



HOE KAN HET DETECTEREN VAN HERSENTUMOREN GEAUTOMATISEERD WORDEN?

INTERNE PROMOTOR: MARIE DEWITTE
EXTERNE PROMOTOR: ROBBERT VERBEKE

ONDERZOEKSVRAAG UITGEVOERD DOOR

OUASSIM BOUTALLISS

VOOR HET BEHALEN VAN DE GRAAD VAN BACHELOR IN DE

MULTIMEDIA & CREATIVE TECHNOLOGIES

HOWEST | 2021-2022

Woord vooraf

Met deze bachelorproef die gaat over het automatiseren van de detectie van hersentumoren sluit ik mijn driejarige opleiding van Multimedia en Communicatie Technologie aan Howest te Kortrijk af. In deze bachelorproef bespreek ik eerst de research en technische demo uit de module 'Researchproject'.

Het gekozen onderwerp komt voornamelijk door mijn persoonlijke interesse in de geneeskunde en AI. Een oude kennis van mij is namelijk overleden aan een misdiagnose van een hersentumor. Hierdoor ben ik gemotiveerd om mijn uiterst best te doen voor het maximale uit dit project te halen. Met dit onderzoek wil ik vermijden dat dit nog eens voorkomt.

Vooraf dank ik al de docenten van Howest Kortrijk. Hierbij zou ik ook mijn begeleider van researchproject willen bedanken namelijk Marie Dewitte.

Tot slot wil ik mijn ouders en vrienden bedanken voor hun emotionele steun en vitale gesprekken. Door mijn bevindingen en conclusies tijdens het onderzoek met hen te delen, kon ik mijn gedachten beter ordenen en kreeg ik het vertrouwen dat ik een waardevol onderzoek had uitgevoerd.

Abstract

De onderzoeksvraag van deze bachelorproef is “Hoe kan het detecteren van hersentumoren geautomatiseerd worden?”. Met deze onderzoeksvraag wil ik ervoor zorgen dat hersentumors vroeg en nauwkeurig gedetecteerd kunnen worden en zo het leven van een patiënt redden.

Dit onderzoek is in twee onderdelen verdeeld. Het ene onderdeel gaat over het classificeren van hersentumoren. Hierbij werd er gebruik gemaakt van verschillende CNN modellen. In het tweede onderdeel gaat het over het segmenteren van de tumor zelf. Hierbij is er gebruik gemaakt van twee verschillende segmentatie modellen: UNet en Deeplabv3+. Inspiratie voor de gebruikte algoritmes kwam vooral van de populariteit op het internet.

Aan de hand van de verschillende uitgevoerde tests wordt er geëvalueerd welke methoden van praktische uitvoering het best zouden werken. Er wordt vooral rekening gehouden met de accuraatheid, f1-score van de tumor klasse en de training snelheid.

Het aantal false positives en de accuraatheid zijn heel belangrijke aspecten bij het classificeren van de hersentumoren. Het is heel cruciaal dat het model meer false positives (valse alarm van tumor) voorspellingen doet, dan false negatives (tumor aanwezig, maar niet voorspeld wordt) dit omdat het een leven kan redden. Bij de segmentatie gedeelte zijn de dice_coef, training snelheid en de gereconstrueerde images van belang.

Uit dit onderzoek bleek dat ensemble learning betere resultaten behaalde dan wanneer er gebruik gemaakt werd van één model. Er kon ook geconcludeerd worden dat het effectief wel mogelijk is om de detectie van hersentumoren te gaan automatiseren doormiddel van AI technieken.

De gebruikte modellen in de proof of concept applicatie die later aan bod komt behaalden een zeer hoge accuracy score.

Inhoudopgave

Figurenlijst	8
Lijst met afkortingen	10
Verklarende woordenlijst	11
1 Inleiding	12
2 Research	13
2.1 Wat is een hersentumor?	13
2.1.1 Soorten hersentumoren	13
2.1.2 Hoe wordt de diagnose gesteld	13
2.1.3 Wat is een MRI scan	14
2.2 Welke algoritme wordt toegepast op het classificeren van MRI scans?	14
2.2.1 Transfer Learning?	14
2.3 Welke techniek kan er toegepast worden om de tumor te gaan segmenteren?	15
2.3.1 Semantic versus Instance Segmentation	15
2.3.2 Verschillende semantic segmentatie methodes	16
2.3.3 UNet	16
2.3.4 Deeplabv3+	17
2.3.5 Fully Convolutional Network (FCN)	18
2.3.6 Ensemble learning	18
3 Technisch onderzoek	19
3.1 Algemene informatie	19
3.1.1 Programmeertaal en libraries	19
3.1.2 Algoritmes	19
3.2 Algemeen resultaat	19
3.2.1 Classificatie gedeelte	19
3.2.2 Segmentatie gedeelte	19
3.2.3 Applicatie	20
3.3 MRI scans classificatie	20
3.3.1 Werken de modellen beter op grijswaardenafbeeldingen?	20
3.3.2 Behaalde resultaten	20
3.3.3 Eigen CNN modellen vs Transfer learning modellen	26
3.3.4 Beste resultaat	26
3.4 Segmentatie van de tumor	27
3.4.1 Data voorbereiding	27
3.4.2 UNet	28
3.4.3 Deeplabv3+	30
3.4.4 Model vergelijken	31
3.5 Proof of concept applicatie	33
3.5.1 Werking	33
3.5.2 Gebruikte modellen	34

3.5.3	Output.....	34
3.6	Moeilijkheden	35
4	Reflectie.....	36
4.1	Zelfreflectie	36
4.1.1	Wat ging er beter dan verwacht.....	36
4.1.2	Wat ging er minder dan verwacht.....	36
4.1.3	Wat zou ik anders doen?.....	36
4.2	Reflectie van het resultaat met externe	36
4.2.1	Sterke punten	36
4.2.2	Zwakke punten	36
4.2.3	Wat zou er beter gekund hebben	37
4.3	Bruikbaarheid.....	37
4.3.1	Implementatie hindernissen.....	37
4.3.2	Implementatie	37
4.4	Meerwaarde.....	37
4.4.1	Praktisch/maatschappelijk relevant?.....	37
4.5	Vervolg onderzoek.....	38
4.5.1	Algoritmes	38
4.5.2	Masks.....	38
4.5.3	Ernst van de diagnose.....	38
5	Advies.....	39
5.1	Bruikbaarheid.....	39
5.1.1	top 5 van de 10 Ai toepassingen in de gezondheidszorg	39
5.1.2	Pros en cons van AI in de gezondheidszorg	40
5.1.3	Onderzoek in de praktijk	40
5.2	Concrete aanbevelingen	40
5.3	Stappenplan	41
5.4	Ontwikkelde tools.....	41
6	Conclusie.....	42
7	Literatuurlijst	43
8	Bijlages.....	45
	Bronnen	49

Figurenlijst

Figuur 1 - Hersentumor [7]

Figuur 2 - Verschillende hersentumoren [6]

Figuur 3 - MRI Scan [7]

Figuur 4 - Transfer Learning [12]

Figuur 5 – Semantic Segmentatie [13]

Figuur 6 - UNet [15]

Figuur 7 - Deeplabv3+ [18]

Figuur 8 – RGB versus grijs score

Figuur 9 - Eigen CNN Loss

Figuur 10 - Eigen CNN Metrics

Figuur 11 - Eigen CNN met data augmentatie loss

Figuur 12 - Eigen CNN met augmentatie Metrics

Figuur 13 - Transfer learning: EfficientNetB0 loss

Figuur 14 - Transfer learning: EfficientNetB0 scores

Figuur 15 - Transfer learning: InceptionV3 loss

Figuur 16 - Transfer learning: InceptionV3 scores

Figuur 16 - Transfer learning: InceptionV3 scores

Figuur 17 -Transfer learning: ResNet50 scores

Figuur 18 - Transfer learning: VGG19 loss

Figuur 19 - Transfer learning: VGG19 scores

Figuur 20 – Vergelijking beste classificatie modellen

Figuur 21 - beste resultaten

Figuur 22 - moeilijkheid ensemble model

Figuur 23 - pre-processed data

Figuur 25 -- UNET model code

Figuur 27 - Deeplabv3+ model code

Figuur 28 - Deeplabv3+ loss

Figuur 29 - Segmentatie Model vergelijken

Figuur 30 – Unet resultaat epoch

Figuur 31 - UNET resultaat easy part

Figuur 33 - Deeplabv3+ resultaat epoch

Figuur 34 - - Deeplabv3+ resultaat easy part

Figuur 35 - Deeplabv3+ resultaat hard part

Figuur 36 - Proof of concept voorbeeld

Figuur 37 - proof of concept input

Figuur 38 - proof of concept output 1

Figuur 39 - proof of concept output 2

Lijst met afkortingen

AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
CNN	Convolutional Neural Network
MRI	Magnetic Resonance Imaging
CDSS	Clinical Decision Support Systems

Verklarende woordenlijst

Data augmentatie: Gegevensvergroting bij gegevensanalyse zijn technieken die worden gebruikt om de hoeveelheid gegevens te vergroten door licht gewijzigde kopieën van reeds bestaande gegevens of nieuw gemaakte synthetische gegevens van bestaande gegevens toe te voegen.[1]

Deep Learning: Deep learning of diep leren is een deel van een grotere familie van methodes van machinaal leren, gebaseerd op kunstmatige neurale netwerken. Het leren kan gesuperviseerd gebeuren, semi-gesuperviseerd, of niet gesuperviseerd.[2]

Ensemble Learning: In statistiek en machine learning gebruiken ensemblemethoden meerdere leeralgoritmen om betere voorspellende prestaties te verkrijgen dan zou kunnen worden verkregen met elk van de samenstellende leeralgoritmen alleen.[3]

Transfer Learning: In statistiek en machine learning gebruiken ensemblemethoden meerdere leeralgoritmen om betere voorspellende prestaties te verkrijgen dan zou kunnen worden verkregen met elk van de samenstellende leeralgoritmen alleen.[4]

1 Inleiding

Deze bachelorproef genaamd “Hoe kan het detecteren van hersentumoren geautomatiseerd worden?”. Onderzoekt, of het al dan niet mogelijk is om met verschillende algoritmes de hersentumoren automatisch te gaan detecteren. Het gekozen onderwerp komt voornamelijk door mijn persoonlijke interesse in de geneeskunde en AI. Een oude kennis van mij is namelijk overleden aan een misdiagnose van een hersentumor. Hierdoor ben ik gemotiveerd om mijn uiterst best te doen voor het maximale uit dit project te halen. Met dit onderzoek wil ik vermijden dat dit nog eens voorkomt.

De data voor dit onderzoek zijn verzameld door Mateusz Buda. De data kan bezichtig worden door op de link te klikken. <https://www.kaggle.com/mateuszbuda/lgg-mri-segmentation>. De data bestaat uit MRI scans van de hersenen. Iedere MRI scan heeft ook zijn eigen masker, daaruit kan afgeleid worden of er een tumor aanwezig is. De dataset is ongebalanceerd, maar de data bestaat uit genoeg hersenen met tumoren als zonder. Dit werd simpelweg aangepakt door hersenen te selecteren die wel als geen tumoren bevatten en zo eigenlijk de gelijkheid een beetje ter creëren voor de trainingsdata. De dataset bestaat uit glioom tumoren. De scans werden geschaald naar 128x128px, omdat de computer waarop getraind werd het niet anders niet aankan.

Dit onderzoek is in twee onderdelen verdeeld. Het ene onderdeel gaat over het classificeren van hersentumoren. Hierbij werd er gebruik gemaakt van verschillenden CNN modellen: eigen kleine Convolutional Neural Network (CNN), CNN met augmentatie toegepast, EfficientNetB0, InceptionV3, ResNet50 en VGG19. De methode die in deze studie gebruikt werden, is een gemengde aanpak gebaseerd op de thesis van S. Gull, S. Akbar, en H. U. Khan, met als titel , ‘Automated Detection of Brain Tumor through Magnetic Resonance Images Using Convolutional Neural Network’. [5]

Een convolutioneel neuraal netwerk behoort tot de deep learning algoritmes. CNN's vertegenwoordigen een enorme doorbraak in de computer vision. Ze worden voornamelijk gebruik om afbeeldingen te classificeren. CNN wordt op het huidig moment al gebruikt in de dagdagelijkse applicaties zoals: Facebook, Instagram en Snapchat. De algoritme kan gebruikt worden voor afbeelding van de gezondheidzorg tot aan beveiliging.

De gebruikte modellen komen vaak voor in gelijkaardige onderzoeken. In het tweede onderdeel gaat het over het segmenteren van de tumor zelf. Hierbij is er gebruik gemaakt van twee verschillende segmentatie modellen: UNet en Deeplabv3+. [5]

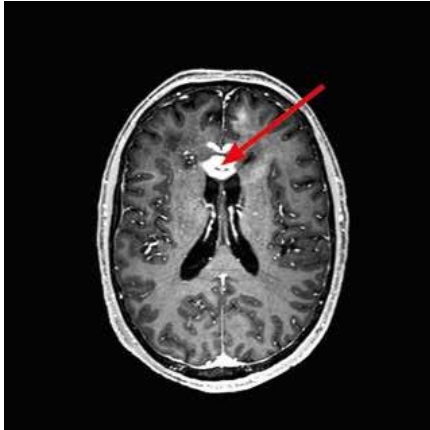
Voor dit onderzoek werden er verschillende deelvragen opgesteld om dit zo efficiënt mogelijk uit te werken. De opgestelde deelvragen zijn: “Welke algoritme wordt het best toegepast op het classificeren van hersenen? (wel of geen tumor)”, “Wordt er het best gewerkt met transfer learning of met een eigen CNN?” En “Welke techniek kan er toegepast worden om de tumor te gaan segmenteren?”.

De technische uitwerking gebaseerd op deze bachelorproef werd uitgevoerd in de maand januari van het jaar 2022. De technische uitwerking werd verricht met advies van docenten aan Howest.

2 Research

2.1 Wat is een hersentumor?

Een hersentumor is een groei van abnormale cellen in de hersenen tumoren kunnen in de hersenen beginnen, maar ook in andere delen van de lichaam en zich gaan uitzaaien naar de hersenen toe. [6]



Figuur 1 - Hersentumor [7]

2.1.1 Soorten hersentumoren

Er bestaan veel verschillende soorten hersentumoren. Sommige hersentumoren zijn niet-kwaadaardig en sommige zijn wel-kwaadaardig. De niet-kwaadaardige tumor heeft een schil om de cellen heen zitten. De cellen breiden zich wel uit, maar blijven binnen in de schil en gaan zich niet verder gaan verspreiden over het lichaam heen. Bij de kwaadaardige tumoren bevindt er zich geen schil rondom de tumor en zal zich dus wel gaan uitzaaien doorheen het lichaam.

Elk soort heeft zijn eigen kenmerken en vraag daarom dus om een andere behandeling. De dataset bevat de laaggradig glioom tumoren.

Glioom: tumor uit de steuncellen
Meningeoom: tumor in het hersenvlies
Tumor in de hypofyse
Brughoektumor
Tumor in of bij het ruggenmerg
Prolactinoom
Hemangioblastoom
Meningeoom

Figuur 2 - Verschillende hersentumoren [6]

2.1.2 Hoe wordt de diagnose gesteld

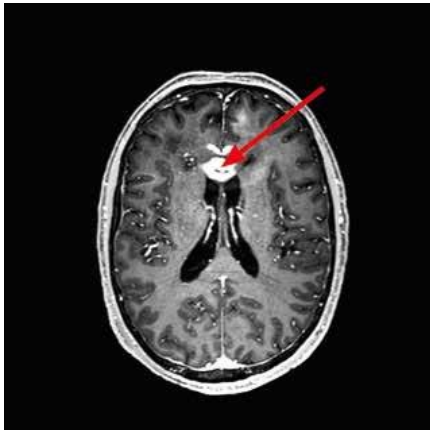
Bij het ervaren van de volgende symptomen: aanhoudende hoofdpijn, verandering in verstandelijk vermogen en gedrag, uitvalsverschijnselen en epileptische aanvallen, dan is een bezoekje aan de huisarts vereist. De huisarts zal een diagnose stellen en de patiënt dan doorverwijzen naar een neuroloog als het om een tumor blijkt te gaan.

Bij het eerste onderzoek als de patiënt zich bij de neuroloog bevindt, vraagt de specialist naar de ervaren klachten. Daarna zal er een MRI scan genomen worden van de hersenen.[8]

Vanaf dit gedeelte kan er gebruik worden gemaakt van verschillende AI algoritmes om de hersentumor te gaan detecteren en het leven van een patiënt te redden van een misdiagnose.

2.1.3 Wat is een MRI scan

Een MRI scan werkt met magneetveld en radiogolven. Een MRI-apparaat neemt foto's, hierop zal de arts kunnen waarnemen of er al dan niet een tumor aanwezig is. Zie voorbeeld hieronder.



Figuur 3 - MRI Scan [7]

2.2 Welke algoritme wordt toegepast op het classificeren van MRI scans?

CNN is de meest gebruikte algoritme voor allerlei computer vision taken, inclusief het classificeren van MRI scans. Er is een mogelijkheid om een eigen CNN model samen te stellen, maar wat hierbij van belang is de hoeveelheid data er aanwezig is. Indien er een kleine hoeveelheid aan data aanwezig is, dan is het veel efficiënter om te werken met transfer learning modellen. De dataset bestaat uit 2556 hersenen zonder tumoren en 1373 hersenen die wel een tumor bevatten. [10]

2.2.1 Transfer Learning?

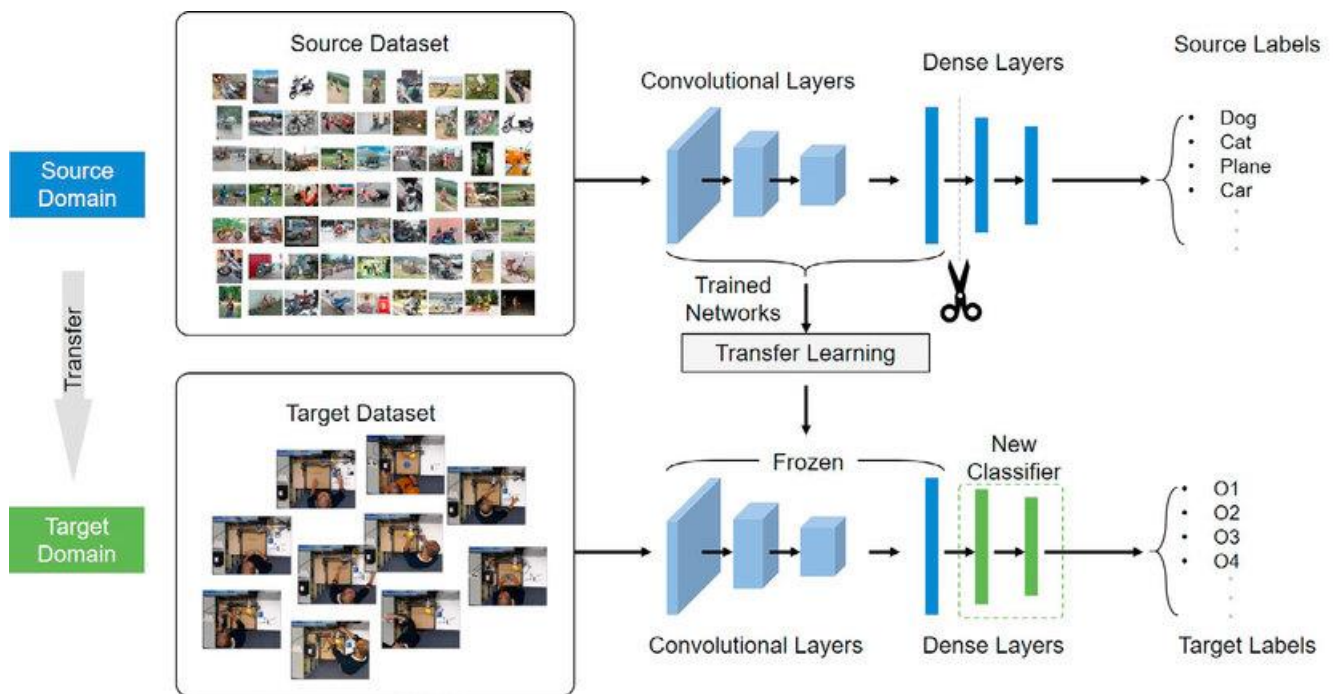
Indien de beschikbare data aan de lage kant is, dan kan er gebruik gemaakt worden van de transfer learning modellen. Transfer learning is een methode waarbij een voorgetrainde model op grootschalige data bruikbaar wordt, om te gebruiken in een eigen project. Dit laat toe om een hogere accuracy score te behalen alhoewel er weinig data beschikbaar is.

Er is een hele lijst aan transfer learning modellen beschikbaar voor CNN. [11]

Top 4 meest gebruikte modellen:

- Inception
- EfficientNet
- VGG19
- ResNet

Deze modellen zullen dan ook later aanbod komen en vergeleken worden.



Figuur 4 - Transfer Learning [12]

2.3 Welke techniek kan er toegepast worden om de tumor te gaan segmenteren?

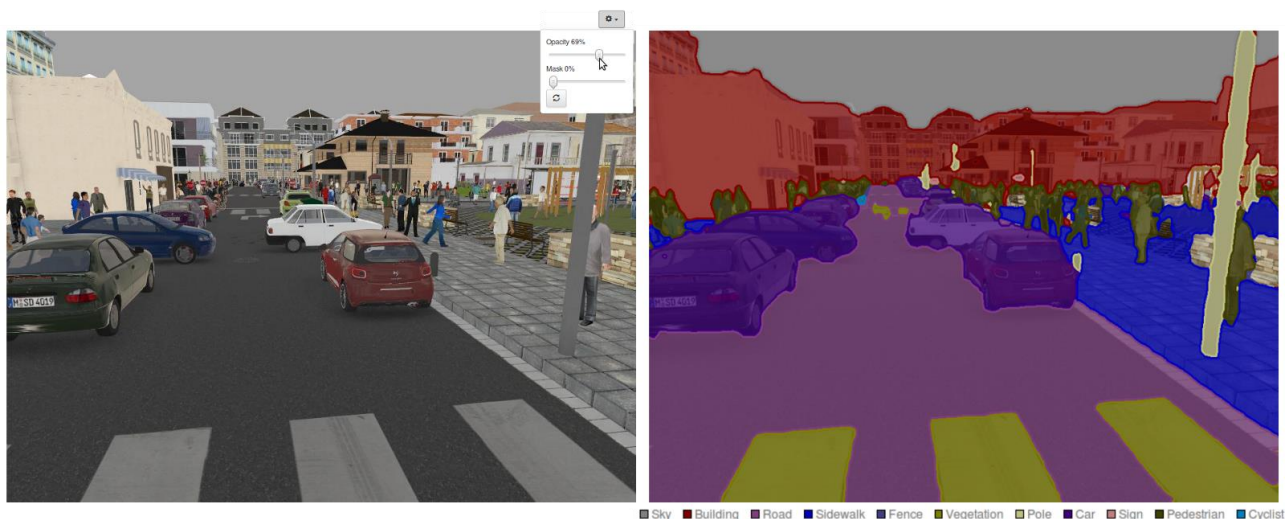
Een van de belangrijkste bewerkingen in de computer vision is segmentatie. Beeldsegmentatie is het clusteren van delen die tot het zelfde objectklasse behoren. Dit proces wordt soms ook wel classificatie op pixelniveau genoemd.

2.3.1 Semantic versus Instance Segmentation

Semantic segmentatie gaat classificeren op pixelniveau door te labelen uit een set met objecten categorieën bv. kat, hond en bomen.

Instance segmentatie gaat hetzelfde doen als bij semantic segmentatie, maar breidt nog verder uit door interessante objecten te detecteren en af te bakenen. [13]

Voor dit onderzoek werd semantic segmentatie gebruikt.



Figuur 5 – Semantic Segmentatie [13]

2.3.1.1 Dice coëfficiënt loss

In de praktijk wordt dikwijls de dice coëfficiënt loss gebruikt. Deze kijkt naar de overlap tussen twee data samples. In dit geval hoeveel de door de gebruikte model genereerde mask overlapt met de ground truth mask. [19]

2.3.2 Verschillende semantic segmentatie methodes

Er zijn een 20-tal verschillende segmentatie modellen die gebruikt kunnen worden om aan segmentatie te gaan doen. Een welbekende segmentatie model die vaak wordt gebruikt bij het segmenteren van bio medische afbeeldingen is het UNet methode. De UNet methode is in staat om de grenzen te lokaliseren en te onderscheiden en werkt op pixel niveau. De UNet werkt eveneens ook snel en vraagt weinig energie. [14]

Verschillende methodes:

- R-CNN (Regions with CNN feature)
- FCN (Fully Convolutional Network)
- UNet
- Deeplab
- ...

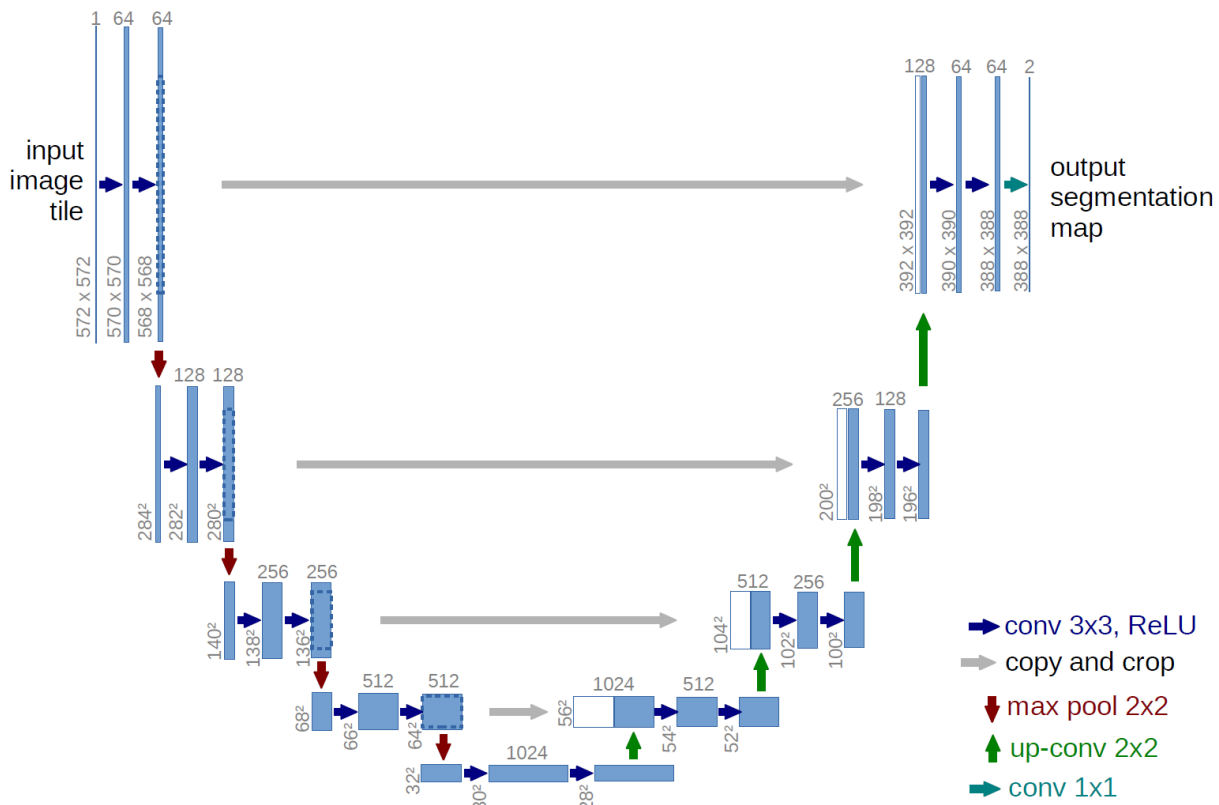
2.3.3 UNet

UNet is oorspronkelijk uitgevonden en voor het eerst gebruikt voor biomedische beeldsegmentatie. Het is een convolutionele netwerkarchitectuur voor een snelle en nauwkeurige segmentatie van verschillende objecten. Tot heden heeft deze methode beter gepresteerd dan eender welk andere methode. Het heeft de Cell Tracking Challenge op ISBI 2015 gewonnen voor de twee meest uitdagende categorieën van lichtmicroscopie (fasecontrast en DIC-microscopie). De UNet architectuur is simpel: een encoder en een decoder met skip connecties. [15]

UNet is in staat om afbeeldingen te lokaliseren door de afbeelding pixel voor pixel te voorspellen en de auteur van UNet beweert in zijn paper dat het netwerk sterk genoeg is om goede voorspellingen te doen op een kleine datasets door gebruik te maken data-augmentatietechnieken. [15]

De encoder is de eerste helft in het architectuurdiagram. Het is meestal een vooraf getraind classificatienetwerk zoals VGG/ResNet waar u convolutieblokken toepast, gevolgd door een maxpool-downsampling om het invoerbeeld te coderen in functierepresentaties op meerdere verschillende niveaus.

De decoder is de tweede helft van de architectuur. Het doel is om de onderscheidende kenmerken (lagere resolutie) die door de encoder zijn geleerd semantisch te projecteren op de pixelruimte (hogere resolutie) om een dichte classificatie te krijgen. De decoder bestaat uit upsampling en concatenatie gevolgd door regelmatige convolutiebewerkingen.



Figuur 6 - UNet [15]

2.3.4 Deeplabv3+

Deeplabv3 is een FCN en werkt met semantic segmentation architectuur die Deeplabv2 met verschillende aanpassingen verbetert. Er zijn modules ontworpen om de problemen van segmentatie aan te pakken. Door atrous-convolutie in cascade of parallel te gebruiken kan multi-schaalcontext vast leggen door meerdere atrous-snelheden toe te passen. Als ruggengraad gebruikt Deeplabv3 verschillende transfer learning modellen om de segmentatie objecten te gaan classificeren. [16]

Deeplabv3 wordt heel vaak ter sprake gebracht als het over segmentatie gaat op verschillende AI forums. Dit heeft zeker de potentie om een goede segmentatie te gaan doen van de tumor.

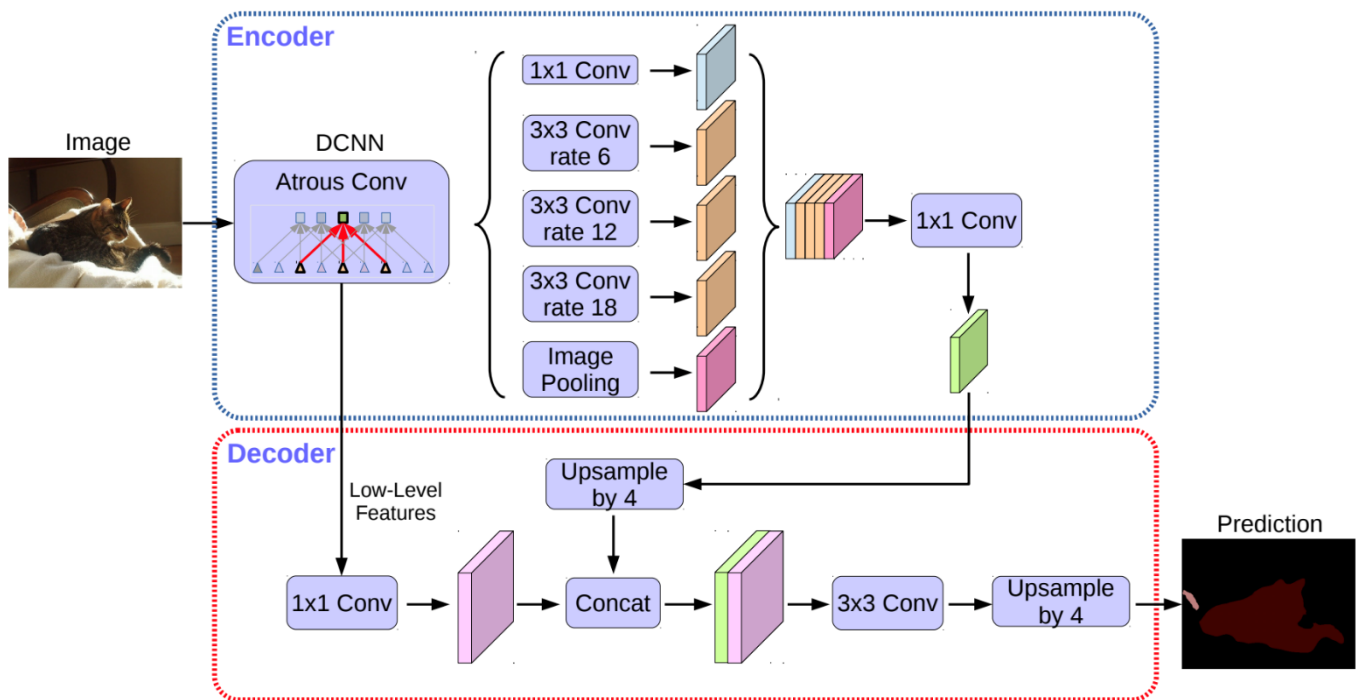
Een Deeplabv3+ werkt min of meer hetzelfde als een UNet model. Het enige grote verschil met UNet is dat er gebruik wordt gemaakt van Atrous Spatial Pyramid Pooling en dat er wordt gewerkt met een backbone model.

Een backbone model refereert naar de feature extraction gedeelte. De feature extraction gedeelte wordt gebruikt om de invoer van het netwerk te coderen in een bepaalde feature-representatie.

Voor mij was dit een onbekende architectuur en moest er heel veel research naar gedaan worden. Zelf was er niet veel te vinden over het gebruik ervan met python, meestal wordt dit model toegepast met Pytorch. Zelf heb ik ervoor gekozen om dit met Python op te zetten.

2.3.4.1 Atrous-convolutie

Atrous convolutie is een alternatief voor het downsamplen. Het vergroot het receptieve veld terwijl de ruimtelijke dimensie van de feature map behouden blijft. [16]



Figuur 7 - Deeplabv3+ [18]

2.3.5 Fully Convolutional Network (FCN)

Een FCN is een neurale netwerk dat alleen convolutiebewerkingen uitvoert. Een FCN is een CNN, maar zonder verbonden lagen. Dit wilt dus zeggen dat het geen gebruik maakt van "Dense" lagen, maar in de plaats een 1x1 convolutie gebruikt die fungeert als een dense laag.

Een convolutie laag is veel gespecialiseerder en efficiënter dan een fully connected laag. In een fully connected laag is elk neuron verbonden met elk neuron in de vorige laag, en elke verbinding heeft zijn eigen gewicht. Dit is een verbindingspatroon voor algemene doeleinden en doet geen aanname over de functies in de gegevens. Het is ook erg duur in termen van geheugen (gewichten) en berekening (verbindingen). Een voorbeeld van een FCN is een UNet methode. [17]

2.3.6 Ensemble learning

Ensemble learning wordt gebruikt om meerdere leeralgoritmen te combineren om betere voorspellingen te behalen. Deze methode werd toegepast om als doel de false positives en negatives te verminderen. Voor de ensemble model werden de drie beste modellen samengevoegd: VGG19, eigen CNN, eigen CNN met augmentatie.

Bij een ensemble learning zijn er 3 verschillende stemmogelijkheden:

- **Hard voting:** Voorspellingen zijn het gemiddelde van bijdragende modellen.
- **Soft voting:** Voorspellingen zijn de meerderheid van stemmen van bijdragende modellen.
- **Weighted voting:** Elk ensemblemodel heeft zijn unieke bijdrage die bepaald kan worden aan de weights die toegekend werd aan dat model

In dit onderzoek werd de **weighted voting** gebruikt, zo kon er een wat hogere weights toegekend worden aan de modellen die meer false positives voorspellingen doen.

3 Technisch onderzoek

3.1 Algemene informatie

Het doel van project vier was om te onderzoeken of het detecteren van hersentumoren geautomatiseerd kon worden doormiddel van AI. Als eerste stap werd er gewerkt aan het classificeren van de hersentumoren. Als tweede stap werd er gewerkt aan het segmenteren van de tumoren. En als laatste stap werd er een proof of concept applicatie gemaakt waarbij beide modellen samenwerken. De applicatie gaat eerste gaan voorspellen of er een tumor aanwezig is, indien er een tumor aanwezig is gaat de applicatie de hersentumor ook gaan segmenteren.

3.1.1 Programmeertaal en libraries

De gekozen programmeertaal waarin dit project is geschreven, is Python. De keuze voor de programmeertaal Python is deels een persoonlijke voorkeur. Daarnaast is Python ook een zeer gekende taal in verband met ML en DL. Python beschikt daarnaast ook over nuttige libraries voor dit project zoals bijvoorbeeld: tensorflow, pandas, opencv, etc.

Gebruikte libraries:

- Sklearn
- Opendcv
- Matplotlib
- Pandas
- Skimage
- Tensorflow
- Seaborn
- Numpy
- Scipy

3.1.2 Algoritmes

De gebruikte algoritmes zijn in de research gedeelte wat meer uitleg.

3.2 Algemeen resultaat

Het resultaat van project vier valt op te delen tussen het classificatie gedeelte en het segmentatie gedeelte. Beide gedeeltes hebben een eigen verantwoordelijkheid in de gemaakte applicatie. De onderzoeksvraag van deze bachelorproef is "Hoe kan het detecteren van hersentumoren geautomatiseerd worden?".

3.2.1 Classificatie gedeelte

Aan de hand van de uitgevoerde onderzoek kon er geconcludeerd worden dat het beste model voor het classificeren van de tumoren de *ensemble learning model* is. De ensemble learning model bestaat uit drie best presterende modellen: *VGG19*, *OwnClassificationNN* en *OwnClassificationNNAUG*. Deze drie modellen werken samen om het beste resultaat te behalen. Wat wel belangrijk is dat de modellen die gebruikt worden in een ensemble learning model wel degelijk goed moeten werken.

Er werd een accuraatheid score van 96.72% behaald. Verschillende modellen werden met elkaar vergeleken zowel eigen CNN modellen als transfer learning modellen. Dit om de drie beste modellen te kunnen selecteren voor een ensemble learning model. Zie technisch onderzoek gedeelte.

Dit model werd dan ook in het applicatie verwerkt.

3.2.2 Segmentatie gedeelte

Aan de hand van twee meest gebruikte segmentatie modellen voor medische afbeeldingen te trainen en met elkaar te vergelijken kon er geconcludeerd worden dat het UNet model op de dataset een klein stukje beter doet dan het Deeplabv3+ model.

Dit kon alleen maar geconcludeerd worden door te gaan onderzoeken hoe goed de modellen werken op wat grotere en kleinere tumoren. Er kon waargenomen worden dat beide modellen wel degelijk

goed werken op grotere tumoren. Bij een wat kleiner tumor heeft het deeplabv3+ model een klein stukje moeilijker dan het UNet model.

Het UNet model werd dan ook in de applicatie verwerkt.

3.2.3 Applicatie

De applicatie werd ontworpen om een proof of concept weer te geven. Alleen de beste modellen werden hierin verwerkt. De applicatie is een hulpmiddel voor de dokter en niet ontwikkeld als vervanger.

3.3 MRI scans classificatie

Als eerst werd er een simpele model(downwards CNN) gecreëerd om zo te kunnen zien hoe goed de dataset hierop werkt. Bleek dat het model wel degelijk goede voorspellingen kon maken, maar nog niet helemaal zoals het zou moeten zijn. De accuraatheid bedroeg een score van 82% en 55 verkeerde geclassificeerde hersenen.

Na het bekijken van de verkeerd geclassificeerde scans kon er geconcludeerd worden dat het model slechtere voorspelling doet op wat schuinere of kleinere hersenen. Dit kon geconcludeerd worden door de mis geclassificeerde hersenen te plotten. 90% van de mis geclassificeerde hersenen waren ofwel wat schuinere ofwel wat groter.

Om dit probleem op te lossen werd er gebruikt gemaakt van data augmentatie. De data augmentatie is uitgevoerd doormiddel van de python package ImageDataGenerator. Na dit uitgevoerd te hebben kon het model al veel beter overweg met wat schuinere of kleinere hersenen.

Vervolgens werd er gebruik gemaakt van verschillende transfer learning modellen: EfficientNetB0, InceptionV3, ResNet50 en VVG19. Om te modellen te optimaliseren werd er gebruikt gemaakt van hyperparameter tuning, dit kon de modellen met ongeveer met 30% verbeteren.

De gebruikte modellen werden met elkaar vergeleken. (zie scores hieronder)

Het beste model (de CNN met augmentatie toegepast) beschikt over een accuraatheid van 96,72% procent en een 97% F1-score voor de tumor klasse. Het aantal verkeerd geclassificeerde hersenen is 12.

3.3.1 Werken de modellen beter op grijswaardenafbeeldingen?

Op rgb afbeeldingen kon het model betere voorspellingen maken dan op de grijswaardenafbeeldingen. Dit werd geconcludeerd aan de hand van de scores die de modellen behaalden wanneer rgb afbeeldingen en grijswaardenafbeeldingen door hetzelfde model werden getraind. Er werd beslist om verder te gaan met rgb afbeeldingen.

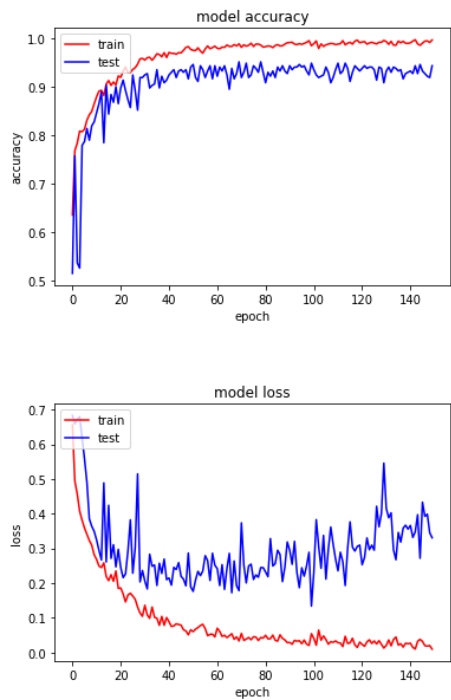
RGB	Grijs
95.81%	92.90%

Figuur 8 – RGB versus grijs score

3.3.2 Behaalde resultaten

Door allerlei hyperparameter tuning toe te passen konden de volgende scores behaald worden.

Eigen CNN

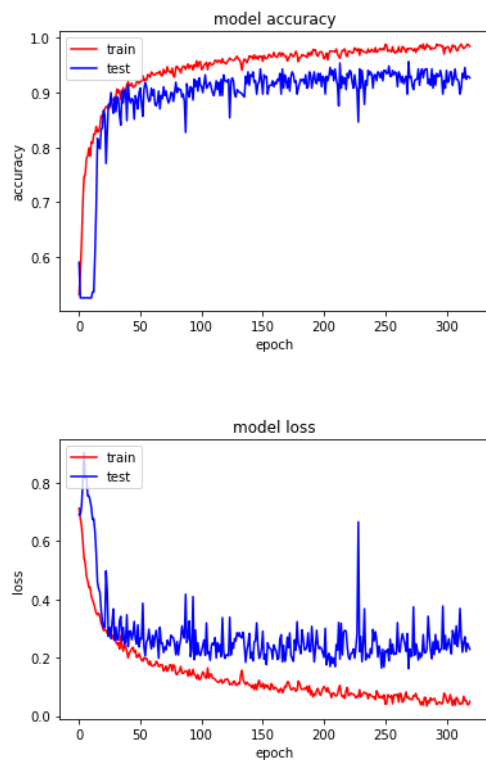


Figuur 9 - Eigen CNN Loss

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
Eigen CNN	92.90%	3	7	96%

Figuur 10 - Eigen CNN Metrics

Eigen CNN met data augmentatie

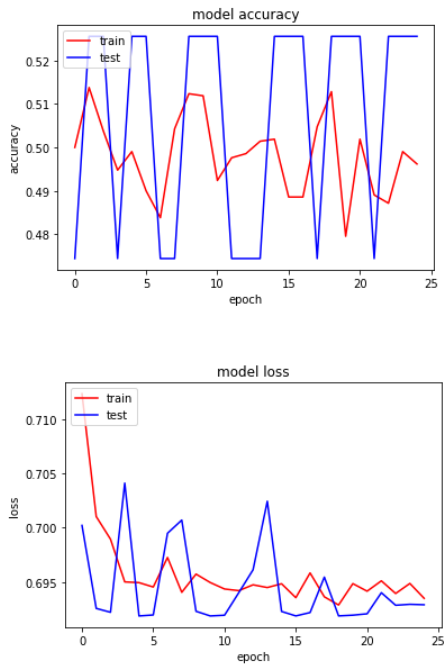


Figuur 11 - Eigen CNN met data augmentatie loss

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
Eigen CNN met data augmentatie	97.09%	6	2	97%

Figuur 12 - Eigen CNN met augementatie Metrics

Transfer learning: EfficientNetB0

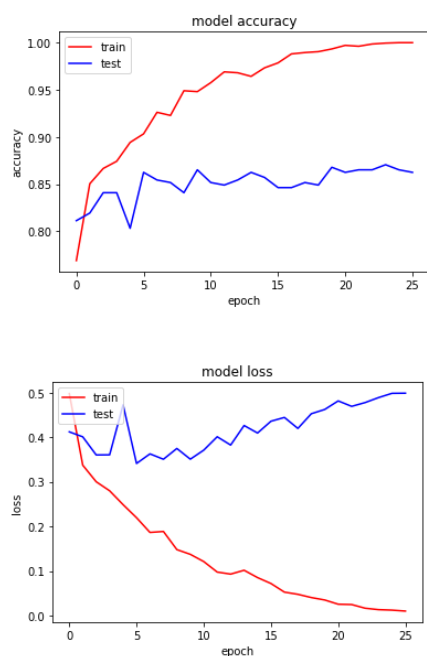


Figuur 13 - Transfer learning: EfficientNetB0 loss

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
EfficientNetB0	47.63%	144	0	65%

Figuur 14 - Transfer learning: EfficientNetB0 scores

Transfer learning: InceptionV3



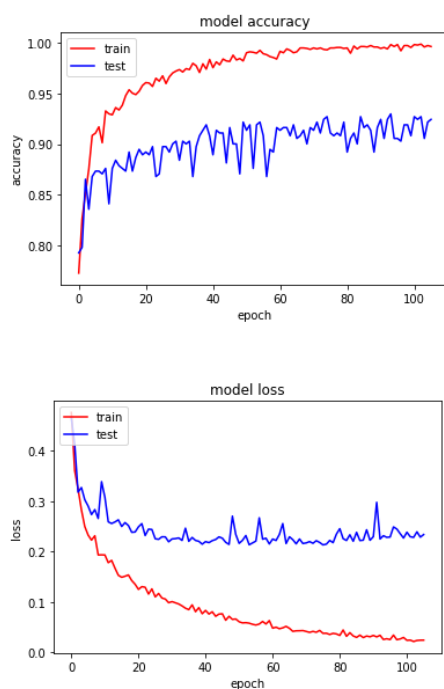
Figuur 15 - Transfer learning: InceptionV3 loss

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
Inceptionv3	92.0%	9	13	91%

Figuur 16 - Transfer learning: InceptionV3 scores

Transfer learning: ResNet50

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
ResNet50	80.36%	27	27	79%

*Figuur 17 - Transfer learning: ResNet50 scores***Transfer learning: VGG19***Figuur 18 - Transfer learning: VGG19 loss*

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
VGG19	94.90%	6	8	95%

Figuur 19 - Transfer learning: VGG19 scores

3.3.3 Eigen CNN modellen vs Transfer learning modellen

Na het aanmaken van de modellen en die getraind te hebben werden de modellen vergeleken op basis van de accuraatheid, f1-score van de tumor klasse en de recall.

3.3.3.1 Vergelijking van de 3 beste modellen

Model	Accuracy	FalsePos	FalseNeg	F1-Score Tumor
CNN	95.27%	4	8	95%
CNN with Aug	97.09%	6	2	97%
VGG19	94.90%	6	8	95%

Figuur 20 – Vergelijking beste classificatie modellen

3.3.3.2 Keuze model:

Eigen classificatie model met augmentatie toegepast.

Motivatie: Mijn eigen kleine classificatie model met augmentatie toegepast doet het toch een stukje beter dan de tranfer learning modellen. (zie f1-scores, recall en acc)

3.3.3.3 Waarom is de f1-score van de tumor klasse een belangrijke score:

Het is heel cruciaal dat het model meer false positives (valse alarm van tumor) predicties doet, dan false negatives (tumor aanwezig, maar niet voorspeld worden). Dit is omdat het een leven kan redden.

3.3.3.4 Waarom kan een tranfer learning slechter presteren?:

Verschillende factoren kunnen hier een rol inspelen. Zoals de image-size waarop dat het model getraind werd niet hetzelfde is met de gebruikte image-size, de pre-processing van data, snelle overfitting, te groot model voor deze dataset, ongezien images (imageNet).

3.3.3.5 Verminderen van false positives and negatives?

Hiervoor werden de 3 beste modellen samengevoegd, zodat ze elkaar steunen om betere voorspellingen te kunnen maken. (ensemble model)

3.3.4 Beste resultaat

Het beste model (eigen CNN model met augmentatie) behaalde een accuracy score van maar liefst 97% en 97% F1-score voor de tumor klasse. Door gebruik te maken van ensemble learning kon de accuracy score verhoogd worden met 1%. 1% lijkt weinig, maar wanneer het gaat om veel data dan is 1% wel veel.

```
accuracy of own model with augmentation: 96.36363636363636
accuracy of own model with augmentation: 97.0909090909091
accuracy of vgg19: 94.9090909090909
accuracy of ensemble model: 98.18181818181819
```

Figuur 21 - beste resultaten

De best presteerde model is de ensemble learning model die een accuraatheid van maar liefst 98% behaalt. Dit model werd dan ook in de applicatie verwerkt.

Een moeilijkheid die moest overwonnen worden was het samenvoegen van de modellen tot een correct werkende ensemble model. Dit houdt in dat het juiste aantal gewicht toegekend moet worden aan ieder model. Dit werd opgelost door gebruik te maken van gridsearch voor de beste combinatie van de gewichten.

	wt1	wt2	wt3	acc
0	0.0	0.0	0.0	52.363636
0	0.0	0.0	0.1	97.090909
0	0.0	0.0	0.2	97.090909
0	0.0	0.0	0.3	97.090909
0	0.0	0.0	0.4	97.090909
...
0	0.3	0.4	0.0	96.727273
0	0.3	0.4	0.1	97.454545
0	0.3	0.4	0.2	97.454545
0	0.3	0.4	0.3	97.818182
0	0.3	0.4	0.4	98.181818

Figuur 22 - moeilijkheid ensemble model

3.4 Segmentatie van de tumor

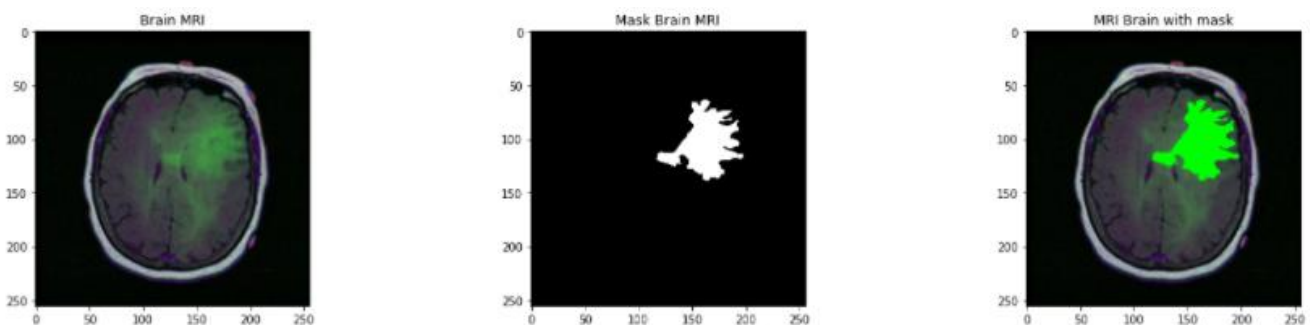
Het doel van de segmentatie gedeelte is om een model te creëren die de hersentumoren zo nauwkeurig mogelijk gaat segmenteren.

Om dit te realiseren werd in dit onderzoek gebruik gemaakt van de UNet en de DeeplabV3+ architectuur. Ook werden de 2 verschillende architecturen vergeleken met elkaar om zo het beste model te kunnen gebruiken al segmentatie model in de applicatie.

3.4.1 Data voorbereiding

Allereerst moest er een target aangemaakt worden, dit werd bereikt door de masker op de MRI scan te plaatsen.

Voorbeeld:



Figuur 23 - pre-processed data

De *Brain MRI* werd als input meegegeven in de modellen en de *MRI Brain with mask* werd als target meegegeven.

De afbeeldingen werden ook allemaal geschaald naar 128x128px zoals eerder vermeld.

3.4.2 UNet

3.4.2.1 Model

input_1 (InputLayer)	[(None, 128, 128, 3)]	[]
lambda (Lambda)	(None, 128, 128, 3) 0	['input_1[0][0]']
conv2d (Conv2D)	(None, 128, 128, 16 448)	['lambda[0][0]']
dropout (Dropout)	(None, 128, 128, 16 0)	['conv2d[0][0]']
conv2d_1 (Conv2D)	(None, 128, 128, 16 2320)	['dropout[0][0]']
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16) 0	['conv2d_1[0][0]']
conv2d_2 (Conv2D)	(None, 64, 64, 32) 4640	['max_pooling2d[0][0]']
dropout_1 (Dropout)	(None, 64, 64, 32) 0	['conv2d_2[0][0]']
conv2d_3 (Conv2D)	(None, 64, 64, 32) 9248	['dropout_1[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32) 0	['conv2d_3[0][0]']
conv2d_4 (Conv2D)	(None, 32, 32, 64) 18496	['max_pooling2d_1[0][0]']
dropout_2 (Dropout)	(None, 32, 32, 64) 0	['conv2d_4[0][0]']
conv2d_5 (Conv2D)	(None, 32, 32, 64) 36928	['dropout_2[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64) 0	['conv2d_5[0][0]']
conv2d_6 (Conv2D)	(None, 16, 16, 128) 73856	['max_pooling2d_2[0][0]']
dropout_3 (Dropout)	(None, 16, 16, 128) 0	['conv2d_6[0][0]']
conv2d_7 (Conv2D)	(None, 16, 16, 128) 147584	['dropout_3[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 128) 0	['conv2d_7[0][0]']
conv2d_8 (Conv2D)	(None, 8, 8, 256) 295168	['max_pooling2d_3[0][0]']
dropout_4 (Dropout)	(None, 8, 8, 256) 0	['conv2d_8[0][0]']
conv2d_9 (Conv2D)	(None, 8, 8, 256) 590080	['dropout_4[0][0]']
conv2d_transpose (Conv2DTransp ose)	(None, 16, 16, 128) 131200	['conv2d_9[0][0]']
concatenate (Concatenate)	(None, 16, 16, 256) 0	['conv2d_transpose[0][0]', 'conv2d_7[0][0]']
conv2d_10 (Conv2D)	(None, 16, 16, 128) 295040	['concatenate[0][0]']
dropout_5 (Dropout)	(None, 16, 16, 128) 0	['conv2d_10[0][0]']
conv2d_11 (Conv2D)	(None, 16, 16, 128) 147584	['dropout_5[0][0]']
conv2d_transpose_1 (Conv2DTran spose)	(None, 32, 32, 64) 32832	['conv2d_11[0][0]']
concatenate_1 (Concatenate)	(None, 32, 32, 128) 0	['conv2d_transpose_1[0][0]', 'conv2d_9[0][0]']
conv2d_12 (Conv2D)	(None, 32, 32, 64) 73792	['concatenate_1[0][0]']
dropout_6 (Dropout)	(None, 32, 32, 64) 0	['conv2d_12[0][0]']
conv2d_13 (Conv2D)	(None, 32, 32, 64) 36928	['dropout_6[0][0]']
conv2d_transpose_2 (Conv2DTran spose)	(None, 64, 64, 32) 8224	['conv2d_13[0][0]']
concatenate_2 (Concatenate)	(None, 64, 64, 64) 0	['conv2d_transpose_2[0][0]', 'conv2d_10[0][0]']
conv2d_14 (Conv2D)	(None, 64, 64, 32) 18464	['concatenate_2[0][0]']
dropout_7 (Dropout)	(None, 64, 64, 32) 0	['conv2d_14[0][0]']
conv2d_15 (Conv2D)	(None, 64, 64, 32) 9248	['dropout_7[0][0]']
conv2d_transpose_3 (Conv2DTran spose)	(None, 128, 128, 16 2064)	['conv2d_15[0][0]']
concatenate_3 (Concatenate)	(None, 128, 128, 32 0)	['conv2d_transpose_3[0][0]', 'conv2d_1[0][0]']
conv2d_16 (Conv2D)	(None, 128, 128, 16 4624)	['concatenate_3[0][0]']
dropout_8 (Dropout)	(None, 128, 128, 16 0)	['conv2d_16[0][0]']
conv2d_17 (Conv2D)	(None, 128, 128, 16 2320)	['dropout_8[0][0]']

Figuur 24 - UNET model code

```

inputs = Input((128, 128, 3))
s = Lambda(lambda x: x / 255)(inputs)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(s)
c1 = Dropout(0.1)(c1)
c1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c1)
p1 = MaxPooling2D((2, 2))(c1)

c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(p1)
c2 = Dropout(0.1)(c2)
c2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c2)
p2 = MaxPooling2D((2, 2))(c2)

c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(p2)
c3 = Dropout(0.2)(c3)
c3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c3)
p3 = MaxPooling2D((2, 2))(c3)

c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(p3)
c4 = Dropout(0.2)(c4)
c4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c4)
p4 = MaxPooling2D(pool_size=(2, 2))(c4)

c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(p4)
c5 = Dropout(0.3)(c5)
c5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c5)

u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(u6)
c6 = Dropout(0.2)(c6)
c6 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c6)

u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(u7)
c7 = Dropout(0.2)(c7)
c7 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c7)

u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same')(c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(u8)
c8 = Dropout(0.1)(c8)
c8 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c8)

u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same')(c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(u9)
c9 = Dropout(0.1)(c9)
c9 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same')(c9)

```

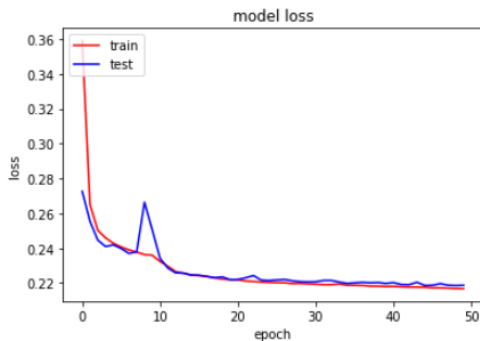
Figuur 25 -- UNET model code

3.4.2.2 Resultaat

Naast het opzetten van het UNet model moest er ook aan hyperparameter tuning gedaan worden om het model te optimaliseren. Al snel kon het UNet model al goede scores halen.

Behaalde scores:

- Dice_coef_loss: 0.0022



Figuur 26 - UNET loss

3.4.3 Deeplabv3+

3.4.3.1 Model

```
""" Strous Spatial Pyramid Pooling """
def ASPP(inputs):
    shape = inputs.shape

    y_pool = AveragePooling2D(pool_size=(shape[1], shape[2]), name='average_pooling')(inputs)
    y_pool = Conv2D(filters=128, kernel_size=1, padding='same', use_bias=False)(y_pool)
    y_pool = BatchNormalization(name='bn_1')(y_pool)
    y_pool = Dropout(0.1)(y_pool)
    y_pool = Activation('relu', name='relu_1')(y_pool)
    y_pool = Upsampling2D((shape[1], shape[2]), interpolation='bilinear')(y_pool)

    y_1 = Conv2D(filters=128, kernel_size=1, dilation_rate=1, padding='same', use_bias=False)(inputs)
    y_1 = BatchNormalization()(y_1)
    y_1 = Dropout(0.1)(y_1)
    y_1 = Activation('relu')(y_1)

    y_6 = Conv2D(filters=128, kernel_size=3, dilation_rate=6, padding='same', use_bias=False)(inputs)
    y_6 = BatchNormalization()(y_6)
    y_6 = Dropout(0.1)(y_6)
    y_6 = Activation('relu')(y_6)

    y_12 = Conv2D(filters=128, kernel_size=3, dilation_rate=12, padding='same', use_bias=False)(inputs)
    y_12 = BatchNormalization()(y_12)
    y_12 = Dropout(0.1)(y_12)
    y_12 = Activation('relu')(y_12)

    y_18 = Conv2D(filters=128, kernel_size=3, dilation_rate=18, padding='same', use_bias=False)(inputs)
    y_18 = BatchNormalization()(y_18)
    y_18 = Dropout(0.2)(y_18)
    y_18 = Activation('relu')(y_18)

    y = Concatenate()([y_pool, y_1, y_6, y_12, y_18])

    y = Conv2D(filters=128, kernel_size=1, dilation_rate=1, padding='same', use_bias=False)(y)
    y = BatchNormalization()(y)
    y = Activation('relu')(y)
    return y

def Deeplabv3Plus(shape):
    """ Inputs """
    inputs = Input(shape)

    """ Pre-trained ResNet50 """
    base_model = ResNet50(weights='imagenet', include_top=False, input_tensor=inputs)

    """ Pre-trained ResNet50 Output """
    image_features = base_model.get_layer('conv4_block6_out').output
    x_a = ASPP(image_features)
    x_a = Upsampling2D((4, 4), interpolation='bilinear')(x_a)

    """ Get low-level features """
    x_b = base_model.get_layer('conv2_block2_out').output
    x_b = Conv2D(filters=32, kernel_size=1, padding='same', use_bias=False)(x_b)
    x_b = BatchNormalization()(x_b)
    x_b = Activation('relu')(x_b)

    x = Concatenate()([x_a, x_b])

    x = Conv2D(filters=128, kernel_size=3, padding='same', activation='relu', use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    x = Conv2D(filters=128, kernel_size=3, padding='same', activation='relu', use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Upsampling2D((4, 4), interpolation='bilinear')(x)

    """ Outputs """
    x = Conv2D(3, (1, 1), name='output_layer')(x)
    x = Activation('sigmoid')(x)

    """ Model """
    Deeplabv3Plus1 = Model(inputs=inputs, outputs=x)
    return Deeplabv3Plus1
```

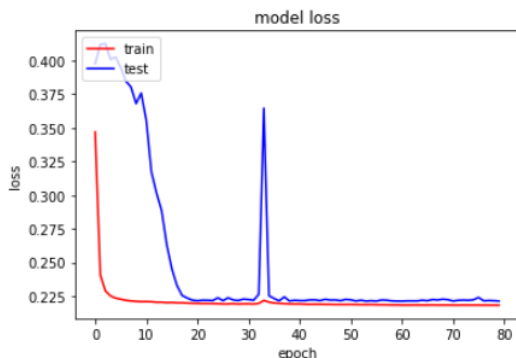
Figuur 27 - Deeplabv3+ model code

3.4.3.2 Resultaat

Naast het opzetten van het Deeplabv3+ model moest er ook een backbone model toegekend worden dit werd de ResNet50. Ook moest er aan hyperparameter tuning gedaan worden om het model te optimaliseren.

Behaalde scores:

- Dice_coef_loss: 0.0042



Figuur 28 - Deeplabv3+ loss

3.4.4 Model vergelijken

Na het trainen en het optimaliseren van de modellen werden de modellen vergeleken op basis van trainingssnelheid, output, prestatie op kleinere tumoren en dice_coef.

Belangrijke scores:

Model	Loss	Dice_Coef	Dice_Coef_Loss
UNet	0.2250	0.9976	0.0026
Deeplabv3+	0.2264	0.9957	0.0041

Figuur 29 - Segmentatie Model vergelijken

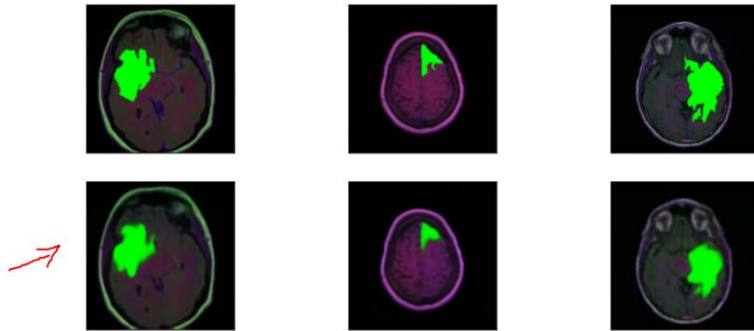
3.4.4.1 UNet resultaten

```
ef_loss: 0.0025
Epoch 00153: early stopping
Wall time: 4min 11s
```

Figuur 30 – Unet resultaat epoch

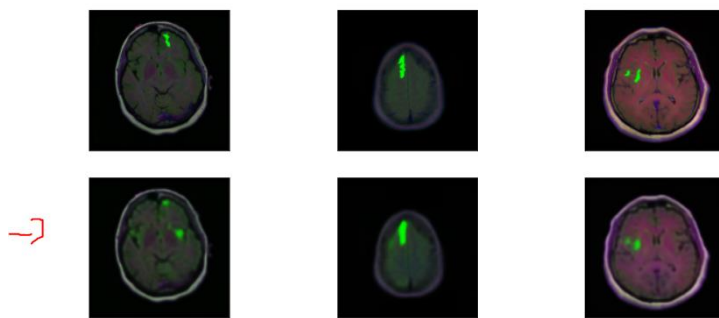
≡ ong. 1.6sec per epoch

Grote tumoren:



Figuur 31 - UNET resultaat easy part

kleine tumoren:



Figuur 32 - UNET resultaat hard part

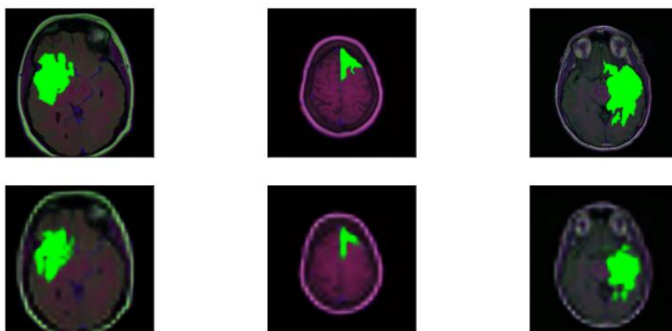
3.4.4.2 Deeplabv3+ resultaten

```
oef_loss: 0.0044
Epoch 00116: early stopping
Wall time: 11min 51s
```

Figuur 33 - Deeplabv3+ resultaat epoch

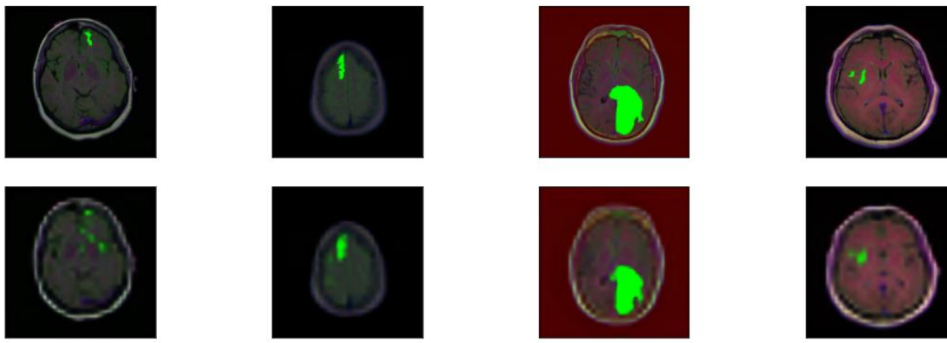
ong. 6 sec per epoch

Grote tumoren:



Figuur 34 - - Deeplabv3+ resultaat easy part

Kleine tumoren:



Figuur 35 - Deeplabv3+ resultaat hard part

3.4.4.3 Keuze model

Als de dice_coef_loss score vergeleken wordt, dan scoort Unet daar en stukje beter op dan deeplabv3+. Unet doet het bijna 5X sneller per epoch om te trainen.

De UNet model werd dan ook in de applicatie verwerkt voor de segmentatie gedeelte.

Easy Part: Wat grotere tumoren, makkelijker te segmenteren.

Hard Part: Veel kleinere tumoren, veel moeilijker dit correct te segmenteren.

3.4.4.4 Waarom behaalde Deeplabv3+ mindere scores

Het probleem hierbij is niet het model, maar de pre-trainend model ResNet50 die gebruikt werd als backbone. Als er terug gekeken wordt naar de classificatie gedeelte kan er waargenomen worden dat de pre-trained (resnet50) model het niet echt goed doet. Dit betekent dus dat de pre-trainend ResNet50 model voor dit dataset niet geschikt is en dat het verstandiger is om een andere pre-trained model te gebruiken als backbone zoals de VGG19. Door tijdgebrek kon ik dit niet concreet uittesten.

3.5 Proof of concept applicatie

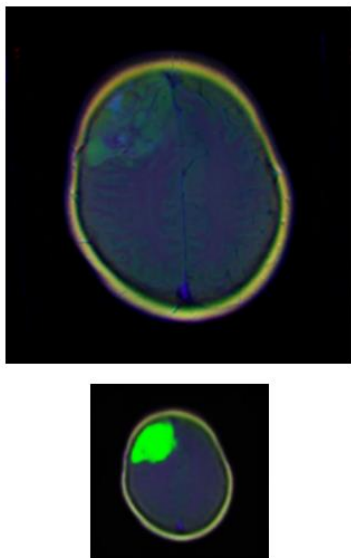
Het doel van de proof of concept applicatie is eigenlijk om te kunnen waarnemen of het idee wel omgezet kan worden naar werkelijkheid.

3.5.1 Werking

Een lijst van MRI scans die eerst door de classificatie gedeelte doorgestuurd worden, indien het model een tumor gedetecteerd heeft zal het ook door het segmentatie model gestuurd worden.

Dit applicatie is gemaakt als hulpmiddel voor de dokter en niet als vervanger.

Voorbeeld:



Figuur 36 - Proof of concept voorbeeld

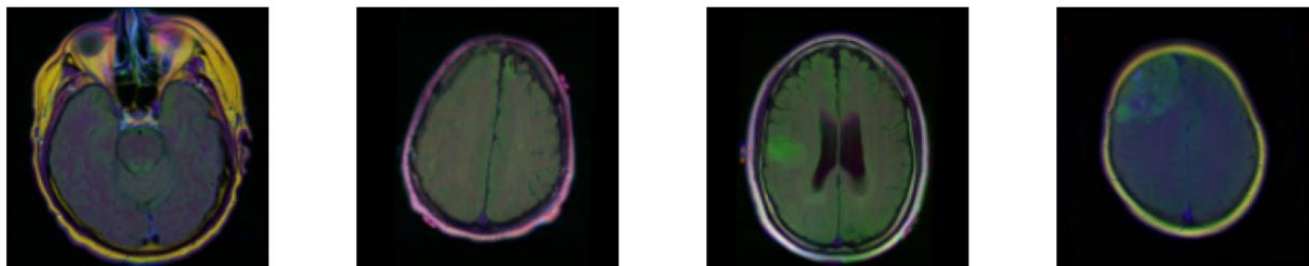
3.5.2 Gebruikte modellen

Voor de classificatie gedeelte werd de *Eigen classificatie model met augmentatie toegepast* gebruikt. Voor de segmentatie gedeelte werd de *Ensemble learning model* gebruikt.

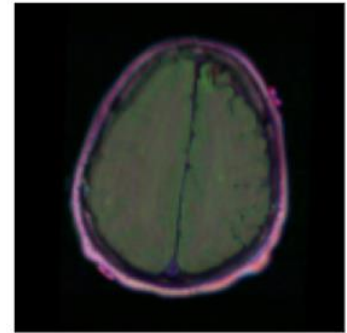
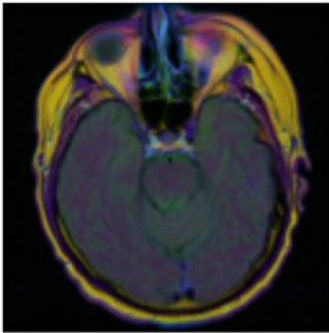
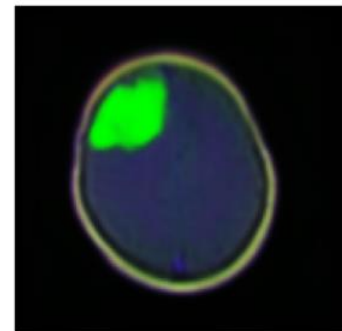
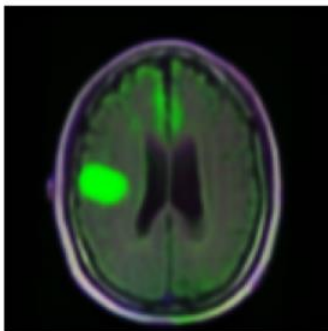
3.5.3 Output

Hierbij een voorbeeld van de werking en output van de applicatie.

Ingegeven input:



Figuur 37 - proof of concept input

Output:**List - No tumor detected***Figuur 38 - proof of concept output 1***List - tumor detected***Figuur 39 - proof of concept output 2*

3.6 Moeilijkheden

Tijdens het werken aan project vier zijn er een paar moeilijkheden overwonnen. Als eerste moeilijkheid was het verlagen van de false negatives/positives na het uitvoeren van hyperparameter tuning bij de classificatie gedeelte. Dit probleem werd opgelost door de hulp van de community stackoverflow. Als oplossing werd voorgesteld om de beste modellen samen te voegen tot een ensemble learning model.

Als tweede moeilijkheid die overwonnen werd was het ontwikkelen van een deeplabv3+ model in Python. Deeplabv3+ is meestal geprogrammeerd met Pytorch, dit was moeilijker over te schakelen dan gedacht. Uiteindelijk is dit wel gelukt en werd ResNet50 als backbone model gebruikt. Na het toepassen van de hyperparameter tuning kon geconcludeerd worden dat er beter een andere pre-trainend model gebruikt moest worden. Dit kon jammer genoeg niet getest worden.

4 Reflectie

Dit onderdeel werd besproken met Robbert Verbeke, werkend als data scientist bij Sirris. Hij beschikt over de volgende diploma's: een master diploma in de wiskunde, doctoraat in de Astrofysica en een postdoctoraat in de research. Hierbij was hij de beste keus om dit onderdeel mee te bespreken.

4.1 Zelfreflectie

4.1.1 Wat ging er beter dan verwacht

Door een stevige basis aangeleerd te hebben aan de Howest vond dat ik het modelleren snel en op een correcte wijze kon uitvoeren. Ook het onderzoeken van nieuwe algoritmes kon ik snel verwerken en toepassen.

De behaalde accuraatheden liggen redelijk hoog, door de doorzettingsvermogen die ik had doorheen dit project. Door dit project heb ik mezelf leren motiveren doorheen moeilijke periodes.

4.1.2 Wat ging er minder dan verwacht

Wat minder goed ging is de data analyse, ik ben te snel over dit stukje gelopen. Achteraf gezien is dit wel het meest cruciale stukje dat niet overgeslagen zou moeten worden. Het probleem waarom ik dit niet heb gedaan, is omdat ik dit zo een beetje aangeleerd heb gekregen. Meestal kregen we op school een "perfecte dataset" waarbij niet veel tijd gependend moest worden aan de data analyse.

Doordat ik samen met mijn stagebegeleider dit project heb overgelopen had hij al meteen een fout gelabelde data ontdekt, dit wilt dus zeggen dat er meerdere hersenen verkeerd gelabeld zouden kunnen zijn. De verkeerd gelabelde MRI scan was een hersen die geen tumor bevatte, maar wel gelabeld was als tumor. Wanneer dit het geval zou zijn bij meerdere scans, zou dit kunnen verklaren waarom mijn model false positives voorspelde.

4.1.3 Wat zou ik anders doen?

De dataset die gebruikt werd bevat jammer genoeg maar één soort tumor(glioom), waardoor dit een beetje moeilijker in de praktijk gebruikt kan worden. Indien ik dit opnieuw zou mogen doen, zou ik vooral meer tijd spenderen aan het selecteren van een dataset met verschillende tumoren en bruikbaar data.

4.2 Reflectie van het resultaat met externe

4.2.1 Sterke punten

Er werd goed gebruik gemaakt van zowel eigen kleine modellen als transfer learning modellen. In het onderzoek zijn verschillende modellen getest geweest voor zowel classificatie als segmentatie. Degene met goeie performance werden gebruikt in de applicatie. In het geval van classificatie, werden meerder modellen gecombineerd in een ensemble methode om de accuracy nog verder te verbeteren.

In het onderzoek werden hoge accuracy scores behaald, die kunnen aantonen dat het automatiseren van het detecteren van hersentumoren effectief wel haalbaar is.

4.2.2 Zwakke punten

Voor transfer learning bij de classificatie hebben sommige pre-trained models zoals EfficientNetB0 een slechte accuracy. Enkele van de latere layers als trainable zetten zou ervoor kunnen zorgen dat het beter features kan leren voor het probleem. Dit geldt ook voor VGG19, die wel een goeie accuracy behaalt.

Een analyse van de ROC curves van de modellen zou kunnen helpen bij een betere threshold voor het classificeren in de verschillende klassen.

Sommige modellen lijken overfitted. Een kleiner model (of in geval van transfer learning, kleinere trainable layers) of een hogere dropout rate kunnen hier helpen.

4.2.3 Wat zou er beter gekund hebben

Een error analyse van de samples die fout voorspeld worden is altijd een goeie stap. Zeker van de samples die door alle modellen fout voorspeld worden. Dit zou inzicht kunnen geven in waarom het verkeerd voorspeld wordt of een mislabeling kunnen aantonen. Bvb in cel 14 (met de functie `show_image`) worden enkel voorbeelden met tumors getoond, maar 2 van de voorbeelden hebben geen, of een zeer kleine mask.

4.3 Bruikbaarheid

Dag van vandaag gebeurt de detectie van hersentumoren handmatig door een neuroloog. Dit komt voornamelijk omdat men geen vertrouwen heeft in machine learning algoritmes, zeker als het gaat over medische applicaties.

De technologie is zeker in staat om dit op automatische wijze te gaan detecteren zie behaalde resultaten, maar doordat machine learning over een black box beschikt nemen dokters daar afstand van. [20]

4.3.1 Implementatie hindernissen

"Veel clinici blijven op hun hoede voor machinaal leren vanwege langdurige zorgen over 'black box'-modellen. "Black box" is een afkorting voor modellen die zo complex zijn dat ze niet eenvoudig te interpreteren zijn voor mensen. Gebrek aan interpreteerbaarheid in voorspellende modellen kan het vertrouwen in die modellen ondermijnen, vooral in de gezondheidszorg, waarin zoveel beslissingen – letterlijk – kwesties van leven en dood zijn. Er is recentelijk een explosie van onderzoek geweest op het gebied van verklaarbare machine learning om deze zorgen weg te nemen." [20]

Als 2^{de} hindernis die overwonnen zou moeten worden is dat er voldoende data aanwezig zou moeten zijn en die bovendien correct gelabeld zijn. In het bedrijfswereld zijn de masks niet altijd voorhanden en zou bv. een eigen methode gecreëerd moeten worden om te gaan labelen.

4.3.2 Implementatie

De gemaakte applicatie zou in samenwerking met een expert gebruikt kunnen worden, die de resultaten evalueert. Dit op te merken dat de gemaakte applicatie geen vervanger probeert te worden maar een tool.

4.4 Meerwaarde

4.4.1 Praktisch/maatschappelijk relevant?

In voorgaande stukken werd aangehaald dat een misdiagnoses bij tumoren een hedendaags probleem is. De oorzaak hiervan is vaak dat het om een kleine tumor gaat waardoor de neuroloog dit over het hoofd ziet.

Als dit onderzoek in de werkelijkheid gebruikt zou worden zouden er heel veel levens gered kunnen worden van een misdiagnose. Dit zal een maatschappelijk inpakt hebben en het stellen van een vroegtijdige diagnose vergemakkelijken.

4.5 Vervolg onderzoek

4.5.1 Algoritmes

Dit onderzoek maakte al gebruik van verschillende soorten classificatie en segmentatie algoritmes, de voornaamste suggestie zou zijn om meerdere segmentatie modellen op te zetten en te gaan testen op de dataset net als dat gedaan werd bij de classificatie gedeelte. Dit zou kunnen leiden tot een betere resultaat en een wat uitgebreider vergelijking.

4.5.2 Masks

Zoals eerder aangekaart te hebben in de te overwinnen hindernissen kon gelezen worden dat de masks van tumoren niet altijd voorhanden ligt en dat dit zelf gecreëerd zou moeten worden. Dit zou opgelost kunnen worden door gebruik te maken van region based segmentation of edge detection segmentation op de scans.

4.5.3 Ernst van de diagnose

Een volgende mogelijke stap is om de “ernst” van de diagnose te gaan voorspellen en de beste behandeling (bvb. chemotherapie, operatie,...). Hoewel de ernst geschat zou kunnen worden aan de hand van de grootte van de mask, zouden voor beide vragen idealiter extra target labels voor handen zijn.

5 Advies

Op de dag van vandaag worden de diagnoses van hersentumoren handmatig geteld door een neuroloog. Als het gaat over het gebruik van artificiële intelligentie in de gezondheidszorg, dan kan er geteld worden op twee handen hoeveel Ai toepassingen er daadwerkelijk worden gebruikt. Dit heeft voornamelijk te maken met dat Ai over verschillende beperkingen beschikt en verschillende problemen met zich meebrengt. De problemen komen later in dit hoofdstuk aanbod.

5.1 Bruikbaarheid

In dit onderdeel zal het gaan over de bruikbaarheid van Ai toepassingen die op dag van vandaag gebruikt worden in de gezondheidszorg. De pros en cons van Ai zal worden toegelicht en ook zal uitgelegd worden hoe mijn onderzoek in toepassing gebruikt kan worden. [21][22]

Door de verschillende hedendaagse Ai applicaties te gaan bekijken, zou afgeleid kunnen worden als dit project wel in de werkelijkheid bruikbaar zal zijn.

5.1.1 top 5 van de 10 Ai toepassingen in de gezondheidszorg

Het steeds groeiende aantal toepassingen die machine learning met zich mee brengt voor de gezondheidszorg geeft een glimp van de toekomst. Jammer genoeg wordt op dag van vandaag ML maar voor een beperkt aantal toepassingen gebruikt.

5.1.1.1 1) Klinisch onderzoek

Klinisch onderzoeken en proeven zijn waardevol en langdurige processen. Doordat nieuwe medicijnen en medische procedures eerst veilig verklaard moeten worden voordat ze op grote schaal gebruikt mogen worden.

Gelukkig is er een manier gevonden om het proces te verkorten met behulp van Ai. Het kan worden gebruikt om de beste steekproef voor de proef te bepalen om zo een zeer representatief beeld te creëren, meer gegevenspunten te verzamelen, de lopende gegevens van de proefdeelnemers te analyseren en foute gegevens te verminderen.

5.1.1.2 2) Voorspelling uitbraak infectieziektes

De covid-19-pandemie heeft ons laten zien hoe onvoorbereid we waren op een uitbraak van een infectieziekte. Machine learning beschikt over de nodige tools om de tekenen van een epidemie in een vroeg stadium op te sporen. Machine learning werd ook gebruikt om de mate van verspreiding te gaan voorspellen en op basis van die cijfers werden er maatregelen genomen.

5.1.1.3 3) Gegevens verzameling

Een van de belangrijkste verantwoordelijkheden van een arts is het bijhouden van gegevens van een patiënt. Dit kan vaak een uitdaging zijn, omdat de patiënt geen specialist is en niet weet welke gegevens relevant zijn om openbaar te maken.

Doormiddel van ML in zorgmanagement kunnen zorgprofessionals op basis van verschillende indicatoren bepalen wat de meest relevante vragen zijn die ze aan een patiënt moeten stellen. Dit zal helpen om relevante gegevens te verzamelen en tegelijkertijd een voorspelling te krijgen van de meest waarschijnlijke omstandigheden.

5.1.1.4 4) Klinische beslissingsondersteunende systemen

ML helpen ook bij klinische besluiten en helpen bij het analyseren van grote hoeveelheid data om een ziekte te identificeren, keuze voor de volgende behandelingsfase te voorspellen en eventuele problemen vast te stellen. CDSS is een krachtig hulpmiddel dat de arts helpt zijn werk efficiënt en snel te doen, en het verkleint de kans op het krijgen van de verkeerde diagnose of het voorschrijven van een ineffectieve behandeling.

5.1.1.5 5) Slimme administratie

Een van de toepassingen van machine learning in de gezondheidszorg is het gebruik van optische tekenherkenningstechnologie (OCR) op het handschrift van artsen, waardoor gegevens snel en naadloos kunnen worden ingevoerd. Deze gegevens kunnen vervolgens worden geanalyseerd door andere machine learning-tools om de besluitvorming en patiëntenzorg te verbeteren.

5.1.2 Pros en cons van AI in de gezondheidszorg

5.1.2.1 Pros

1. Zorgt voor real-time data
2. Data streams
3. Bespaart tijd en middelen
4. Helpt bij onderzoek
5. Kan de stress van de arts verminderen

5.1.2.2 Cons

1. Heeft menselijk toezicht nodig
2. Houdt geen rekening met sociale variabelen
3. Kan leiden tot werkloosheid
4. Onnauwkeurigheden zijn nog steeds mogelijk
5. Gevoelig voor veiligheidsrisico's

[23][24]

5.1.3 Onderzoek in de praktijk

Door met externen in gesprek te zijn gegaan en door verschillende studie geraadpleegd te hebben over het onderwerp Ai in de gezondheidszorg kon alleen maar geconcludeerd worden, dat de gemaakte applicatie alleen in gebruik genomen kan worden wanneer dit in samenwerking met een expert gebruikt zou worden die de resultaten evalueert. Dit op te merken dat de gemaakte applicatie geen vervanger probeert te worden maar een tool.

5.2 Concrete aanbevelingen

Wanneer er een beperkte kennis is over hersentumoren is het zeker aan te raden om dit eerst te bestuderen. Idealiter gebeurt de labeling door een expert die kennis beschikt over de nodige zaken.

Als het beschikbare tools en middelen om de masks handmatig te gaan labelen beperkt is, dan is de beste optie om de masks te gaan segmenteren doormiddel van een region based segmentatie algoritme. Hetzelfde geldt als er beperkt kennis is over de verschillende tumoren, dan zou dit een gevaar kunnen zijn en kan leiden tot doden. Dit soort errors moeten zeker vermeden worden en het beste optie zou dan zijn om terug met een region based segmentatie algoritme te werken.

Wat vooral geweten moet worden is dat een Ai nooit een 100% accuraatheid heeft en wat dus betekent dat dit niet als standalone gebruikt kan worden. Ook kunnen er verschillende dingen verkeer lopen, zoals eerder aangekaart kan er bv. misgelabeld worden die dan fouten in het systeem veroorzaken.

Als beste advies dat gegeven kan worden is dat de gemaakte applicatie niet een vervanger gezien mag worden, maar wel als hulpmiddel. Dit dient dan ook wel onder toezicht van een expert gebruikt te worden, die de resultaten nakijkt en aanpast. Misschien in de loop van tijd dat het Ai model dan wel voldoende heeft bijgeleerd en als standalone gebruikt kan worden.

5.3 Stappenplan

Om te beginnen is er niet veel kennis vereist om de MRi scans te gaan verzamelen en correct op te slaan. De scans zou moeten bestaan uit zowel hersenen die een tumor van het variant glioom bevatten als gezonde. De hoeveelheid hersenen die een tumor bevatten als gezonden zouden ook ongeveer gelijk moeten zijn om zo geen ongelijk verdeelde dataset te verkrijgen. Ook zou er naar een grootte hoeveelheid scans gestreefd moet worden om zo een snellere betere voorspelling te kunnen behalen.

Het volgend stuk vereist wel de nodige kennis, de vereiste kennis is het verschil kennen tussen een gezonde hersenen en hersenen die een tumor bevatten. Het volgende stap is dus om de hersenen handmatig te gaan labelen. Dit zou wel met de nodig attentie moeten gebeuren, want bij een fout kan het later voor problemen zorgen. Nadat de labeling gebeurd is zou dit best nog eens nakeken worden door een expert of een neuroloog om de fouten eruit te halen.

Als dit gebeurd is de volgende stap om de tumoren te gaan segmenteren. Hierbij is de nodige middelen en kennis over dit onderwerp zeer zeker van toepassing. De tumoren zouden zo precies mogelijk gesegmenteerd moeten worden om de beste resultaat te kunnen behalen. Zeker in geval van wat kleinere tumoren is het zeker een verstandige keuze om hiervoor wat preciezer te werk te gaan. Indien er niet over de nodige kennis beschikt dan kan er gewerkt worden met region based segmentatie algoritme.

De volgende stap is om verschillende modellen te trainen en met elkaar te vergelijken. Merk op dat het gebruik maken van transfer learning modellen als eigen modellen een must is. Nadat de modellen getraind en geëvalueerd zijn kunnen ze samen gevoed worden tot een ensemble learning model.

De laatste stap is om een applicatie met de getrainde modellen te ontwikkelen die gebruikt kan worden door een neuroloog.

5.4 Ontwikkelde tools

Uit project vier zou afgeleid kunnen worden dat het gebruik maken van eigen classificatie modellen als transfer learning modellen relevant is. Zo kan er een vergelijking gemaakt worden en zouden verschillende conclusies uit getrokken kunnen worden. Wat voornamelijk het accuraatheid heeft opgetrokken bij de classificatie gedeelte, is het gebruik maken van een ensemble learning model. Een ensemble learning model kan tevens ontwikkeld worden voor allerlei doeleinden.

De ontwikkelde tool uit project vier is de proof of concept applicatie die gebruikt zou kunnen worden om hersentumoren automatisch te gaan detecteren. Er zou alleen maar images geïmporteerd moeten worden en de applicatie zal het werk doen. Indien de applicatie een tumor gedetecteerd heeft zal het ook door het segmentatie model doorgestuurd worden, die de tumor zal inkleuren. Indien de scan geen tumor bevat zal het als geen tumor worden geclassificeerd en weergegeven.

De applicatie is eens getest met een andere dataset. In het begin werkte het applicatie niet goed, na de data analyse gedaan te hebben kon worden waargenomen dat het gaat over verschillende types hersentumoren. Er werd dan opzoek gegaan naar een andere dataset waarvan de hersentumoren van het type glioom zijn en dit werkte wel.

6 Conclusie

Dit onderzoek werd uitgevoerd om te achterhalen of het mogelijk is om de detectie van hersentumoren te gaan automatiseren. Door de gemaakte applicatie en de behaalde resultaten uit project vier is het bewezen dat het effectief wel mogelijk is, maar dat er wel verschillende moeilijkheden eerst overwonnen moeten worden vooraleer het gemaakte applicatie in gebruik genomen kan worden.

Aan de hand van de uitgevoerde onderzoek kon er geconcludeerd worden dat het beste model voor het classificeren van de tumoren de ensemble learning model is. De ensemble learning model bestaat uit drie best presterende modellen: VGG19, OwnClassificationNN en OwnClassificationNNAUG. Deze drie modellen werken nauw samen om het beste resultaat te behalen. Het model behaalde een accuraatheid van 98,18%.

Door twee meest gebruikte segmentatie modellen(voor medische afbeeldingen) te trainen en met elkaar vergeleken te hebben, kon er geconcludeerd worden dat het UNet model op de dataset een stukje beter presteert dan het Deeplabv3+ model. Het UNet model behaalde een dice coef loss van 0.0022, wat heel goed is.

De ontwikkelde tool uit project vier is de proof of concept applicatie die gebruikt zou kunnen worden om hersentumoren automatisch te gaan detecteren. Er zou alleen maar images geïmporteerd moeten worden en de applicatie zal het werk doen. Indien de applicatie een tumor gedetecteerd(classificatie gedeelte) heeft zal het ook door het segmentatie model doorgestuurd worden, die de tumor zal inkleuren. Indien de scan geen tumor bevat zal het als geen tumor worden geclassificeerd en weergegeven.

Dag van vandaag gebeurt de detectie van hersentumoren handmatig door een neuroloog. Dit komt omdat men geen vertrouwen heeft in AI algoritmes, zeker als het gaat over medische applicaties. Doordat AI nooit een accuraatheid van 100% kan halen en dat er altijd een fout aanwezig is heeft men weinig vertrouwen en denkt men het handmatig beter te kunnen. Ook wanneer dat er een fout over het hoofd werd gezien doorheen het preprocessing proces dat het model dan helemaal niet goed gaat werken. Dit zijn voornamelijk de zaken van waarom artsen niet willen werken met AI.

Door de aangekaarte zaken in rekening te nemen en door in gesprek te gaan met derden kon er geconcludeerd worden dat artsen wel in staat zouden zijn om de gemaakte proof of concept applicatie als hulpmiddel te gebruiken, maar niet als vervanger. Het proces zou dan wel onder strikte toezicht moeten uitgevoerd worden om menselijk fouten te vermijden.

7 Literatuurlijst

- [1]
'Data augmentation', *Wikipedia*. 24 april 2022. Geraadpleegd: 30 april 2022. [Online]. Beschikbaar op: https://en.wikipedia.org/w/index.php?title=Data_augmentation&oldid=1084451686
- [2]
'Deep learning', *Wikipedia*. 9 april 2021. Geraadpleegd: 30 april 2022. [Online]. Beschikbaar op: https://nl.wikipedia.org/w/index.php?title=Deep_learning&oldid=58677184
- [3]
'Ensemble learning', *Wikipedia*. 22 januari 2022. Geraadpleegd: 30 april 2022. [Online]. Beschikbaar op: https://en.wikipedia.org/w/index.php?title=Ensemble_learning&oldid=1067251577
- [4]
'Transfer learning', *Wikipedia*. 3 april 2022. Geraadpleegd: 30 april 2022. [Online]. Beschikbaar op: https://en.wikipedia.org/w/index.php?title=Transfer_learning&oldid=1080756337
- [5]
S. Gull, S. Akbar, en H. U. Khan, 'Automated Detection of Brain Tumor through Magnetic Resonance Images Using Convolutional Neural Network', *BioMed Research International*, vol. 2021, p. e3365043, nov. 2021, doi: [10.1155/2021/3365043](https://doi.org/10.1155/2021/3365043).
- [6]
'Wat is een hersentumor? Hersentumor (astrocytoma)', *Hersenstichting*. <https://www.hersenstichting.nl/hersenaandoeningen/hersentumor/> (geraadpleegd 26 februari 2022).
- [7]
'Fig. 7 en 8: Sagittale en axiale T1-gewogen MRI-beelden + Gd tonen een...', *ResearchGate*. https://www.researchgate.net/figure/en-8-Sagittale-en-axiale-T1-gewogen-MRI-beelden-Gd-tonen-een-duidelijke-afname-van-de_fig1_324161655 (geraadpleegd 20 maart 2022).
- [8]
'Onderzoek en diagnose bij hersentumoren'. <https://www.kanker.nl/kankersoorten/hersentumoren/onderzoek-en-diagnose/onderzoek-en-diagnose-bij-hersentumoren> (geraadpleegd 26 februari 2022).
- [9]
'Exploring Brain Tumours and Brain Tumour Misdiagnosis'. <https://hatchbrenner.co.uk/news/exploring-brain-tumours-and-brain-tumour-misdiagnosis> (geraadpleegd 26 februari 2022).
- [10]
A. Bonner, 'The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks', *Medium*, 1 juni 2019. <https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb> (geraadpleegd 26 februari 2022).
- [11]
'A Comparison of 4 Popular Transfer Learning Models', *Analytics India Magazine*, 1 september 2021. <https://analyticsindiamag.com/a-comparison-of-4-popular-transfer-learning-models/> (geraadpleegd 26 februari 2022).

[12]

'Fig. 5: The architecture of our transfer learning model.', *ResearchGate*.

https://www.researchgate.net/figure/The-architecture-of-our-transfer-learning-model_fig4_342400905 (geraadpleegd 20 maart 2022).

[13]

'How to do Semantic Segmentation using Deep learning', *AI & Machine Learning Blog*, 19 mei 2021.

<https://nanonets.com/blog/how-to-do-semantic-segmentation-using-deep-learning/> (geraadpleegd 27 februari 2022).

[14]

H. Lamba, 'Understanding Semantic Segmentation with UNET', *Medium*, 17 februari 2019.

<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47> (geraadpleegd 27 februari 2022).

[15]

O. Ronneberger, P. Fischer, en T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation', in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, en A. F. Frangi, Red. Cham: Springer International Publishing, 2015, pp. 234–241. doi: [10.1007/978-3-319-24574-4_28](https://doi.org/10.1007/978-3-319-24574-4_28).

[16]

'Papers with Code - DeepLabv3 Explained'. <https://paperswithcode.com/method/deeplabv3>

(geraadpleegd 27 februari 2022).

[17]

H. Rowlani, 'Implementing a fully convolutional network (FCN) in TensorFlow 2', *Medium*, 9 oktober 2020.

<https://towardsdatascience.com/implementing-a-fully-convolutional-network-fcn-in-tensorflow-2-3c46fb61de3b> (geraadpleegd 15 mei 2022).

[18]

E. Tiu, 'Metrics to Evaluate your Semantic Segmentation Model', *Medium*, 3 oktober 2020.

<https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2> (geraadpleegd 25 januari 2022).

[19]

J. Petch, S. Di, en W. Nelson, 'Opening the Black Box: The Promise and Limitations of Explainable Machine Learning in Cardiology', *Canadian Journal of Cardiology*, vol. 38, nr. 2, pp. 204–213, feb. 2022, doi: [10.1016/j.cjca.2021.09.004](https://doi.org/10.1016/j.cjca.2021.09.004).

[20]

'Top 10 Applications of Machine Learning in Healthcare - FWS'.

<https://www.flatworldsolutions.com/healthcare/articles/top-10-applications-of-machine-learning-in-healthcare.php> (geraadpleegd 12 april 2022).

[21]

'Machine Learning in Healthcare: 12 Real-World Use Cases – NIX United', *NIX United – Custom Software Development Company in US*, 6 oktober 2021. <https://nix-united.com/blog/machine-learning-in-healthcare-12-real-world-use-cases-to-know/> (geraadpleegd 12 april 2022).

[22]

'Pros & Cons of Artificial Intelligence in Medicine | Drexel CCI'. <https://drexel.edu/cci/stories/artificial-intelligence-in-medicine-pros-and-cons/> (geraadpleegd 12 april 2022).

[23]

'The downside of machine learning in health care', *MIT News / Massachusetts Institute of Technology*.

<https://news.mit.edu/2022/marzyeh-ghassemi-explores-downside-machine-learning-health-care-0201> (geraadpleegd 12 april 2022).

8 Bijlages

Classificatie modellen vergelijken

Research Project 2022

Boutalliss Ouassim

Top3 modellen:

Own classification model

	precision	recall	f1-score	support
0	0.94	0.97	0.96	144
1	0.97	0.93	0.95	131
accuracy			0.95	275
macro avg	0.95	0.95	0.95	275
weighted avg	0.95	0.95	0.95	275

```
[[140  4]
 [ 9 122]]
95.27272727272728
```

Own classification model with augmentation

	precision	recall	f1-score	support
0	0.97	0.94	0.96	144
1	0.94	0.97	0.95	131
accuracy			0.96	275
macro avg	0.96	0.96	0.96	275
weighted avg	0.96	0.96	0.96	275



```
[[136  8]
 [ 4 127]]
95.63636363636364
```

Best performing transfer learning -> vgg19

	precision	recall	f1-score	support
0	0.95	0.92	0.94	144
1	0.92	0.95	0.93	131
accuracy			0.93	275
macro avg	0.93	0.94	0.93	275
weighted avg	0.94	0.93	0.93	275

```
[[133 11]
 [ 7 124]]
93.45454545454545
```

- Keuze model: own classification model with augmentation

Motivatie: Mijn eigen kleine classificatie model met augmentatie toegepast doet het toch een stukje beter, dan de tranfer learning modellen. (zie f1-scores, recall en acc)

Belangrijke scores:

- **eigen model met augmentation:** F1-score(97%) , recall(96%) -> van de tumor klasse/ totale acc: 96,72%
- **beste transferlearning model vgg19:** F1-score(93%), recall(93%) -> van de tumor klasse / totale acc: 93,81%

Waarom kijken naar f1-score van de tumor klasse: Het is heel cruciaal dat het model meer false positives(valse alarm van tumor) predicties doet, dan false negatives(tumor aanwezig, maar niet gepredict wordt). Dit is omdat het een leven kan redden.

Image-size rescaled: 128x128 i.p.v. 256x256, omdat mijn GPU dat anders niet aan kan

Waarom doet de tranfer learning model slechter?: Verschillende factoren kunnen hier een rol inspelen. Zoals de image-size waarop dat het model getraind is niet matched met de classificatie model, de pre-processing van data, snelle overfitting, te groot model voor deze dataset, ongezien images(imageNet).

How to reduce false positives and negatives: Hiervoor zal ik een 3 modellen gebruiken die elkaar steunen om betere predicties te maken. (ensemble model)

Weighted average ensemble (grid search for best weights)

	precision	recall	f1-score	support
0	0.97	0.97	0.97	144
1	0.96	0.97	0.97	131
accuracy			0.97	275
macro avg	0.97	0.97	0.97	275
weighted avg	0.97	0.97	0.97	275

```
[[139  5]
 [ 4 127]]
96.72727272727273
```

```
!]: print("accuracy of own model with augmentation:",acc_ownModel)
print("accuracy of own model with augmentation:",acc_aug)
print("accuracy of vgg19:",acc_vgg19)
print("accuracy of ensemble model:",acc_ensemble)
```

```
accuracy of own model with augmentation: 95.27272727272728
accuracy of own model with augmentation: 95.63636363636364
accuracy of vgg19: 93.45454545454545
accuracy of ensemble model: 96.72727272727273
```

recall: the ratio of correctly predicted. [8]

f1-score: the weighted average of Precision and Recall. [9]

BRONNEN

[1]

'Exploring the limits of transfer learning', 7 februari 2020. <https://www.allerin.com/blog/exploring-the-limits-of-transfer-learning> (geraadpleegd 22 januari 2022).

[2]

'neural network - Is there any proven disadvantage of transfer learning for CNNs?', *Data Science Stack Exchange*. <https://datascience.stackexchange.com/questions/30859/is-there-any-proven-disadvantage-of-transfer-learning-for-cnns> (geraadpleegd 22 januari 2022).

[3]

K. H. Tiet, 'When Does Transfer Learning Fail in Deep Learning', *Aviation Software Innovation*, 3 maart 2021. <https://medium.com/aviation-software-innovation/when-does-transfer-learning-fail-in-deep-learning-31bc2c93f18a> (geraadpleegd 22 januari 2022).

[4]

'(PDF) The Role of Ensemble Learning in Stock Market Classification Model Accuracy Enhancement Based on Naive Bayes Classifiers'. https://www.researchgate.net/publication/338230614_The_Role_of_Ensemble_Learning_in_Stock_Market_Classification_Model_Accuracy_Enhancement_Based_on_Naive_Bayes_Classifiers#pf6 (geraadpleegd 25 januari 2022).

[5]

DigitalSreeni, *213 - Ensemble of networks for improved accuracy in deep learning*, (14 april 2021). Geraadpleegd: 25 januari 2022. [Online]. Beschikbaar op: https://www.youtube.com/watch?v=-ix_Mjzu8BU

[6]

'213 - Ensemble of networks for improved accuracy in deep learning - YouTube'. https://www.youtube.com/watch?v=-ix_Mjzu8BU (geraadpleegd 25 januari 2022).

[7]

'213 - Ensemble of networks for improved accuracy in deep learning - YouTube'. https://www.youtube.com/watch?v=-ix_Mjzu8BU (geraadpleegd 25 januari 2022).

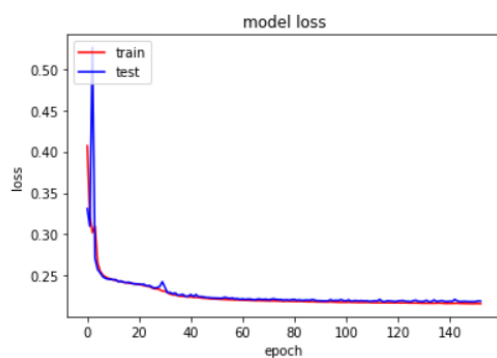
[8]

'sklearn.metrics.recall_score', *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.metrics.recall_score.html (geraadpleegd 25 januari 2022).

[9]

'sklearn.metrics.f1_score', *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.metrics.f1_score.html (geraadpleegd 25 januari 2022).

Unet



```
In [125]: print(Unet.evaluate(X_test_with_mask,y_test_with_mask))
```

```
5/5 [=====] - 0s 14ms/step - loss: 0.2250 - dice_coef_loss: 0.0026  
[0.22495147585868835, 0.002567797200754285]
```

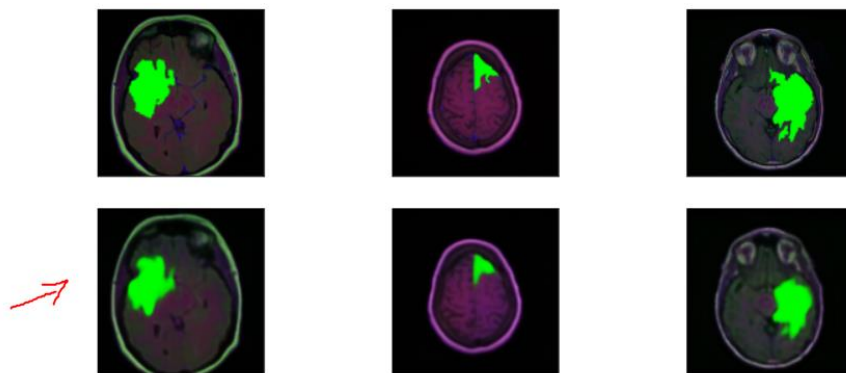
```
ef_loss: 0.0025
```

```
Epoch 00153: early stopping
```

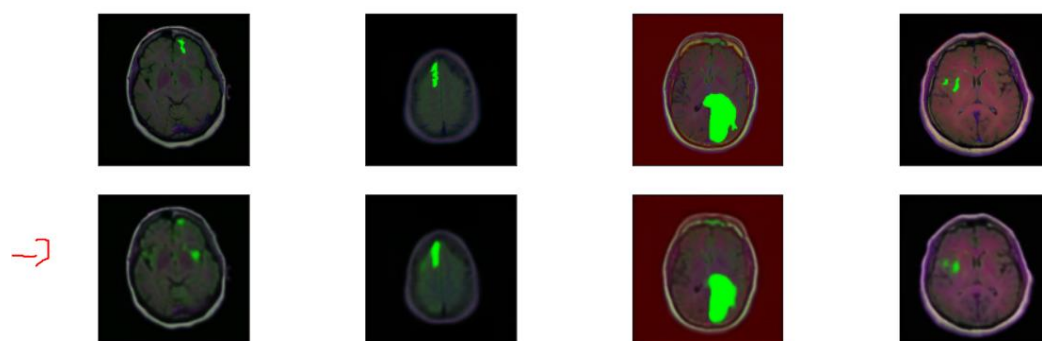
```
Wall time: 4min 11s
```

= ong. 1.6sec per epoch

Easy part segmentation:



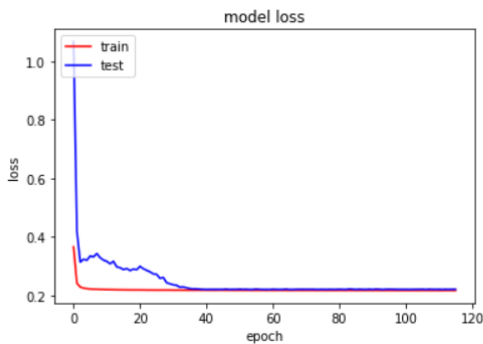
Hard part segmentation:



DeepLabv3+

```
In [103]: print(DeepLabPlus1.evaluate(X_test_with_mask,y_test_with_mask))
```

```
5/5 [=====] - 0s 53ms/step - loss: 0.2264 - dice_coef_loss: 0.0041  
[0.22635498642921448, 0.004124650731682777]
```

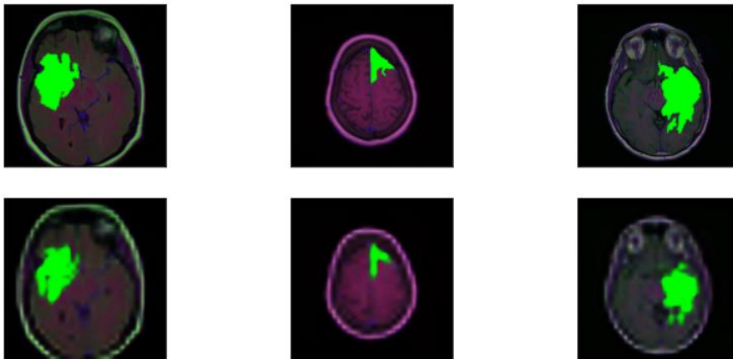


```
oef_loss: 0.0044  
Epoch 00116: early stopping  
Wall time: 11min 51s
```

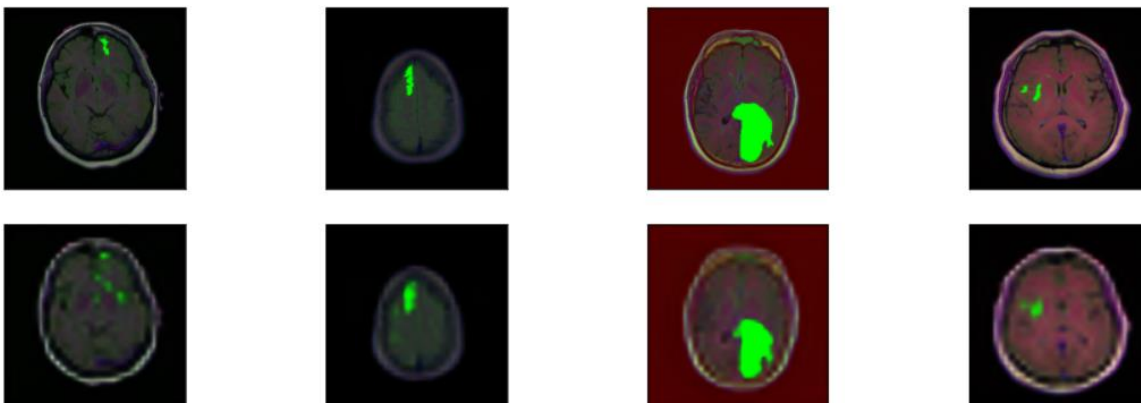
ong.

6 sec per epoch

Easy part:



Hard part:



- Keuze model: UNet

Motivatie: Als we kijken naar de dice_coef_loss, dan scoort Unet daar en stukje beter op dan deeplabv3+. Unet doet het bijna 5X sneller per epoch om te trainen. De gereconstrueerde image ziet er ook veel beter uit bij het UNet model. Bij het kijken naar de *easy part*, dan kan je waarnemen dat

zowel het UNet model als het deeplabv3+ model het degelijk goed doen. Bij het kijken naar de **hard part** kunnen we waarnemen dat het UNet model daar wel een stukje beter op doet.

Easy Part: Wat grotere tumoren, makkelijker te segmenteren.

Hard Part: Veel kleinere tumoren, veel moeilijker dit correct te segmenteren.

Belangrijke scores:

- **UNet:** Loss (0.2250) ,dice_coef(0.9976), dice_coef_loss(0.0026)
- **Deeplabv3+:** Loss (0.2264) ,dice_coef(0.9957), dice_coef_loss(0.0041)

Waarom doet Deeplabv3+ hier een stukje minder op: Het probleem hierbij is niet het model, maar de pre-trainend model ResNet50 die gebruikt wordt als backbone. Als we kijken naar de classificatie gedeelte dan kan je waarnemen dat de pre-trained (resnet50) model het niet echt goed doet. Dit betekent dus dat de pre-trainend ResNet50 model voor dit dataset niet geschikt is. En dat we beter een andere pre-trained model gebruiken als backbone zoals VGG19.

Waarom Dice_coef_loss: In de praktijk wordt dikwijls de dice coëfficiënt loss gebruikt. Deze kijkt naar de overlap tussen twee data samples. In dit geval hoeveel de door de gebruikte model geneerde mask overlapt met de ground truth mask.

Dice_coef: *"Dice coefficient is a measure of overlap between two masks. 1 indicates a perfect overlap while 0 indicates no overlap"* [4]

Wat had ik beter nog gedaan:

- Andere pre-trained model gebruiken als backbone voor deeplabv3+

BRONNEN

[1]

'Evaluating image segmentation models.', *Jeremy Jordan*, 30 mei 2018.

<https://www.jeremyjordan.me/evaluating-image-segmentation-models/> (geraadpleegd 25 januari 2022).

[2]

'Exploring the limits of transfer learning', 7 februari 2020. <https://www.allerin.com/blog/exploring-the-limits-of-transfer-learning> (geraadpleegd 22 januari 2022).

[3]

S. Tas, 'How To Evaluate Image Segmentation Models', *Medium*, 17 oktober 2020.

<https://towardsdatascience.com/how-accurate-is-image-segmentation-dd448f896388> (geraadpleegd 25 januari 2022).

[4]

E. Tiu, 'Metrics to Evaluate your Semantic Segmentation Model', *Medium*, 3 oktober 2020.

<https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2> (geraadpleegd 25 januari 2022).

[5]

'Metrics to Evaluate your Semantic Segmentation Model | by Ekin Tiu | Towards Data Science'.

<https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2> (geraadpleegd 25 januari 2022).

RGB - classificatie

	precision	recall	f1-score	support
0	0.96	0.96	0.96	280
1	0.96	0.96	0.96	270
accuracy			0.96	550
macro avg	0.96	0.96	0.96	550
weighted avg	0.96	0.96	0.96	550

```

[[268 12]
 [ 11 259]]
95.81818181818181

```

Grayscale images classificatie

	precision	recall	f1-score	support
0	0.92	0.94	0.93	280
1	0.94	0.92	0.93	270
accuracy			0.93	550
macro avg	0.93	0.93	0.93	550
weighted avg	0.93	0.93	0.93	550

```

[[263 17]
 [ 22 248]]
92.9090909090909

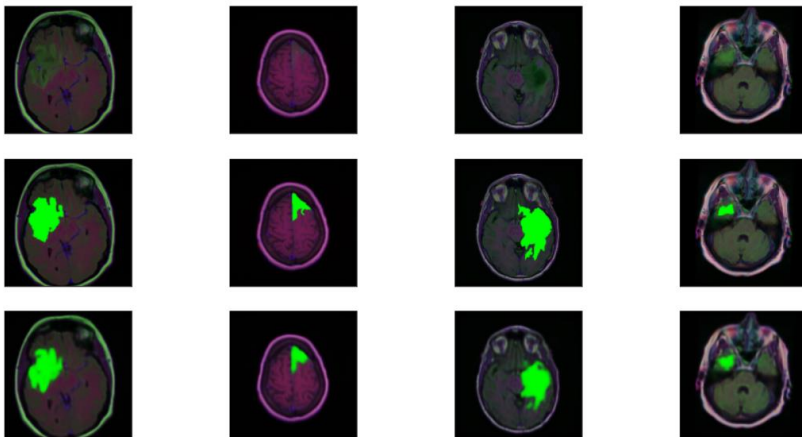
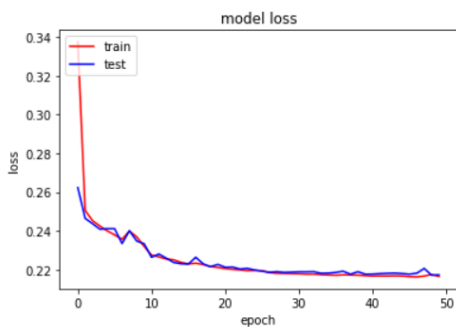
```


RGB -> bij u-net

```

ef_loss: 0.0025
Epoch 45/50
70/70 [=====] - 2s 30ms/step - loss: 0.2167 - dice_coef_loss: 0.0021 - val_loss: 0.2180 - val_dice_co
ef_loss: 0.0025
Epoch 46/50
70/70 [=====] - 2s 30ms/step - loss: 0.2164 - dice_coef_loss: 0.0021 - val_loss: 0.2176 - val_dice_co
ef_loss: 0.0024
Epoch 47/50
70/70 [=====] - 2s 30ms/step - loss: 0.2162 - dice_coef_loss: 0.0020 - val_loss: 0.2182 - val_dice_co
ef_loss: 0.0024
Epoch 48/50
70/70 [=====] - 2s 30ms/step - loss: 0.2167 - dice_coef_loss: 0.0021 - val_loss: 0.2206 - val_dice_co
ef_loss: 0.0032
Epoch 49/50
70/70 [=====] - 2s 30ms/step - loss: 0.2175 - dice_coef_loss: 0.0025 - val_loss: 0.2171 - val_dice_co
ef_loss: 0.0022
Epoch 50/50
70/70 [=====] - 2s 30ms/step - loss: 0.2165 - dice_coef_loss: 0.0021 - val_loss: 0.2173 - val_dice_co
ef_loss: 0.0022

```



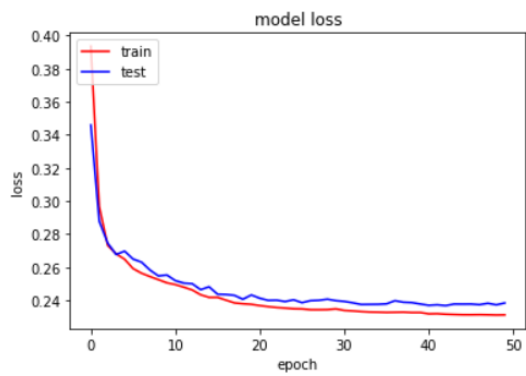
Grayscale bij unet -> segmentation

```

70/70 [=====] - 2s 27ms/step - loss: 0.2312 - dice_coef_loss: 9.9787e-04 - val_loss: 0.2373 - val_dic
e_coef_loss: 0.0015
Epoch 48/50
70/70 [=====] - 2s 27ms/step - loss: 0.2311 - dice_coef_loss: 9.7914e-04 - val_loss: 0.2381 - val_dic
e_coef_loss: 0.0015
Epoch 49/50
70/70 [=====] - 2s 27ms/step - loss: 0.2310 - dice_coef_loss: 9.6738e-04 - val_loss: 0.2372 - val_dic
e_coef_loss: 0.0014
Epoch 50/50
70/70 [=====] - 2s 27ms/step - loss: 0.2311 - dice_coef_loss: 9.6780e-04 - val_loss: 0.2383 - val_dic
e_coef_loss: 0.0016

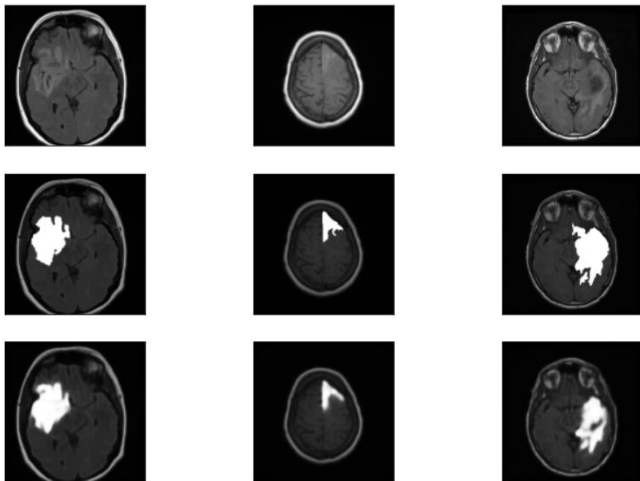
```

```
146]: # Plot history
```



```
[147]: print(UNet.evaluate(X_test_with_mask,y_test_with_mask))
```

```
5/5 [=====] - 1s 78ms/step - loss: 0.2430 - dice_coef_loss: 0.0015  
[0.2429526299238205, 0.0015369134489446878]
```








Besluit:

De modellen werken iets beter op rgb images dan op grayscale images.

Stap 1

De broncode(https://github.com/ouassimboutalliss/research_project_How-can-the-detection-of-brain-tumors-be-automated) downloaden als .zip bestand en uitpakken. De files moeten er als volgt uitzien:

 dataset	26/01/2022 12:21	Bestandsmap	
 Diagram	26/01/2022 12:21	Bestandsmap	
 Model	26/01/2022 13:09	Bestandsmap	
 README	26/01/2022 12:21	Markdown Source...	1 kB
 Research_project	26/01/2022 12:21	Jupyter Source File	8.493 kB

Stap 2

Voor je de bestanden zal kunnen runnen moet je een aantal libraries installeren. Dit doe je door “pip install ...” te doen in je commandprompt.

De te installeren libraries zijn:

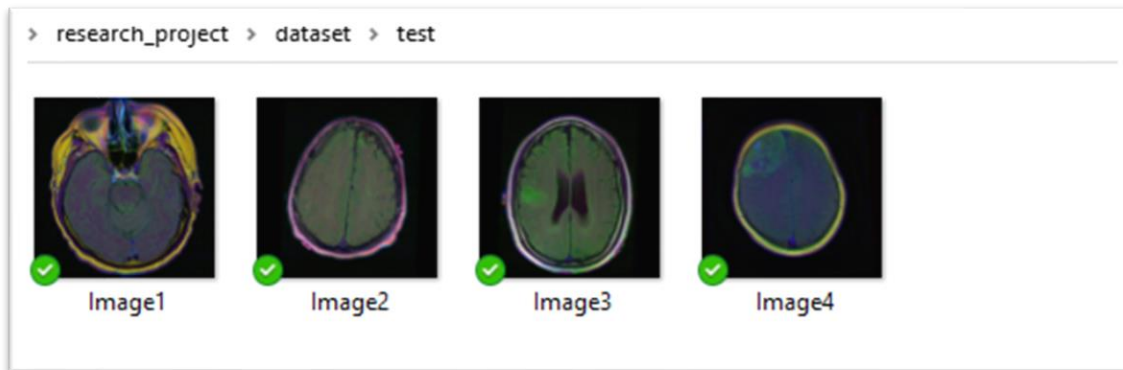
- Numpy
- Pandas
- Matplotlib
- Scikit-learn
- Opencv-python
- Scipy
- Seaborn
- Tensorflow

Volledig onderzoek

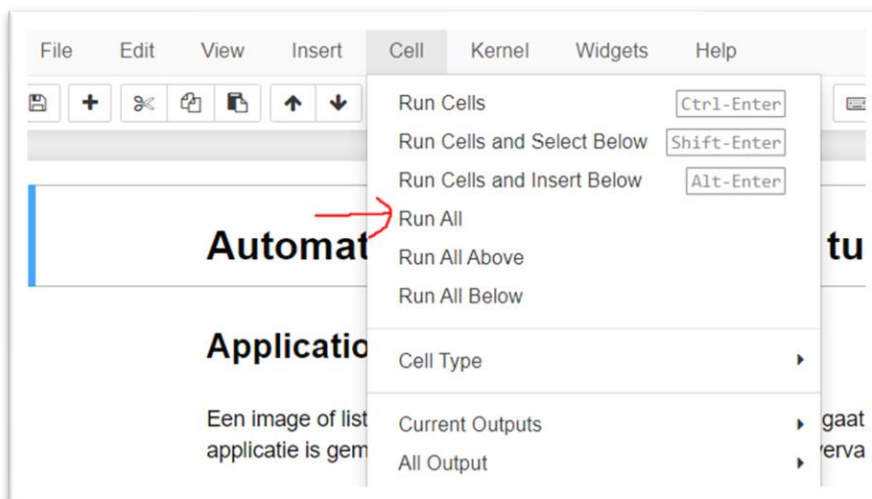
In de main folder van de broncode staat een bestand genaamd `research_project.ipynb`. Als u deze zal gaan openen zal u mijn volledig onderzoek kunnen bezichtigen.

Application

In de main folder van de broncode staat een bestand genaamd `Application_tumor_detection.ipynb`. Als u deze zal gaan openen zal u de applicatie kunnen gebruiken, maar vooraleer u de applicatie kunt uitvoeren. Moet u eerst nog de MRI scans gaan plaatsen in de map `./dataset/test`. (zie voorbeeld hieronder)



Indien dit is gebeurd hoeft u alleen maar alle cellen te runnen. Dit kunt u doen via de navigatiebalk `Cell>Run All`. (zie voorbeeld hieronder)



Prediction

Achter de schermen zullen de images door een ensemble classificatie model gestuurd worden. Indien het model een tumor heeft gedetecteerd, dan zal het door ook het segmentatie model doorgestuurd worden. Dit betekend dus dat de tumor ingekleurd zal worden, zodat het zichtbaar wordt waar de tumor zich bevindt. Indien er geen tumor gevonden werd, zal het geclassificeerd worden als **No tumor detected**.

Na het uitvoeren van de applicatie, dan zal u dit te zien krijgen. MRI scans zonder tumoren zullen onder de **List – No tumor detected** komen. (zie voorbeeld hieronder)

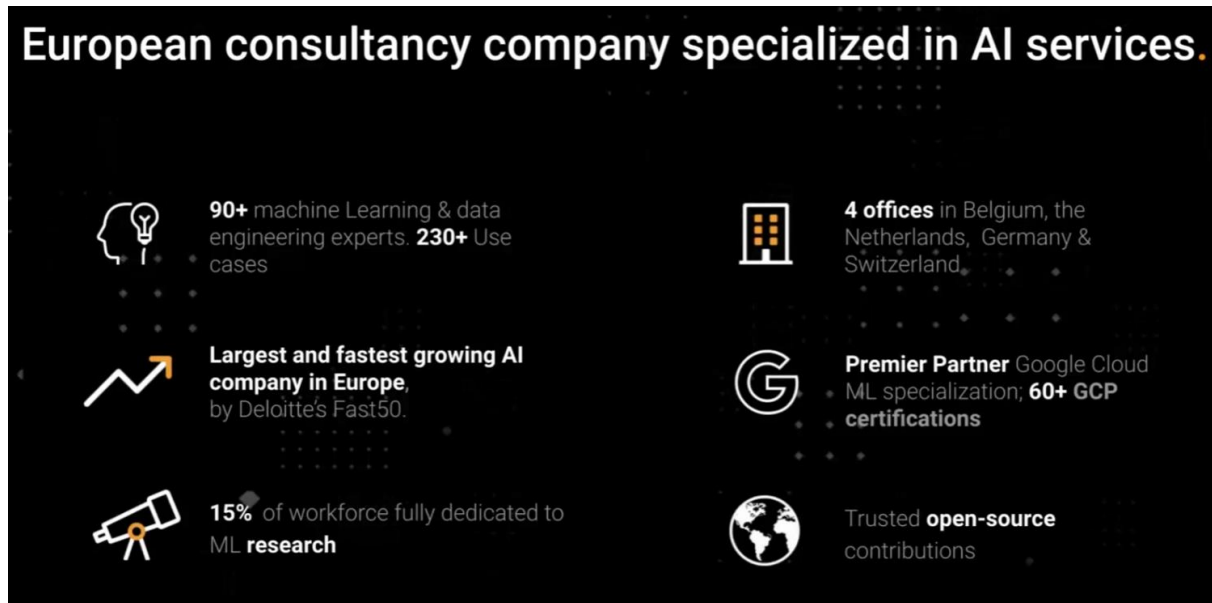


Indien de MRI scans wel een tumoren bevatten, dan zullen de scans onder **List-Tumor detected** komen. (zie voorbeeld hieronder)



Op donderdag 20 Januari, 2022, kwam de CTO (Chief Technologie Officer) van ML6 spreken over explainable Artificial Intelligence. Ze houden zich voornamelijk bezig met consulting voor andere bedrijven om zo hun Ai toepassingen te implementeren.

Het bedrijf bestaat uit meer dan 90 machine learning & data engineering experts. Via de website wordt er ook vermeld dat ze in 4 verschillende landen gevestigd zijn. In België, Nederland, Duitsland en Zwitserland.



De sessie werd geopend met de voorstelling van het bedrijf. Het werd al snel duidelijk dat ML6 een consulting bedrijf is die zich focust op Machine Learning projecten. De projecten die ze aannemen zijn voornamelijk innovatieve projecten. Voor het bedrijf zelf maakt het eigenlijk niet uit wat voor soort project het is, zolang het rond Machine Learning draait kunnen ze aan de slag.

Zie hier een afbeelding met de verschillende partners van ML6.

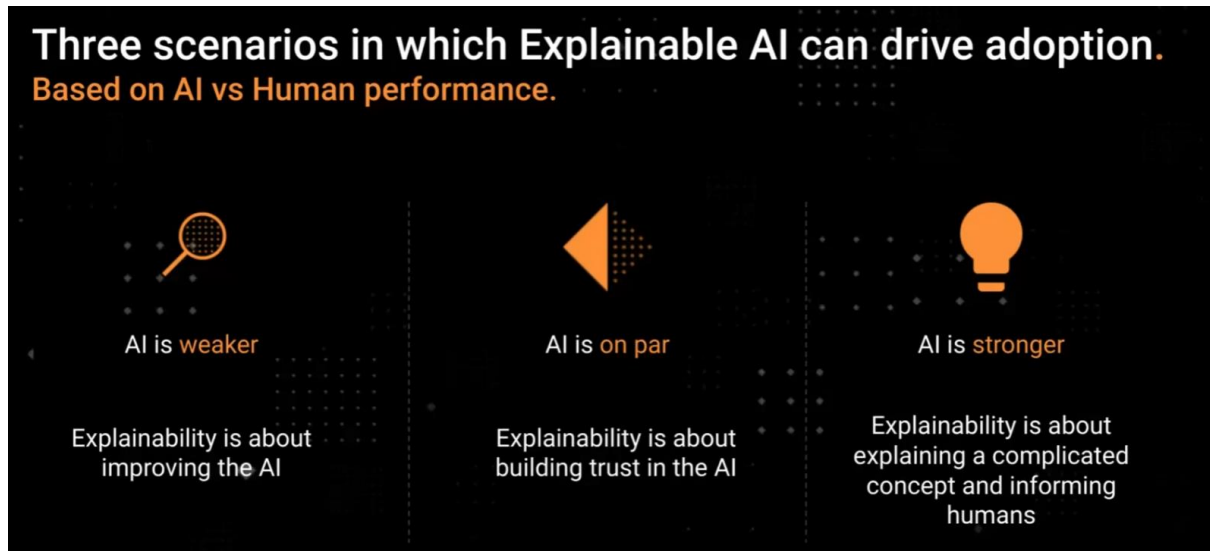


Na de uitleg over het bedrijf werden er enkele voorbeelden gegeven over het concept Explainable Ai. Hierbij werden de 3 categorieën rond explainable AI uitgelegd aan de hand van enkele voorbeelden. Het grootste probleem met ML is dat de output niet zomaar verklaarbaar is. Dit wilt zeggen dat het moeilijk te verklaren valt van waarom het model nu juist die voorspelling maakt (blackbox). De eindgebruiker is hiermee niet gediend en zou graag een verklaring willen, dit is meer om vertrouwen in het model te krijgen. Daarom dat ML6 nu zich een beetje verdiept in het hele Explainable Ai gebeuren.

“Black boxes bij Kunstmatige Intelligentie treden op wanneer machine learning systemen beslissingen maken die door de gebruiker niet duidelijk of vanzelfsprekend zijn. Het gebruik van de zelflerende algoritmes kan onbedoeld leiden tot discriminatie of een gebrek aan vertrouwen” [1]

Allereerst moet er geweten worden wanneer het noodzakelijk is om een model te kunnen verklaren. Dit zijn de 3 scenario's:

- Wanneer een Ai slechter of trager presteert dan een mens
- Wanneer het even goed kan werken
- Wanneer het sneller en beter kan werken



Wat is nu het doel om in staat te zijn de voorspellingen te verklaren?

Het doel hierbij is dat er vertrouwen wordt gecreëerd, mogelijkheid om het model te verbeteren en duidelijkheid.

Enkele voorbeelden werden gegeven om de bovenstaande wat te verduidelijken:

Wanneer het AI model slechter presteert dan een mens.

- **Hoofdoel:** model verbeteren
- Visualisatie van model outputs
 - o Maak duidelijk wat het model doet, zodat het vertrouwen in het model niet verloren gaat
- Active Learning
 - o De mens de mogelijkheden geven om het model te helpen, zodat het beter wordt
 - o Als opportuniteit om het beter te maken
- Options to override model
 - o Back-up plan voorzien

Wanneer het AI model gelijk presteert als een mens.

- **Hoofdoel:** Argumenteren waarom het model zo presteert
- Waarom kiest het model dat
- Belangrijkste features aanduiden
- Werken met explainable outputs (geen ja of nee)

Wanneer het AI model beter presteert dan een mens.

- **Hoofdoel:** Argumenteren waarom het model een complexe voorspelling beter kan doen dan de mens
- Waarom maakt het model dit voorspelling
- Welke features zijn van belang
- Correlaties aanduiden
- Belangrijkste features zoeken

Mathias heeft een checklist opgesteld om een project explainable te krijgen.

Five generic design patterns: check list om een project explainable te maken:

- **Problem reframing**
 - o Goed begrijpen wat het probleem is
- **Interpretable model**
 - o Inzicht bieden in het model voor de eindgebruiker
- **Feature attribution**
 - o Welke features eigenlijk van belang zijn
- **Transparency & transferability**
 - o Stukje rond data -> andere data gebruiken om je model mee te testen
- **Intuitive visualizations**
 - o Output moet leesbaar zijn en duidelijk door bijvoorbeeld grafieken, afbeeldingen, en geen ja of nee te gebruiken

Op het einde van de rit werd er een conclusie gesteld. Als conclusie werd meegegeven dat explainability ai een stuk is van een project en niet alleen van het model. Ten 2^{de} wat explainability ai is, is afhankelijk van het probleem. Ten 3^{de} explainability ai start bij de dataset en eindigt bij de UI. En als laatst explainability ai zou het vertrouwen in het model moeten verhogen en het gebruik ervan.

BRONNEN

[1]

'Algoritmisering en de black box - tijd voor verklaarbare AI?', *Innovatief in Werk*.

<https://www.innovatiefinwerk.nl/toekomst-van-werk-arbeidsmarkt-van-morgen/2019/07/algoritmisering-en-de-black-box-tijd-voor> (geraadpleegd 18 februari 2022).