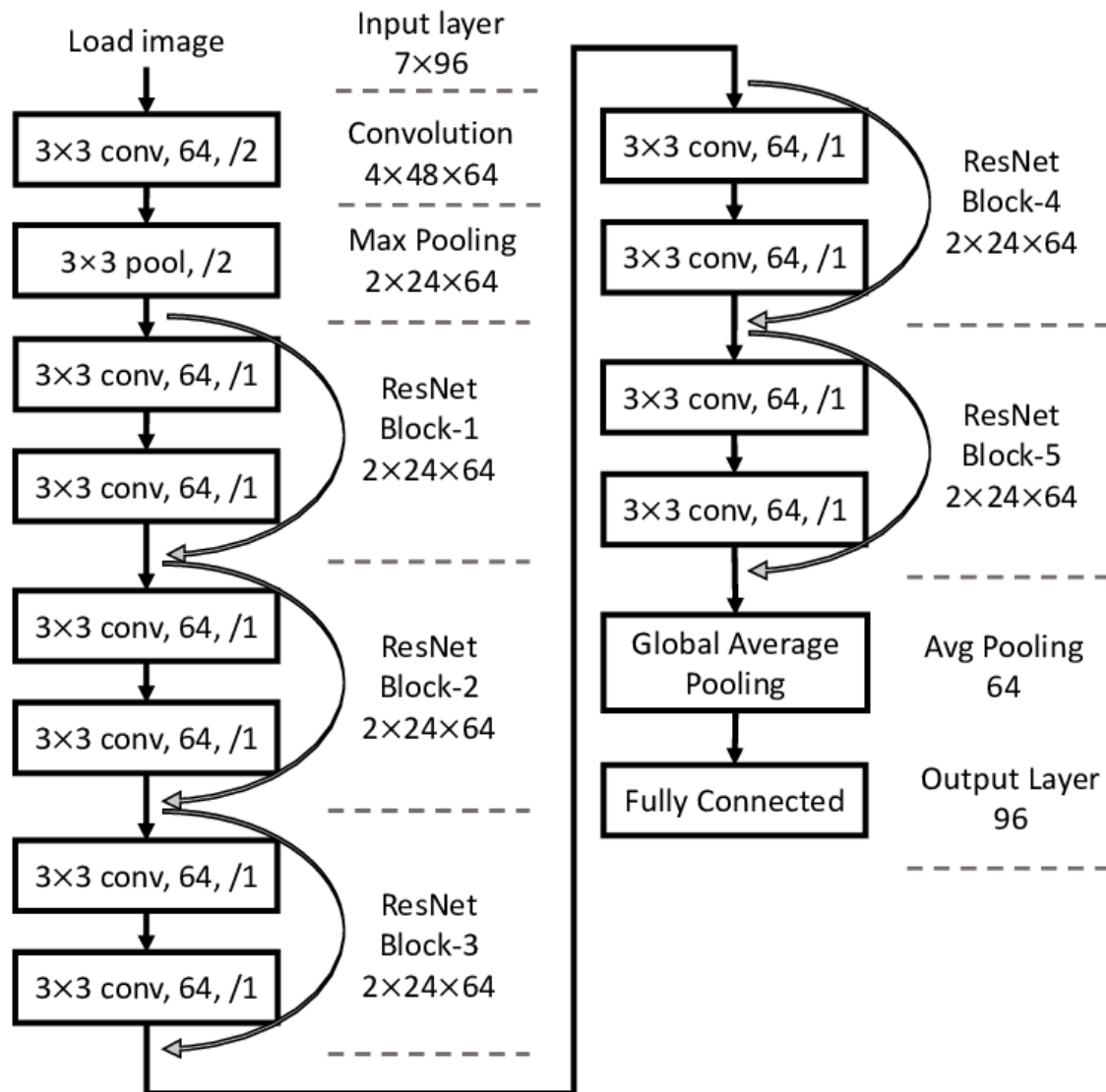
Figuur 20: Het toepassen van padding om de output gelijk te houden aan de input. [96]

Bij een grotere stride length wordt de kernel met grotere stappen over de afbeelding verschoven. Dit heeft als voordeel dat de CNN de afbeelding sneller verwerkt maar als nadeel dat de pixels minder gedetailleerd verwerkt worden. Daarnaast zal een grotere kernel zorgen voor een kleinere matrix. Als een kernel de randen van de afbeelding niet volledig overlapt, kan er geen dot-product berekend worden op dat punt. Om de grootte van de matrix gelijk te houden aan de oorspronkelijke afbeelding, kan *padding* toegepast worden, zie figuur 4. Hierbij worden één of meerdere rijen en kolommen gevuld met nullen rondom de afbeelding toegevoegd, waarna de convolutie wel gemaakt kan worden. Op deze manier gaat de data aan de randen niet verloren. [27,94] De grootte van padding kan berekend worden met de volgende formule:

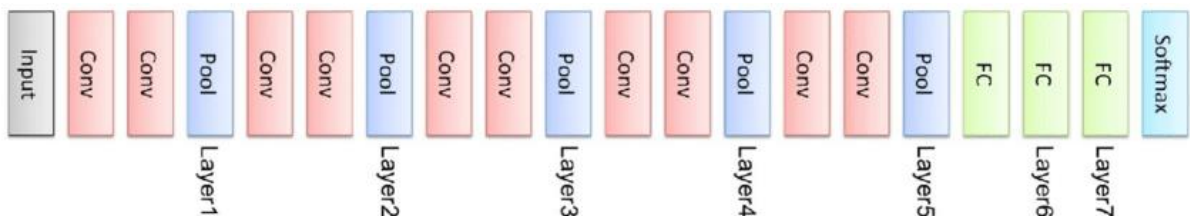
$$\frac{1}{2}(((Output\ matrix\ size - 1) \times Stride\ length) + Kernel\ size - Input\ size)$$

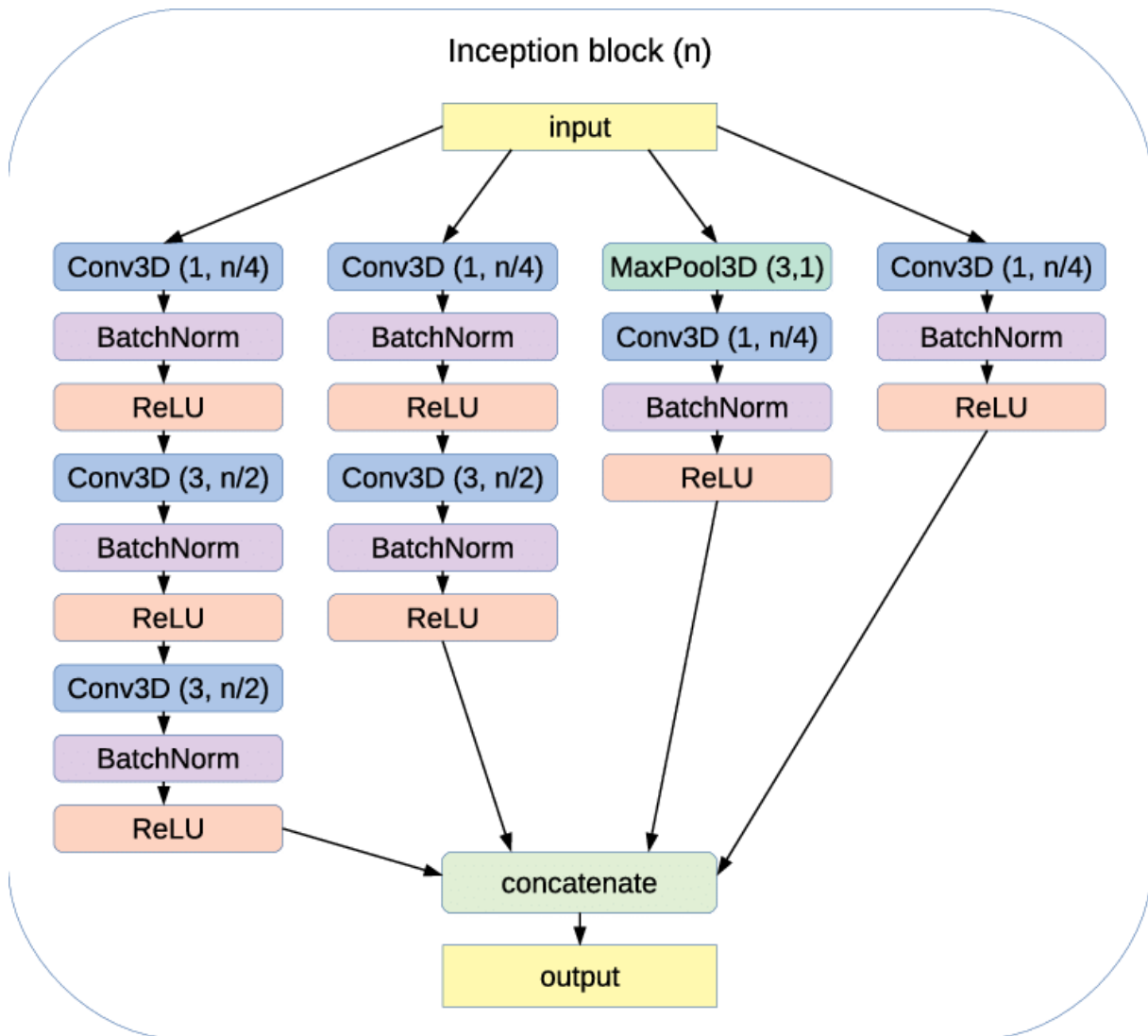
BIJLAGE B: ARCHITECTUREN TOEPASSING CLASSIFICATIE

RESNET [97]



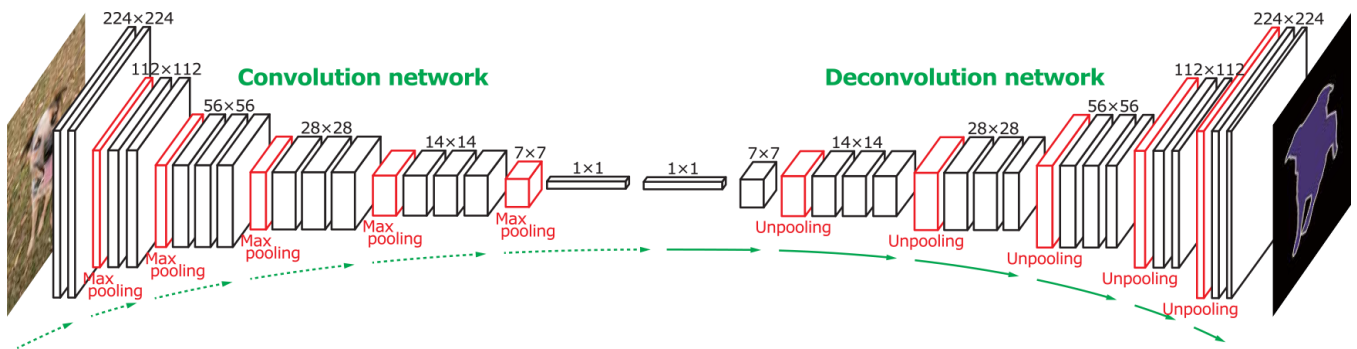
VGGNET [98]



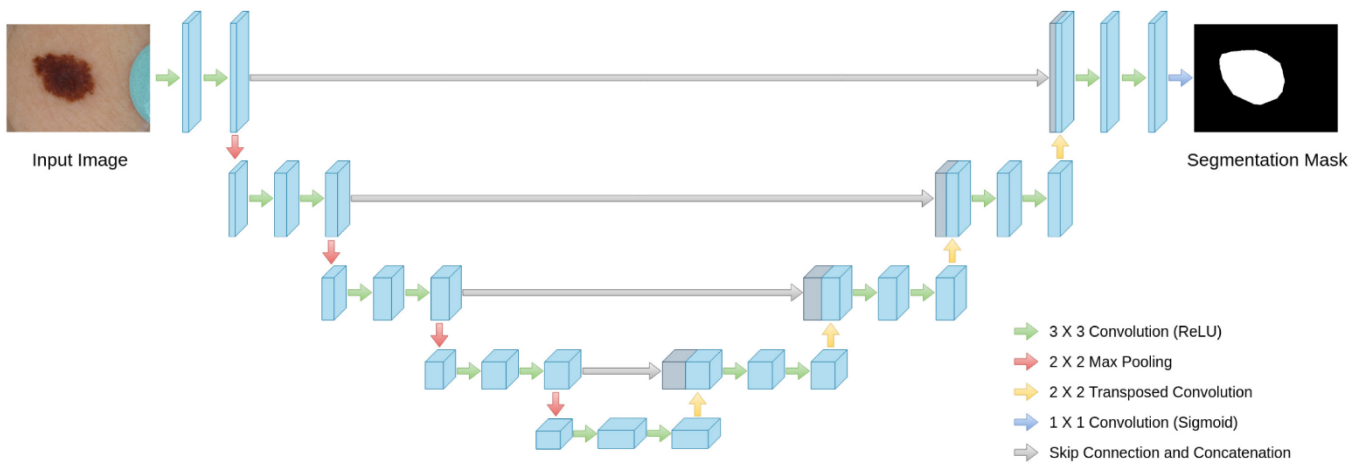


BIJLAGE C: ARCHITECTUREN TOEPASSING SEGMENTATIE

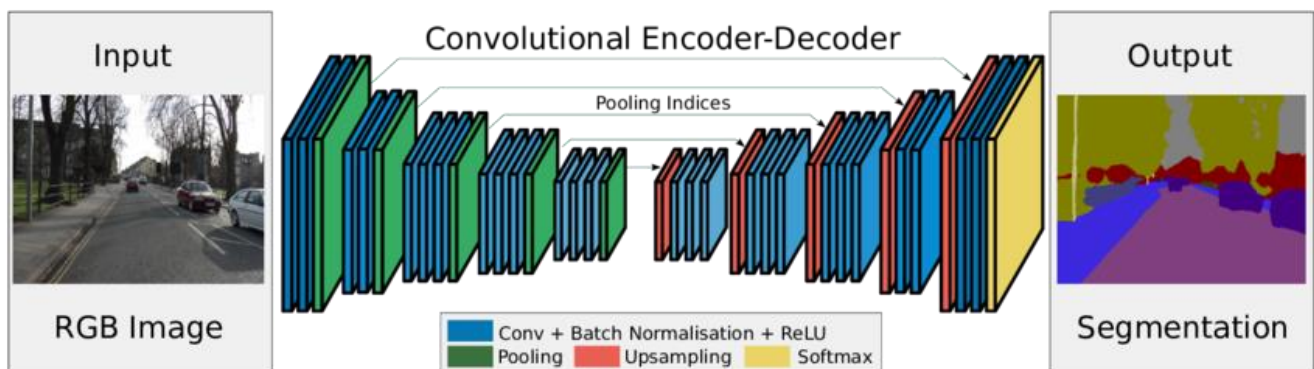
FULLY CONVOLUTIONAL NETWORK [100]



U-NET [63]



SEGET [57]



BIJLAGE D: BEREKENING DSC EN AHD IN PYTHON

Importeren

```
[ ]: !pip install simpleitk
!pip install PySimpleGUI
!pip install pynrrd
!pip install seg_metrics
!pip install monai
!pip install scipy

[ ]: import pandas as pd
import matplotlib.pyplot as plt

from PIL import Image .

import torch
import torch.nn.functional as F
import torchvision

import torchvision.transforms as transforms
from torchvision import datasets
from torch.utils.data import DataLoader, Dataset
from sklearn.model_selection import train_test_split
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim

import numpy as np
import math
from torch.utils.data.sampler import SubsetRandomSampler

from torch.utils.tensorboard import SummaryWriter
import os
from glob import glob
import imageio

import monai
from monai.config import print_config
```

```

from monai.data import ArrayDataset, Dataset
from monai.inferers import sliding_window_inference
from monai.metrics import DiceMetric, compute_meandice
from monai.transforms import(
    AddChannel,
    Compose,
    RandRotate90,
    RandSpatialCrop,
    ScaleIntensity,
    ToTensor,
    RandAdjustContrast,
    SqueezeDim,
    Transpose,
    Flip,
    SplitChannel,
    Rand2DElastic
)
from monai import transforms
from monai.losses import DiceLoss
from monai.visualize import plot_2d_or_3d_image
from monai.networks.nets import UNet, DenseNet
from monai.utils import first, set_determinism
from typing import Optional, Callable, Sequence, Tuple, TypeVar, Union
import time
import statistics

import os
import SimpleITK as sitk
import seg_metrics.seg_metrics as sg
import imageio
from monai.utils import first
from skimage import metrics
import nrrd
import math
import scipy

```

Inladen segmentaties

```

[ ]: from google.colab import drive
drive.mount('/content/drive')

```

Dataset creëren

```

[ ]: class Label_dataset(Dataset):
    def __init__(self, prednames, gtnames,
        ↪ classes=6, transform_seg=None, transform_seg2=None):
        self.files_seg2 = prednames
        self.files_seg = gtnames
        self.classes = classes
        self.transform_seg = transform_seg

```

```

self.transform_seg2 = transform_seg2

def __getitem__ (self,index):

    seg_img = np.array(imageio.imread(self.files_seg[index]))

    seg_img = np.squeeze(seg_img)
    if seg_img.shape[0] == 2:
        seg_img = np.squeeze(seg_img[1, :, : ])
    if seg_img.shape[0] == 3:
        seg_img = np.squeeze(seg_img[1, :, : ])

    seg = np.zeros(shape=np.shape(seg_img) + (self.classes,))

    seg2_img = np.array(imageio.imread(self.files_seg2[index]))
    seg2_img = np.squeeze(seg2_img)

    if seg2_img.shape[0] == 2:
        seg2_img = np.squeeze(seg2_img[1, :, : ])
    if seg2_img.shape[0] == 3:
        seg2_img = np.squeeze(seg2_img[1, :, : ])
    seg2 = np.zeros(shape=np.shape(seg2_img) + (self.classes,))

    for i in range(6):
        ind=np.nonzero(seg_img==i)
        ind2=np.nonzero(seg2_img==i)
        seg[ind[0],ind[1],i]=1
        seg2[ind2[0],ind2[1],i]=1
    return seg,seg2

def __len__(self):
    return len(self.files_seg)

```

```

[:]: direct = '/content/drive/MyDrive/TGO/Systeem/'

gtnames = sorted(glob(os.path.join(direct,"GTFrank/Segmentation*.seg.nrrd")))
prednames = sorted(glob(os.path.join(direct,"EvaluatieStudenten/
↳Evaluatie_training_126_Jesse/test*.nrrd")))
print(prednames)

```

```

[:]: train_imtrans = Compose(
    [
        Transpose([2,0,1])
    ]
)

train_segtrans = Compose(

```

```
[
    Transpose([2,0,1])
]
```

Controle

```
[ ]: print(len(gtnames))
      print(len(prednames))
      print(gtnames)
      print(prednames)
```

Dice Similarity Coefficient

```
[ ]: dataset_seg = Label_dataset(prednames, gtnames, 6, train_segtrans,
    ↪ train_intrans)
seg_loader= DataLoader(dataset_seg, batch_size=1, shuffle=False, num_workers=2)
i=0
for seg, seg2 in seg_loader:

    seg = seg.numpy()
    seg2 = seg2.numpy()
    seg=np.squeeze(seg)
    seg2=np.squeeze(seg2)
    print(prednames[i])
    i=i+1
    for k in range(6):
        dice = np.sum((seg2[:, :, k]==1).astype('int32') * (seg[:, :, k]==1).
    ↪ astype('int32'))*2.0 / (np.sum(seg2[:, :, k]==1).astype('int32') + np.
    ↪ sum(seg[:, :, k]==1).astype('int32'))
        print('Dice similarity score is {}'.format(dice))

del i
```

Average Hausdorff distance Coefficient

```
[ ]: i=0

for seg, seg2 in seg_loader:
    seg = seg.numpy()
    seg2 = seg2.numpy()
    seg=np.squeeze(seg)
    seg2=np.squeeze(seg2)
    print(prednames[i])
    i=i+1
    for k in range(6):
        hausdorff1 = scipy.spatial.distance.directed_hausdorff((seg[:, :,
    ↪ k]==1), (seg2[:, :, k]==1))
        hausdorff2 = scipy.spatial.distance.directed_hausdorff((seg2[:, :,
    ↪ k]==1), (seg[:, :, k]==1))
        H1 = hausdorff1[0]
```

```
H2 = hausdorff2[0]
avg_Hausdorff = (H1 + H2) / 2
print('Hausdorff score is {}'.format(avg_Hausdorff))
```

Combineren van Segmentaties

```
[ ]: direct = '/content/drive/MyDrive/TGO/Persoonlijk_segmenteren/'
Heleen = sorted(glob(os.path.join(direct,"Heleen/Segmentations/Segmentation*.seg.
→seg.nrrd")))
Stan = sorted(glob(os.path.join(direct,"Stan/Segmentations/Segmentation*.seg.
→nrrd")))
Luca = sorted(glob(os.path.join(direct,"Luca/Segmentations/Segmentation*.seg.
→nrrd")))
Jesse =sorted(glob(os.path.join(direct,"Jesse/Segmentations/Segmentation*.seg.
→nrrd")))
```

```
[ ]: class Label_datasetcombi(Dataset):
    def __init__(self, Stan, Jesse, Heleen, Luca, classes=6):
        self.files_segs= Stan
        self.files_segj=Jesse
        self.files_seg1 = Luca
        self.files_segH = Heleen
        self.classes = classes
        #self.transform_seg = transform_seg
        #self.transform_seg2 = transform_seg2

    def __getitem__(self,index):
#stan
        seg_imgs = np.array(imageio.imread(self.files_segs[index]))

        seg_imgs = np.squeeze(seg_imgs)
        if seg_imgs.shape[0] == 2:
            seg_imgs = np.squeeze(seg_imgs[1, :, : ])
        if seg_imgs.shape[0] == 3:
            seg_imgs = np.squeeze(seg_imgs[1, :, : ])

        segs = np.zeros(shape=np.shape(seg_imgs) + (self.classes,))
#jesse
        seg_imgj = np.array(imageio.imread(self.files_segj[index]))
        seg_imgj = np.squeeze(seg_imgj)

        if seg_imgj.shape[0] == 2:
            seg_imgj = np.squeeze(seg_imgj[1, :, : ])
        if seg_imgj.shape[0] == 3:
            seg_imgj = np.squeeze(seg_imgj[1, :, : ])
        segj = np.zeros(shape=np.shape(seg_imgj) + (self.classes,))
#luca
```

```

seg_imgl = np.array(imageio.imread(self.files_seg1[index])) #

seg_imgl = np.squeeze(seg_imgl)
if seg_imgl.shape[0] == 2:
    seg_imgl = np.squeeze(seg_imgl[1, :, :])
if seg_imgl.shape[0] == 3:
    seg_imgl = np.squeeze(seg_imgl[1, :, :])

seg1 = np.zeros(shape=np.shape(seg_imgl) + (self.classes,))
#HELEEN
seg_imgh = np.array(imageio.imread(self.files_seg1[index]))
seg_imgh = np.squeeze(seg_imgh)

if seg_imgh.shape[0] == 2:
    seg_imgh = np.squeeze(seg_imgh[1, :, :])
if seg_imgj.shape[0] == 3:
    seg_imgh = np.squeeze(seg_imgh[1, :, :])
segh = np.zeros(shape=np.shape(seg_imgh) + (self.classes,))
for i in range(6):
    ind=np.nonzero(seg_imgs==i)
    ind2=np.nonzero(seg_imgj==i)
    ind3=np.nonzero(seg_imgl==i)
    ind4=np.nonzero(seg_imgh==i)
    segs[ind[0],ind[1],i]=1
    segj[ind2[0],ind2[1],i]=1
    segl[ind3[0],ind3[1],i]=1
    segh[ind4[0],ind4[1],i]=1
return segs,segj, segl, segh
def __len__(self):
    return len(self.files_segs)

```

```

[ ]: dataset_combi = Label_datasetcombi(Stan, Jesse, Luca, Heleen, 6)
combi_loader= DataLoader(dataset_combi, batch_size=1, shuffle=False,
    ↪ num_workers=2)
segs, segj, segl, segh = first(combi_loader)
print(segs.shape)

```

```

[ ]: import nrrd
import matplotlib.image as mpling
from matplotlib import cm

!pip install simpleitk
import SimpleITK as sitk

```

Creëren van combinatie segmentatie

```

[ ]: pixelgrootte = 0.08467
i=1

```

```

for segs, segj, segl, segh in combi_loader:
    segs= segs.detach().cpu().numpy()
    segj= segj.detach().cpu().numpy()
    segl= segl.detach().cpu().numpy()
    segh= segh.detach().cpu().numpy()
    seg= np.add(segs, segh)
    seg2=np.add(segl, segj)
    segcombi=np.add(seg, seg2)
    segcombi=np.squeeze(segcombi)
    segcombi=np.argmax(segcombi, axis=2)
    segcombi=segcombi.astype('uint8')
    combi_name= 'combiseg' + str(i) + '.nrrd'
    combi_file = os.path.join(direct,'combi', combi_name)

    for l in range(1):
        combi_itk = sitk.GetImageFromArray(segcombi)
        combi_itk.SetSpacing((pixelgrootte, pixelgrootte))
        combiseg = sitk.WriteImage(combi_itk, combi_file)
    i=i+1

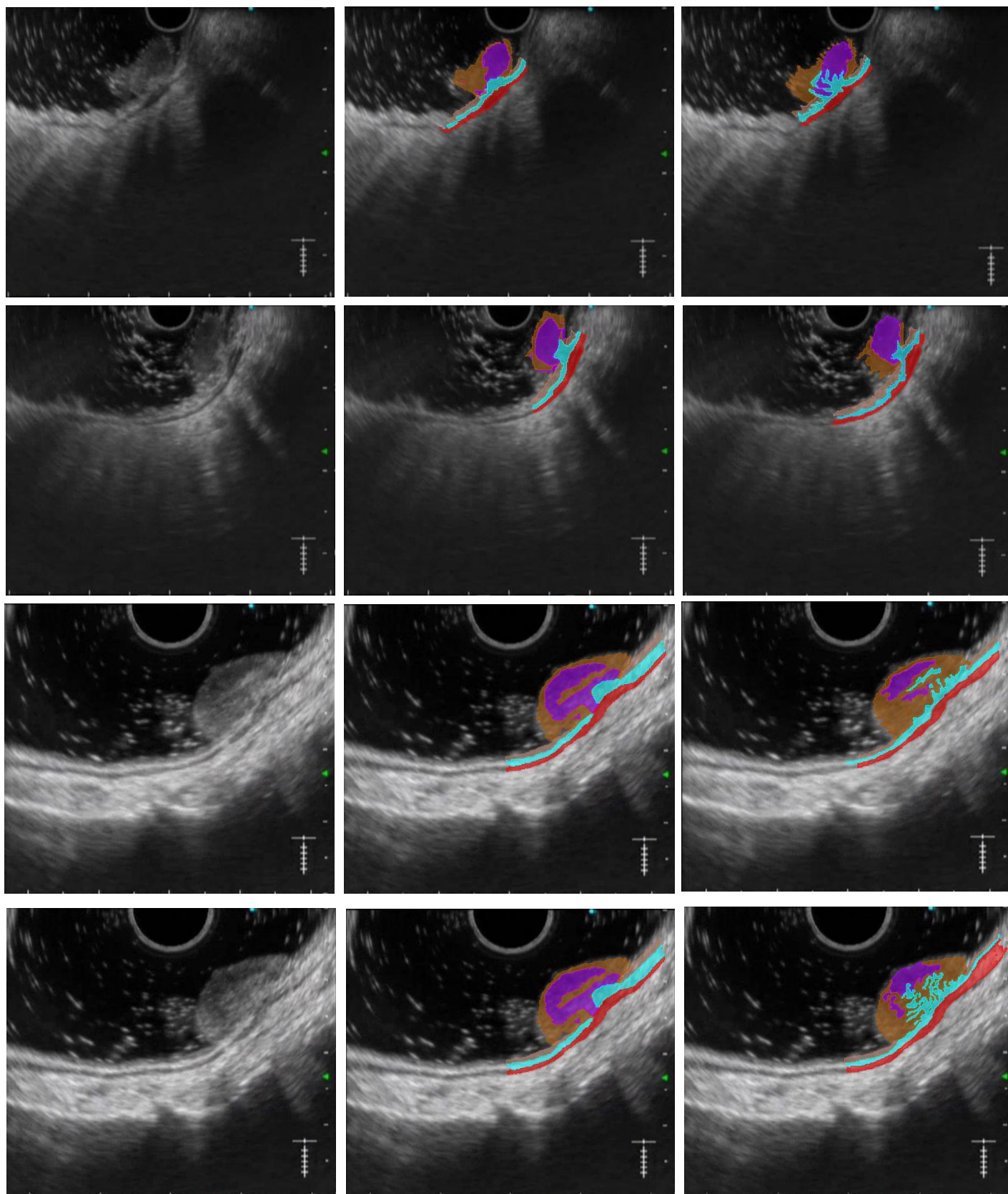
```

BIJLAGE E: VERSCHIL SEGMENTATIE TG-STUDENTEN EN EXPERT

EUS-BEELDEN

COMBINATIE TG-STUDENTEN

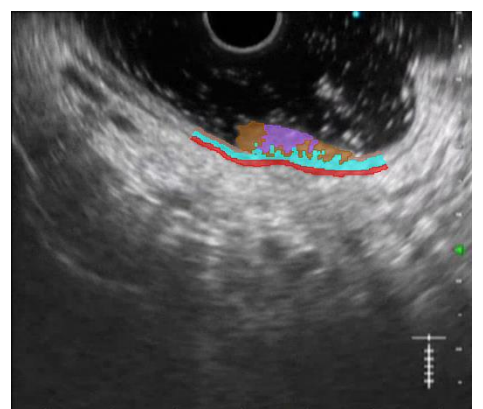
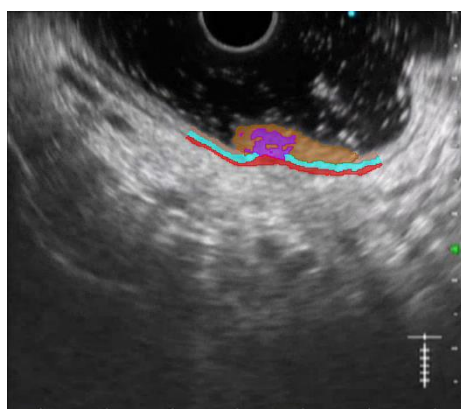
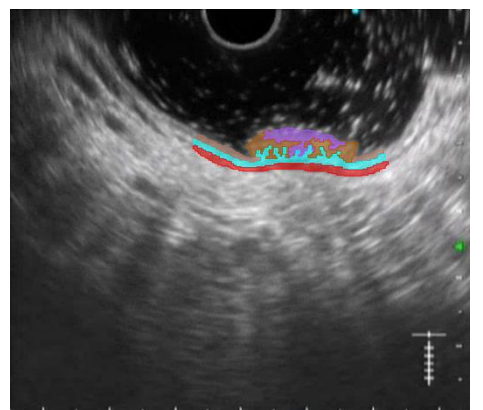
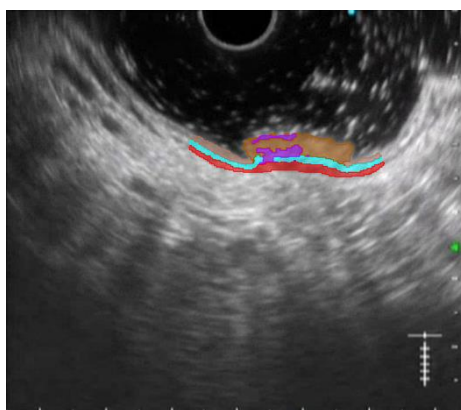
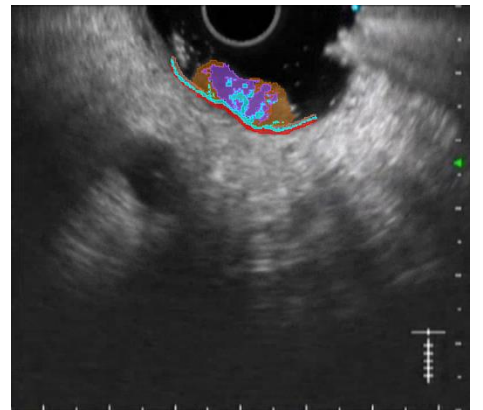
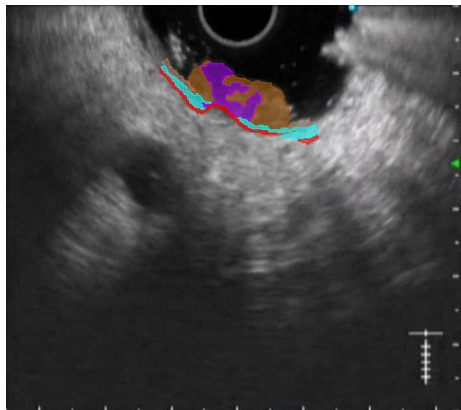
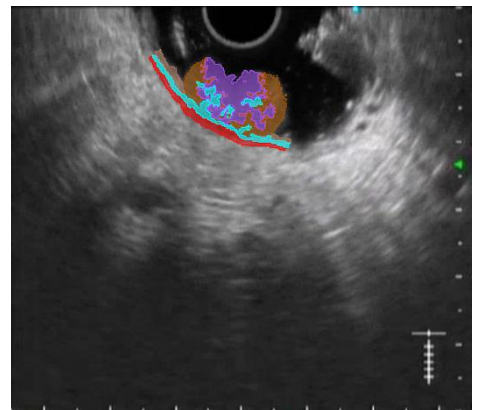
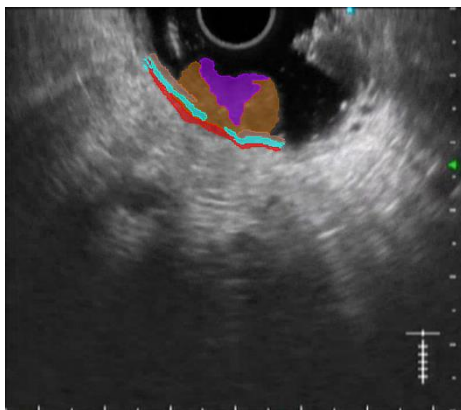
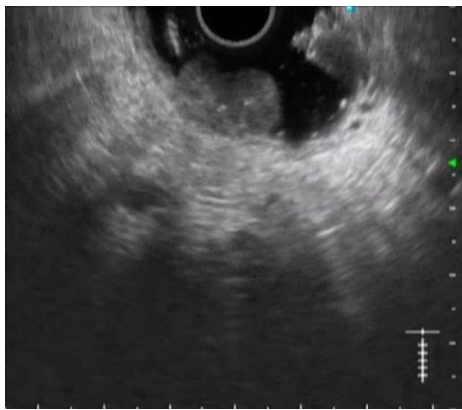
EXPERT



EUS-BEELDEN

COMBINATIE TG-STUDENTEN

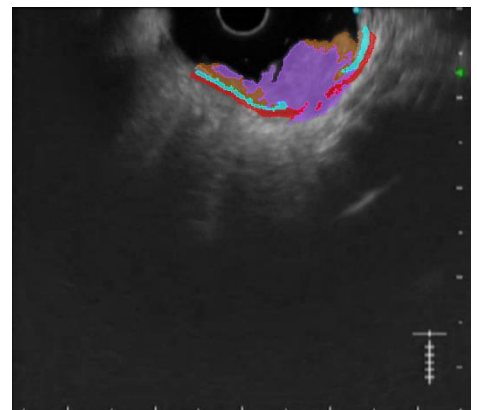
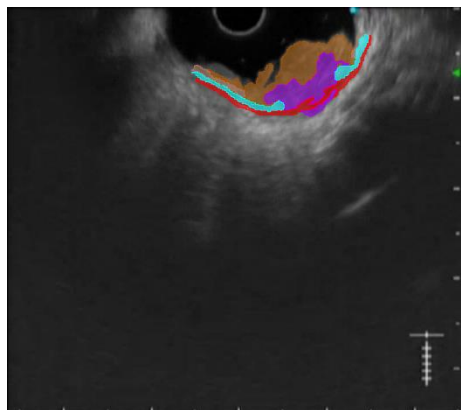
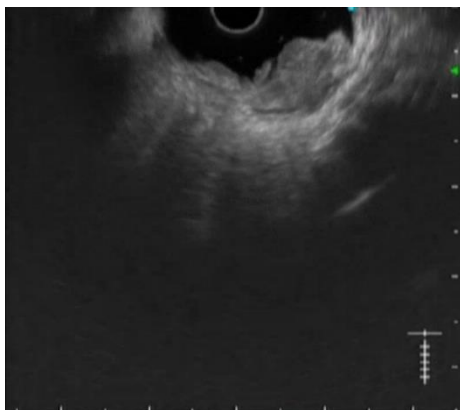
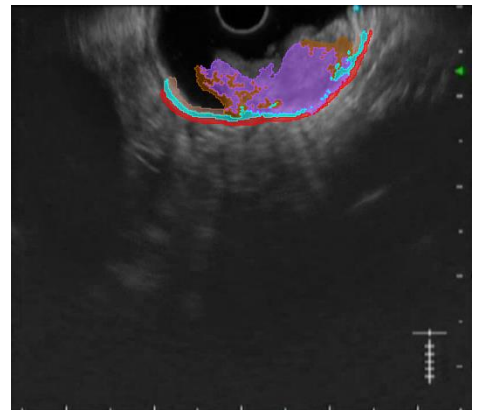
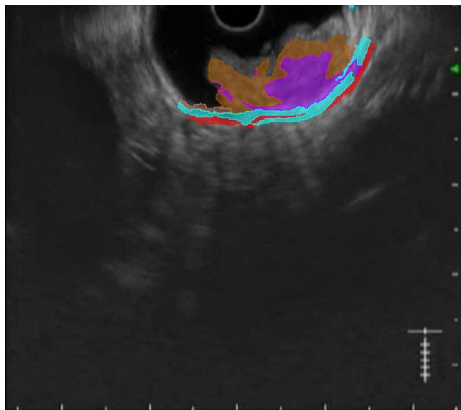
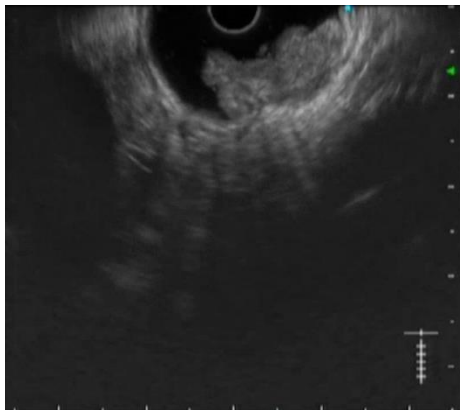
EXPERT



EUS-BEELD

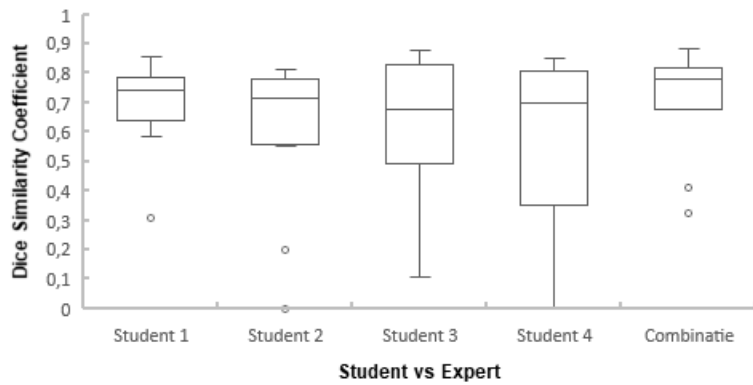
COMBINATIE TG-STUDENTEN

EXPERT

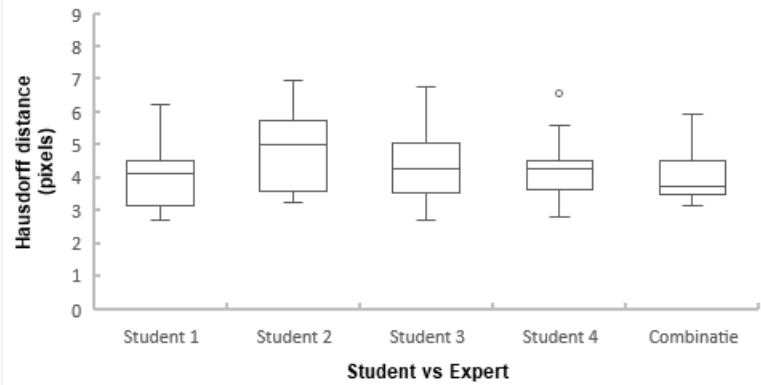


BIJLAGE F: BOXPLOTS INTERBEOORDELAARSVARIABILITEIT

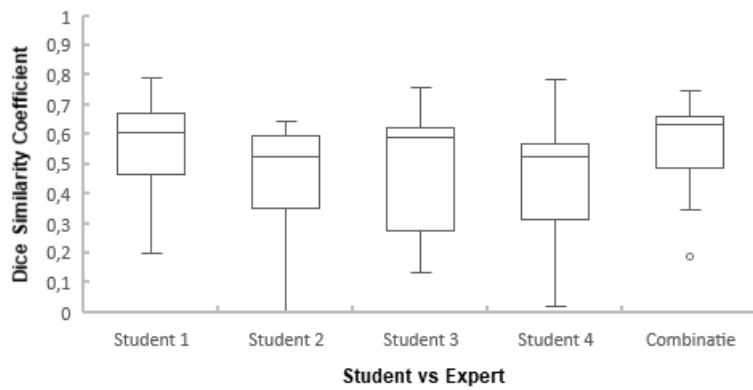
MUSCULARIS PROPRIA



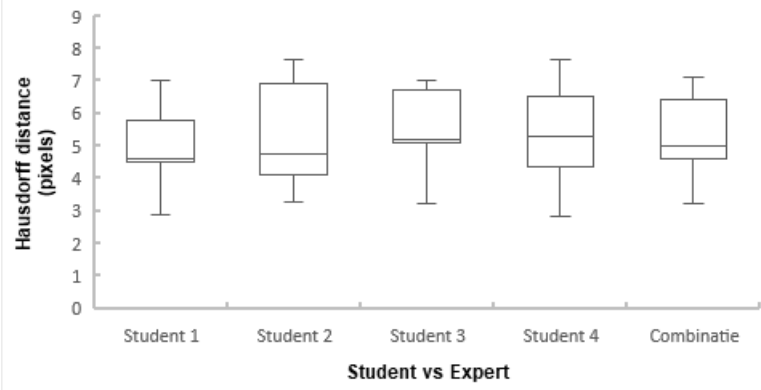
MUSCULARIS PROPRIA



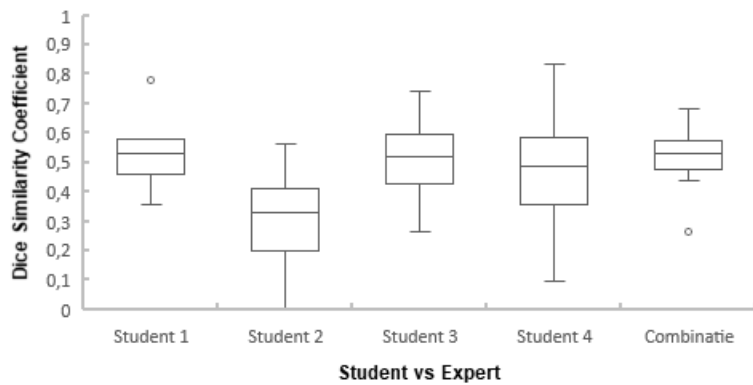
SUBMUCOSA



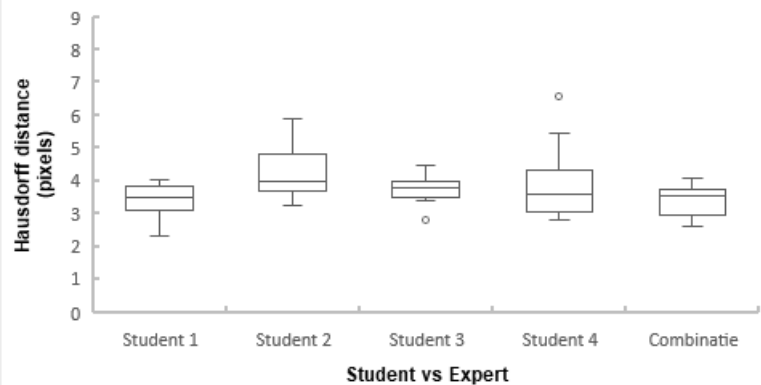
SUBMUCOSA



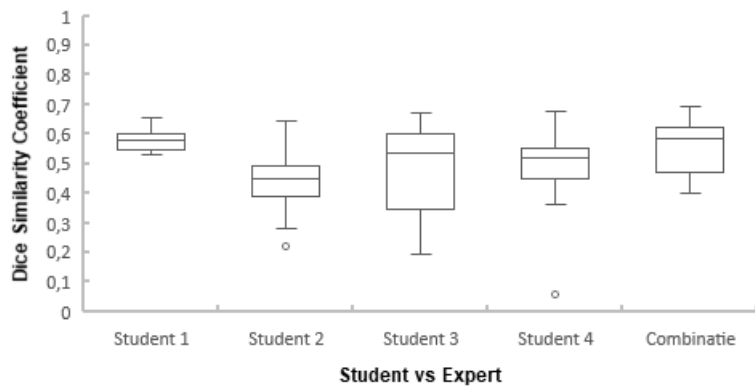
MUCOSA



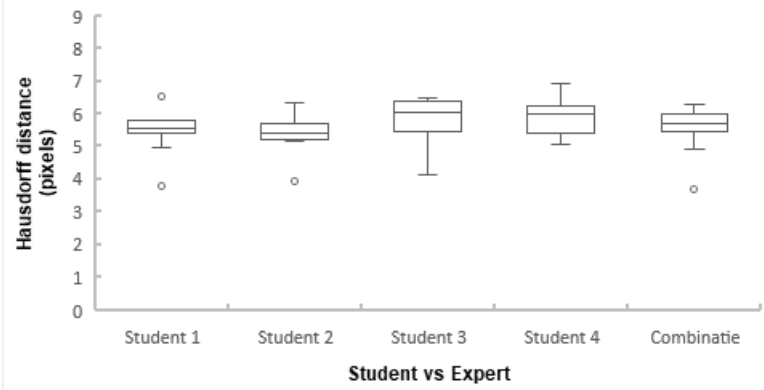
MUCOSA



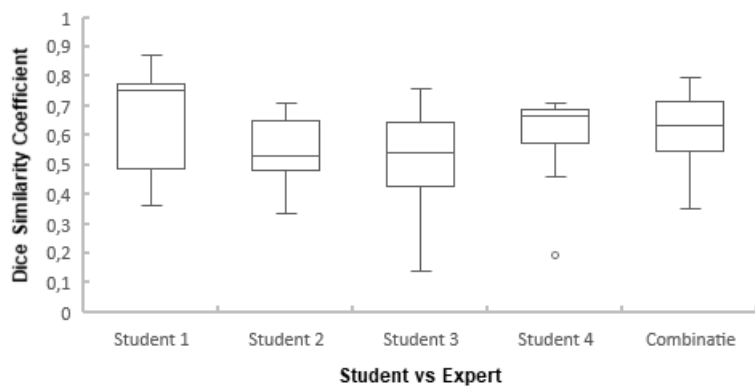
POLIEP (GOEDAARDIG)



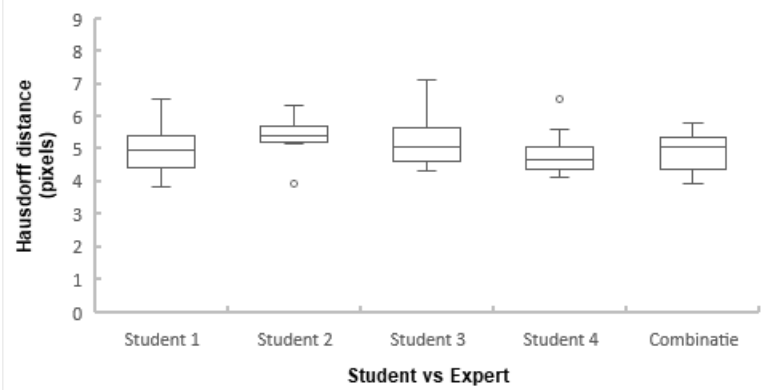
POLIEP (GOEDAARDIG)



POLIEP (KWAADAARDIG)



POLIEP (KWAADAARDIG)



BIJLAGE G: MODEL PYTHON SCRIPT

Importeren van bestanden en libraries.

```
[ ]: !pip install monai
[ ]: !pip install simpleitk
[ ]: !pip install pynrrd

[ ]: from google.colab import drive
[ ]: drive.mount('/content/drive')

[ ]: import pandas as pd
import matplotlib.pyplot as plt
from PIL import Image
import torch
import torch.nn.functional as F
import torchvision
import torchvision.transforms as transforms
from torchvision import datasets
from torch.utils.data import DataLoader, Dataset
from sklearn.model_selection import train_test_split
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import numpy as np
import math
from torch.utils.data.sampler import SubsetRandomSampler
from torch.utils.tensorboard import SummaryWriter
import os
from glob import glob
import imageio
import cv2
import monai
from monai.config import print_config
from monai.data import ArrayDataset, Dataset, CacheDataset
from monai.inferers import sliding_window_inference
from monai.metrics import DiceMetric, compute_meandice
from monai.transforms import(
    AddChannel,
    Compose,
```

```

    RandRotate90,
    RandSpatialCrop,
    ScaleIntensity,
    ToTensor,
    RandAdjustContrast,
    SqueezeDim,
    Transpose,
    Flip,
    SplitChannel,
    Rand2DElastic
)
from monai import transforms
from monai.losses import DiceLoss

from monai.visualize import plot_2d_or_3d_image
from monai.networks.nets import UNet, DenseNet
from monai.utils import first, set_determinism
from typing import Optional, Callable, Sequence, Tuple, TypeVar, Union
import time
from skimage import io, color

```

Set up transforms

```

[: train_imtrans = Compose(
    [
        Transpose([2,0,1]),
        ScaleIntensity(),
        ToTensor()
    ]
)

train_segtrans = Compose(
    [
        Transpose([2,0,1]),
        ToTensor()
    ]
)
train_testtrans= Compose(
    [
        Transpose([2,0,1]),
        ScaleIntensity(),
        ToTensor()
    ]
)

```

Set up directory paths for data

```
[ ]: direct = '/content/drive/MyDrive/TGO/Systeem/' #path naar map met afbeeldingen

images = sorted(glob(os.path.join(direct,"Images/EUS*.jpg"))) #afbeeldingen
segs = sorted(glob(os.path.join(direct,"SegmentationsFrank/Segmentation*.seg.
↳nrrd"))) #segmentaties

test = sorted(glob(os.path.join(direct,"Test/test*.jpg")))
```

Creëren van dataset generator

```
[ ]: class Label_dataset(CacheDataset):
    def __init__(self, files_seg, files_img, classes=6, transform_seg=None,
↳transform_img=None):
        self.files_seg = files_seg
        self.files_img = files_img

        self.transform_seg = transform_seg

        self.transform_img = transform_img
        self.classes = classes

    def __getitem__(self, index):
        img = Image.open(self.files_img[index]).convert('RGB') #afbeelding openen
        seg_img = np.array(imageio.imread(self.files_seg[index])) #segmentatie
↳openen
        seg_img = np.squeeze(seg_img)

        if seg_img.shape[0] == 2:
            seg_img = np.squeeze(seg_img[1, :, :])

        seg = np.zeros(shape=np.shape(seg_img) + (self.classes,)) #een zero-array
↳met de grootte van de classes creëren

        for i in range(6):
            #alle label classes langsgaan
            ind=np.nonzero(seg_img==i) #indices met bepaald label vinden

            seg[ind[0],ind[1],i]=1 #sheet met labels vullen

        if self.transform_seg:
            seg = self.transform_seg(seg) #transformaties toepassen op segmentatie

        if self.transform_img:
            img = self.transform_img(np.array(img)) #transformaties op afbeelding
↳toepassen

        return (img.double(),seg.double()) #segmentatie en image worden weergegeven
```

```
def __len__(self):
    return len(self.files_seg)
```

Test dataset creëren

```
[ ]: class Label_dataset_test(CacheDataset):
    def __init__(self, files_test, classes=6, transform_test=None):
        self.files_test= files_test

        self.transform_test = transform_test
        self.classes = classes

    def __getitem__(self, index):
        testing = Image.open(self.files_test[index]).convert('RGB')

        if self.transform_test:
            testing = self.transform_test(np.array(testing)) #transformaties op
            ↳afbeelding toepassen

        return (testing.double())

    def __len__(self):
        return len(self.files_test)
```

Dataset en dataloader opzetten

```
[ ]: dataset_train = Label_dataset(segs, images, 6, train_segtrans, train_imtrans) ↳
    ↳#Apply Dataset creator

valid_size = 0.25 #validatie van portie
batch_size = 10 #grootte van batch size
```

Indices vinden voor validatie set

```
[ ]: num_train = len(dataset_train)
indices = list(range(num_train))
```

Indices randomiseren

```
[ ]: np.random.shuffle(indices)
split = int(np.floor(num_train*valid_size))
train_index, valid_index = indices[split:], indices[:split]

train_sampler = SubsetRandomSampler(train_index)
valid_sampler = SubsetRandomSampler(valid_index)

train_loader = DataLoader(dataset_train, batch_size=batch_size, ↳
    ↳sampler=train_sampler, num_workers=2, pin_memory=torch.cuda.is_available())
val_loader = DataLoader(dataset_train, batch_size=1, sampler=valid_sampler, ↳
    ↳num_workers=0)
```

```

dice_metric = DiceMetric(include_background=False, reduction="mean_channel")

img, seg = first(train_loader)
print(img.shape, seg.shape)

```

Model opzetten

```

[:]: device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = monai.networks.nets.UNet(
    dimensions=2,
    in_channels=3,
    out_channels=6,
    channels=(16, 32, 64, 128, 256),
    strides=(2, 2, 2, 2),
    num_res_units=2,
).to(device)
model.double()
loss_function = monai.losses.DiceLoss(include_background=True, softmax=True)
optimizer = torch.optim.Adam(model.parameters(), 1e-1)
print(model)

```

Training

```

[:]: epochs = 500
val_interval = 1
best_metric = -1
best_metric_epoch = -1
best_loss = 10
best_loss_epoch = -1
epoch_loss_values = list()
metric_values = list()

save_directory_model='/content/drive/My Drive/TGO/Systeem/'
timestr = time.strftime("%Y%m%d-%H%M")
model_name_m=os.path.join(save_directory_model, "Metric",
    ↳"best_metric_model_segmentation2_array" + timestr + ".pth")
model_name_l=os.path.join(save_directory_model, "Loss",
    ↳"best_metric_model_segmentation2d_array" + timestr + ".pth")

for epoch in range(epochs):
    print("-" * epochs)
    print(f"epoch {epoch + 1}/{epochs}")
    model.train()
    epoch_loss = 0
    step = 0

    for batch_data in train_loader:
        step += 1
        inputs, labels = batch_data[0].to(device), batch_data[1].to(device)

```

```

optimizer.zero_grad()
outputs = model(inputs)
loss = loss_function(outputs, labels)
loss.backward()
optimizer.step()
epoch_loss += loss.item()
epoch_len = len(dataset_train) // train_loader.batch_size
print(f"{step}/{epoch_len}, train_loss: {loss.item():.4f}")

epoch_loss /= step
epoch_loss_values.append(epoch_loss)
print(f"epoch {epoch + 1} average loss: {epoch_loss:.4f}")

if (epoch + 1) % val_interval == 0:
    model.eval()
    with torch.no_grad():
        metric_sum = 0.0
        metric_count = 0
        val_images = None
        val_labels = None
        val_outputs = None
        for val_data in val_loader:
            val_images, val_labels = val_data[0].to(device), val_data[1].
            →to(device)
            roi_size = (96, 96)
            sw_batch_size = 4
            val_outputs = model(val_images)
            value, _ = dice_metric(y_pred=val_outputs, y=val_labels)
            metric_count += len(value)
            metric_sum += value.item() * len(value)
            metric = metric_sum / metric_count
            metric_values.append(metric)
        if metric > best_metric:
            best_metric = metric
            best_metric_epoch = epoch + 1
            torch.save(model.state_dict(), model_name_m)
            print("saved new best metric model")
        if epoch_loss < best_loss:
            a=1
            best_loss = epoch_loss
            best_loss_epoch = epoch + 1
            torch.save(model.state_dict(), model_name_l)
            print("saved new best loss model")
        print(
            "current epoch: {} current mean dice: {:.4f} best mean dice: {:.
            →4f} at epoch {}".format(
                epoch + 1, metric, best_metric, best_metric_epoch

```

```

        )
    )
    print(
        "current epoch: {} current loss: {:.4f} best loss: {:.4f} at_
↪epoch {}".format(
            epoch + 1, epoch_loss, best_loss, best_loss_epoch
        )
    )

print(f"train completed, best_metric: {best_metric:.4f} at epoch:_
↪{best_metric_epoch}")

```

Evaluatie model

```

[: dataset_test = Label_dataset_test(test, 6, train_testtrans)
test_loader= DataLoader(dataset_test,batch_size=1, num_workers=0 )
testing = first(test_loader)
print(testing.shape)

```

Creëren van segmentaties van testbeelden

```

[: import nrrd
import matplotlib.image as mpling
from matplotlib import cm

!pip install simpleitk
import SimpleITK as sitk

[: model.eval()
j=0
test_seg_files=[]
pixelgrootte = 0.08467
with torch.no_grad():
    for testing in test_loader:
        testing=testing.to(device)
        h_seg= model(testing.double())
        h_seg = h_seg.detach().cpu().numpy()
        h_seg= np.squeeze(h_seg)
        h_seg=np.argmax(h_seg, axis=0)
        h2_name = 'test-' + str(j) + '.nrrd'
        h_file = os.path.join(direct,'EvaluatieFrank/Evaluatiebeelden Frank na_
↪training Stan 11 juni', h2_name)
        h_seg = h_seg.astype('uint8')

        for l in range(1):
            h_seg_itk = sitk.GetImageFromArray(h_seg)
            h_seg_itk.SetSpacing((pixelgrootte, pixelgrootte))
            im_seg_h = sitk.WriteImage(h_seg_itk, h_file)
            test_seg_files.append(h_file)
        print(j)

j=j+1
del j

```

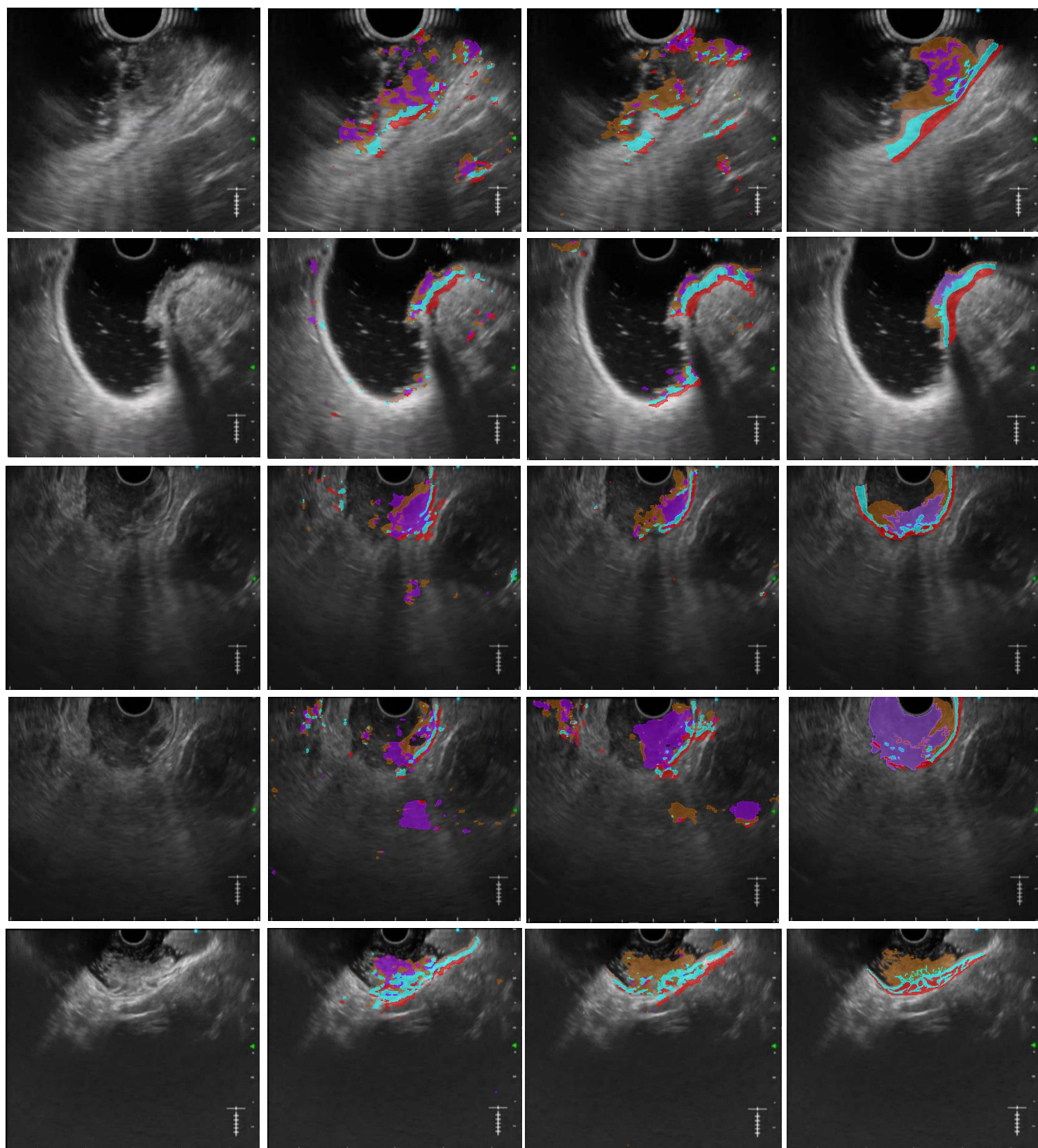
BIJLAGE H: VERSCHIL MODEL TG-STUDENTEN EN EXPERT

EUS-BEELD

MODEL TG-STUDENTEN

MODEL EXPERT

VERGELIJKING EXPERT

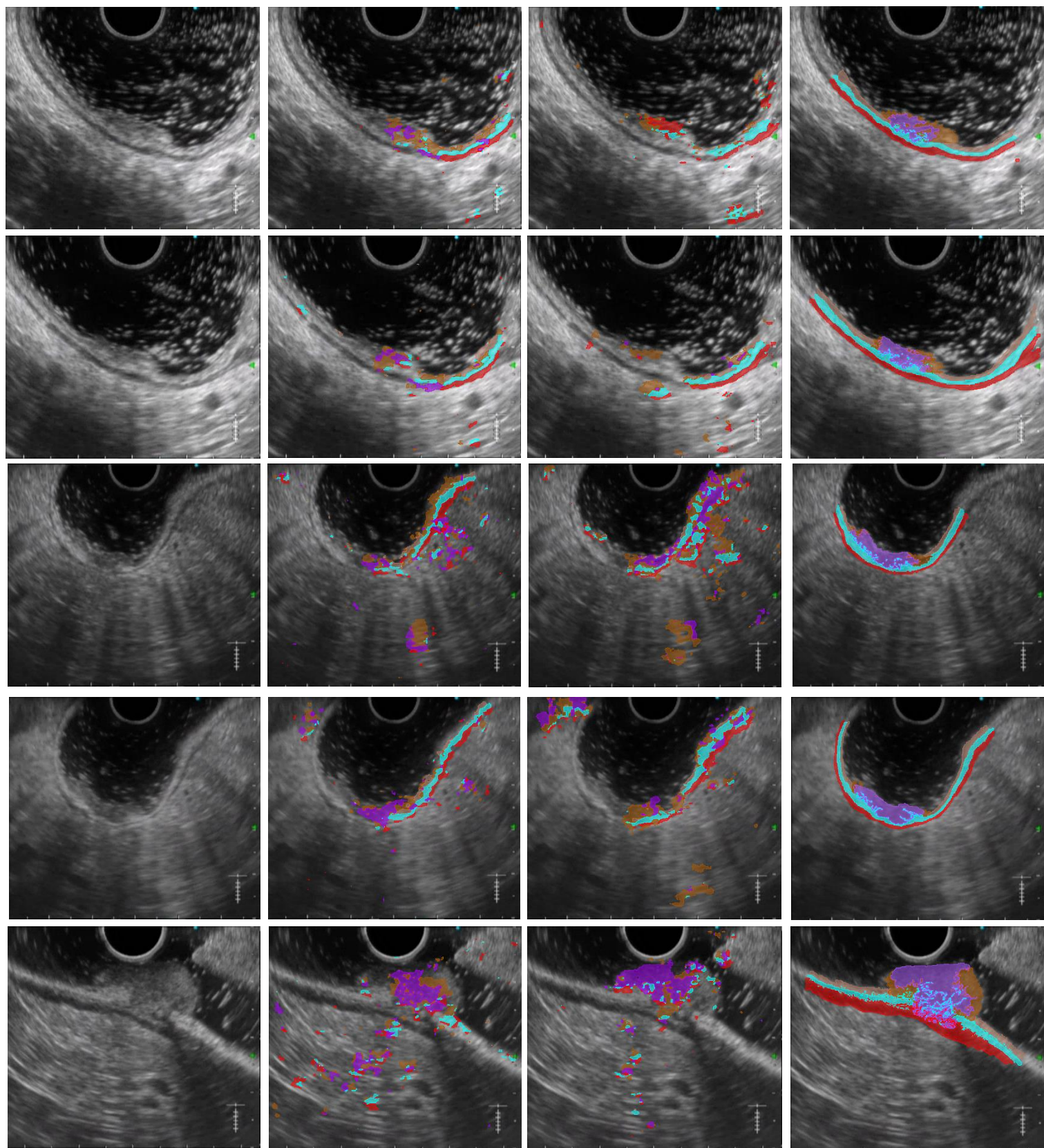


EUS-BEELD

MODEL TG-STUDENTEN

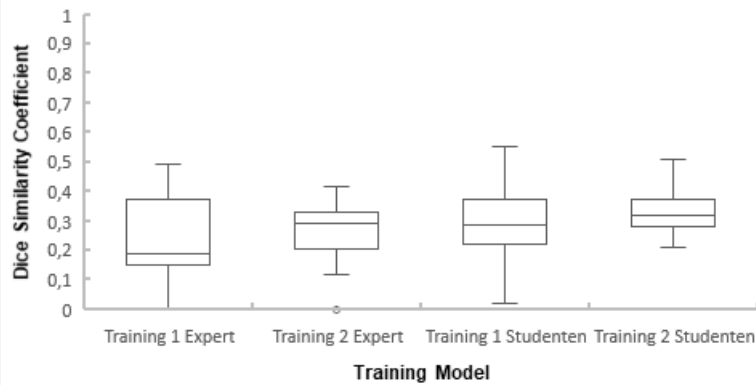
MODEL EXPERT

SEGMENTATIE EXPERT

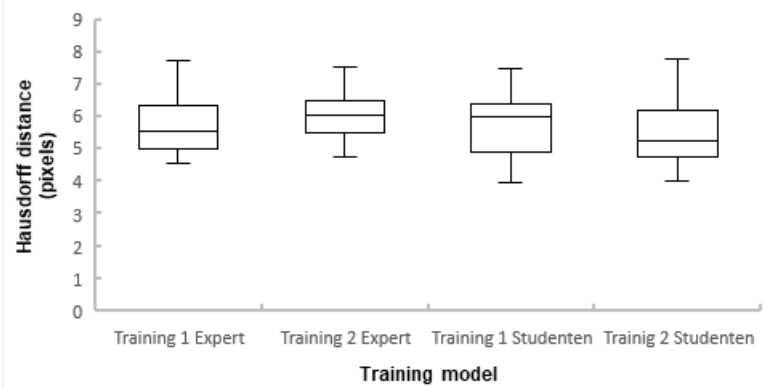


BIJLAGE I: BOXPLOTS MODELVARIABILITEIT

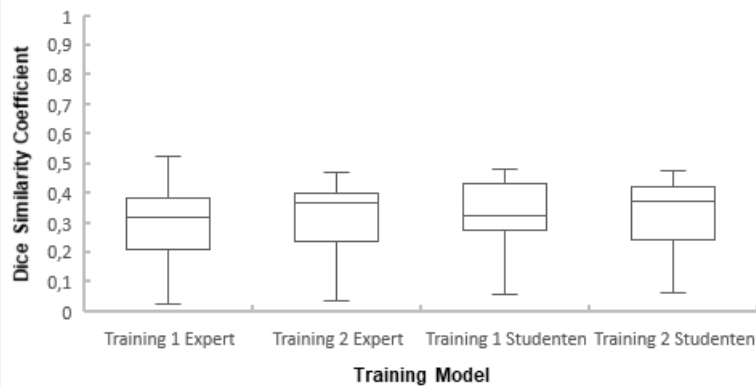
MUSCULARIS PROPRIA



MUSCULARIS PROPRIA



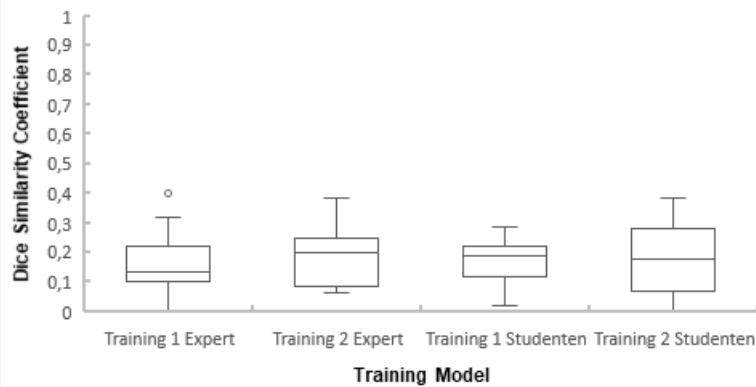
SUBMUCOSA



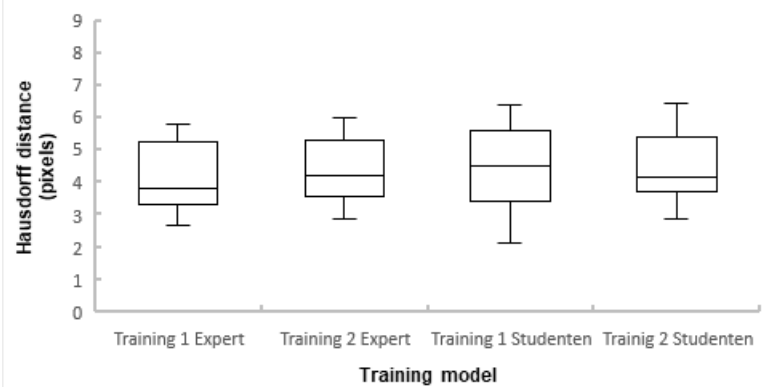
SUBMUCOSA



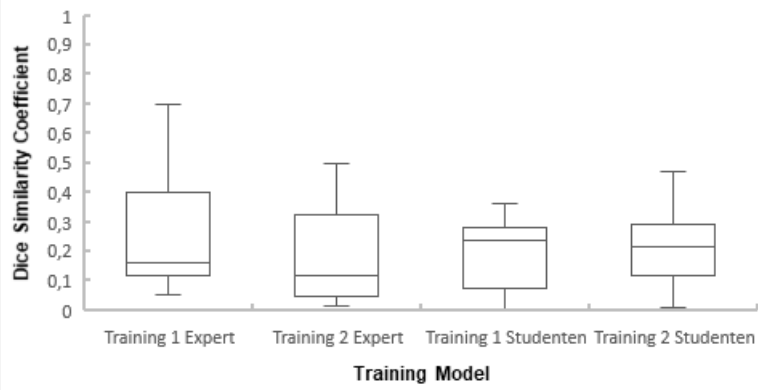
MUCOSA



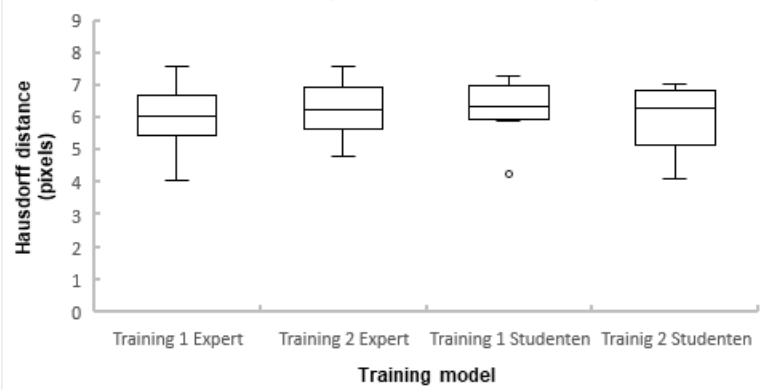
MUCOSA



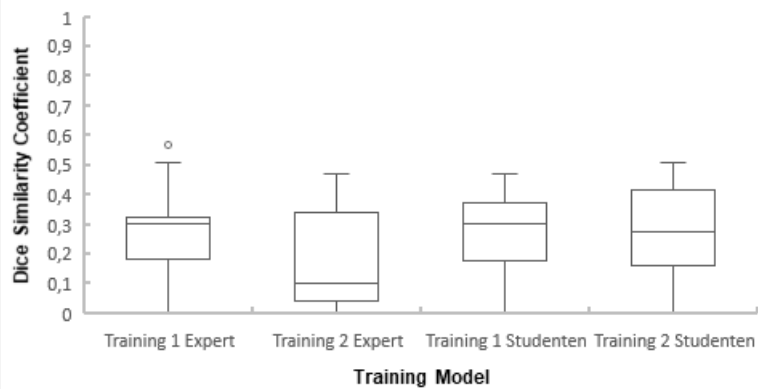
POLIEP (GOEDAARDIG)



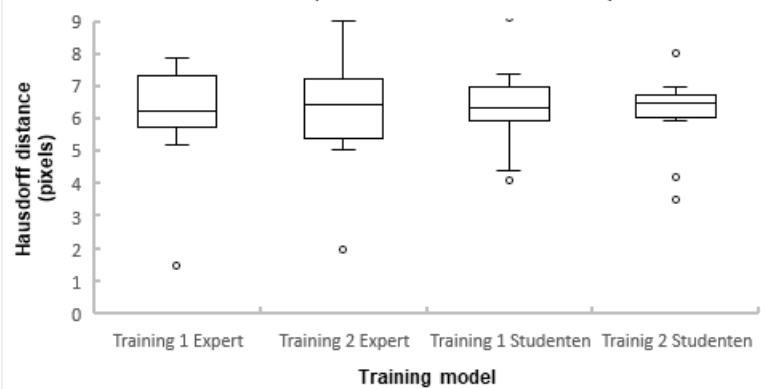
POLIEP (GOEDAARDIG)



POLIEP (KWAADAARDIG)



POLIEP (KWAADAARDIG)



REFERENTIES BIJLAGE

- [27] Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 2018;9:611–29. <https://doi.org/10.1007/s13244-018-0639-9>.
- [57] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017;39:2481–95. <https://doi.org/10.1109/TPAMI.2016.2644615>.
- [63] Ibtehaz N, Rahman MS. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Networks* 2020;121:74–87. <https://doi.org/10.1016/j.neunet.2019.08.025>.
- [93] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science n.d. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (accessed June 18, 2021).
- [94] Ghosh A, Sufian A, Sultana F, Chakrabarti A, De D. Fundamental concepts of convolutional neural network. *Intell. Syst. Ref. Libr.*, vol. 172, Springer; 2019, p. 519–67. https://doi.org/10.1007/978-3-030-32644-9_36.
- [95] A Comprehensive Introduction to Different Types of Convolutions in Deep Learning | by Kunlun Bai | Towards Data Science n.d. <https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215> (accessed June 18, 2021).
- [96] Christopher J, Messou E. Handling Invalid Pixels in Convolutional Neural Networks. 2020.
- [97] Choi H, Ryu S, Kim H. Short-Term Load Forecasting based on ResNet and LSTM. 2018 IEEE Int. Conf. Commun. Control. Comput. Technol. Smart Grids, SmartGridComm 2018, Institute of Electrical and Electronics Engineers Inc.; 2018. <https://doi.org/10.1109/SmartGridComm.2018.8587554>.
- [98] Kataoka H, Iwata K, Satoh Y. Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection 2015.
- [99] Khvostikov A, Aderghal K, Krylov A, Catheline G, Benois-Pineau J. 3D Inception-based CNN with sMRI and MD-DTI data fusion for Alzheimer’s Disease diagnostics 2018.
- [100] Semantic Segmentation using Fully Convolutional Neural Networks | by Wilbur de Souza | Medium n.d. <https://medium.com/@wilburdes/semantic-segmentation-using-fully-convolutional-neural-networks-86e45336f99b> (accessed June 18, 2021).