Deep learning voor dysfagiegradatie

Een voorstudie naar reconstructie van dynamische MR-beelden van slikbewegingen vanuit k-space en relevante klinische aspecten

Multidisciplinaire Opdracht Technische Geneeskunde

Elke Croonen (s1796690)
Tharanghi Logendran (s1700545)
Lianne Straetemans (s1775316)
Renate Visser (s1807161)

Beoordelingscommissie:
Dr. ir. F. Van der Heijden
Prof. dr. L.E. Smeele
Dr. R. Dirven

22 juni 2019

UNIVERSITY OF TWENTE.

Abstract

Probleemstelling: Dysfagie kan leiden tot ondervoeding, uitdroging, longontsteking of zelfs tot overlijden. Naar schatting lijdt 6%van de Nederlandse bevolking aan dysfagie en bij patiënten met ernstige aandoeningen in het hoofd-halsgebied is dat zelfs meer dan 50%. De huidige methode om dysfagie te gradueren is videofluoroscopie. Nadelen hiervan zijn dat de patiënt blootgesteld wordt aan röntgenstraling, weke delen niet goed zichtbaar zijn en de beoordeling subjectief is. Dynamische MRI (dMRI) zou voor deze problemen een oplossing kunnen vormen. Echter, de temporele resolutie van dMRI is op dit moment nog te laag om slikvideo's te maken. Door MRI-data *undersampled* op te nemen kan de temporele resolutie verhoogd worden. Dit gaat echter ten koste van de spatiële resolutie.

Doel: Dit onderzoek heeft als doel het zetten van een eerste stap om met *deep learning* (DL) MR-beelden te reconstrueren uit *undersampled k-space* om zo de spatiële resolutie van dMRI te verhogen en dysfagiegradatie met dMRI mogelijk te maken in de kliniek.

Methode: ADMM-net is getraind met *undersampled k-spacedata* met als referentie-afbeelding een gereconstrueerd MR-beeld van het hoofd-halsgebied. Daarnaast zijn interviews gehouden met specialisten om relevante aspecten van dysfagiegradatie in beeld te brengen.

Conclusie: Het reconstrueren van MR-beelden vanuit *undersampled k-space* is mogelijk met behulp van ADMM-Net. Dysfagiegradatie is met name gebaseerd op het Logemann's model en heeft betrekking op de verwerking van de bolus en de functie van anatomische structuren hierbij.

Discussie: De spatiële resolutie van de output van het DL-netwerk is nog onvoldoende om in de kliniek toegepast te kunnen worden. Voor toekomstig onderzoek wordt daarom aanbevolen om het netwerk te trainen met andere instellingen en datasets, bijvoorbeeld 3D-beelden en beelden van patiënten.

Zoektermen: Dysfagie • dynamische MRI • deep learning • undersampling • ADMM • k-space

Dankwoord

Ten eerste willen wij dr.ir. F. van der Heijden bedanken voor zijn begeleiding tijdens de MDO en het bieden van handvatten bij *deep learning*. Ook willen wij prof. dr. L. E. Smeele bedanken voor zijn medische expertise op het gebied van hoofd-halsoncologie en zijn hulp bij het contacteren van andere medische specialisten. Daarnaast gaat onze dank uit naar L. Voskuilen, MSc, voor zijn uitvoerige uitleg over de techinsche achtergrond van *deep learning* en dynamische MRI en het bieden van zijn software en (proefpersonen)data. Ook zijn wij dr. L. van der Molen, prof. dr. A.J.M. Balm en dr. Maarten van Alphen dankbaar voor de tijd die zij hebben genomen om onze interviewvragen te beantwoorden. Tot slot willen wij onze tutor I.G. van Bruggen bedanken voor haar procesbegeleiding en steun gedurende de hele MDO.

Inhoudsopgave

1 Inleiding **1**
1.1 Klinische achtergrond . . . 1
1.1.1 Fysiologie van de slikbeweging . . . 1
1.1.2 Dysfagie . . . 2
1.2 Diagnostiek en therapie . . . 2
1.2.1 Huidige diagnostiek en therapie . . . 2
1.2.2 Nadelen videofluoroscopie . . . 3
1.2.3 Dynamische Magnetic Resonance Imaging . . . 3
1.3 Doelstelling . . . 4
1.4 Leeswijzer . . . 4

2 Technische Achtergrond **5**
2.1 K-space . . . 5
2.2 Sampling methoden . . . 5
2.3 Undersampling . . . 5
2.4 Compressive sensing . . . 6
2.5 Deep learning . . . 6
2.6 Deep Learning en dynamische MRI . . . 8
2.6.1 ADMM-net . . . 9

3 ADMM-net: Een proof-of-concept **12**
3.1 Methode . . . 12
3.1.1 Data-acquisitie . . . 12
3.1.2 Verificatie van het netwerk . . . 13
3.1.3 Experimentele opstelling . . . 13
3.1.4 Definitieve training . . . 15
3.2 Resultaten . . . 16
3.2.1 Training met 360 beelden met het 20% masker . . . 16
3.2.2 Training met 150 beelden met het 20% masker . . . 16
3.2.3 Training met 8 spokes . . . 16
3.3 Discussie . . . 17
3.3.1 Bespreking Resultaten . . . 17
3.3.2 Onverwachte bevindingen . . . 18
3.3.3 Beperkingen . . . 18
3.3.4 Aanbevelingen . . . 19
3.4 Conclusie . . . 20

4 Klinische aspecten **21**
4.1 Klinische parameters uit videofluoroscopie . . . 21
4.2 De output van een DL-netwerk in de kliniek . . . 21

5 Algehele discussie **23**
5.1 Deep learning en dynamische MRI . . . 23
5.2 Toekomstperspectief . . . 23
5.3 Conclusie . . . 24

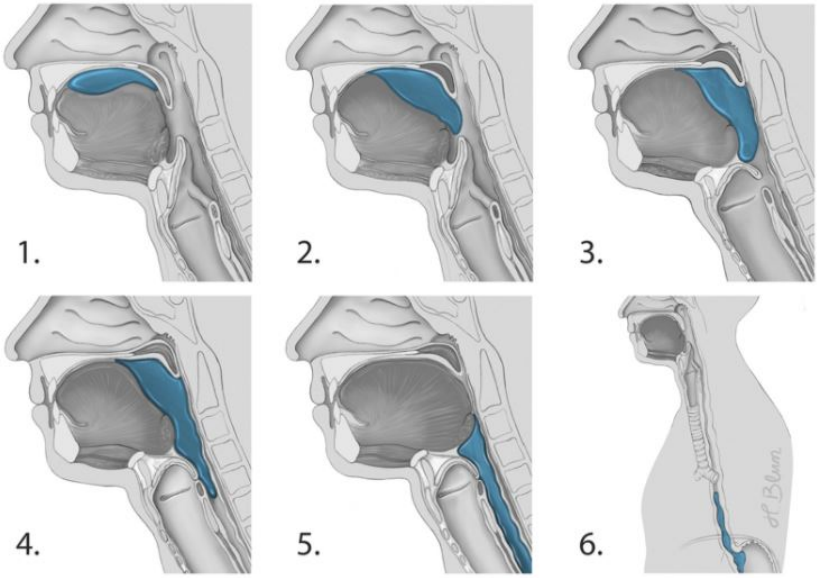
Referenties **25**

Bijlage **29**

1 Inleiding

Het is voor de meeste mensen iets heel gewoons: slikken. Onbewust slikt een persoon gemiddeld één keer per minuut.[1] Toch is slikken niet vanzelfsprekend. Een stoornis bij het slikken, oftewel dysfagie, zorgt ervoor dat patiënten niet actief kunnen deelnemen aan de alledaagse sociale handelingen, wat de kwaliteit van leven verlaagt. Bovendien kan dysfagie leiden tot ondervoeding, uitdroging, longontsteking of zelfs overlijden. Naar schatting lijdt 6% van de Nederlandse bevolking aan dysfagie en bij ouderen is dit zelfs 23%. Bij patiënten met ernstige aandoeningen in het hoofd-halsgebied, zoals keelkanker of na beroertes, lijdt naar schatting meer dan 50% aan dysfagie.[2]

Voor de behandeling van dysfagie is het belangrijk dat het wordt gegradueerd. De huidige methode om dysfagie te gradueren is videofluoroscopie. Echter, nadelen hiervan zijn dat de patiënt blootgesteld wordt aan schadelijke röntgenstraling, weke delen niet goed zichtbaar zijn, de beoordeling subjectief is [3] en het beoordelen van de beelden arbeidsintensief is, omdat zowel een radioloog als een logopedist de beelden moet beoordelen.[4]

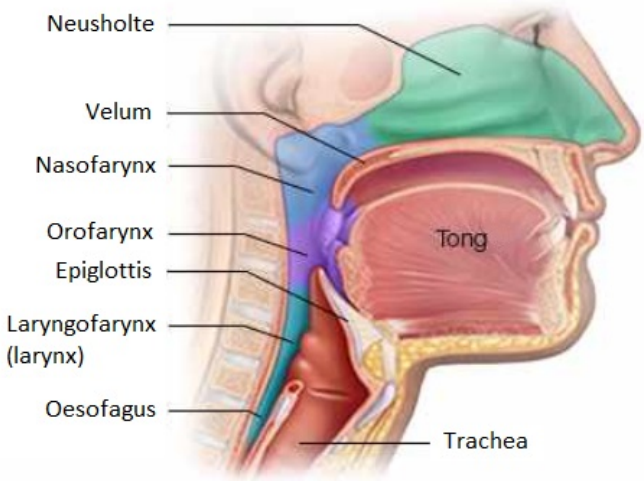


Figuur 1: *De fasen van de slikbeweging: (1) de orale voorbereidende fase, (2) de orale transport fase, (3-5) de faryngeale fase en (6) de oesofageale fase.[1]*

1.1 Klinische achtergrond

1.1.1 Fysiologie van de slikbeweging

Slikken is een complexe beweging die opgedeeld kan worden in vier fasen (zie figuur 1): de orale voorbereidende fase, de orale transportfase, de faryngeale fase en de oesofageale fase.[1] Tijdens de orale voorbereidende fase wordt de bolus mechanisch door tanden en chemisch door speeksel, gedeeltelijk afgebroken. De bolus wordt met de tong en de kaakspieren heen en weer bewogen in de mond tot deze glad genoeg is voor de volgende fase. Vervolgens wordt tijdens de orale transportfase de bolus richting de orofarynx verplaatst, doordat de posterior tong omlaag beweegt en het velum (zie figuur 2) de nasofarynx afsluit. Tijdens de daaropvolgende faryngeale fase worden alle alternatieve routes van de bolus afgesloten: de nasofarynx wordt afgesloten door het velum, de mondholte door de tong en de trachea door de epiglottis, waardoor de bolus enkel richting de oesofagus kan bewegen. De propulsie



Figuur 2: *De anatomie van het hoofd-halsgebied.[7]*

van de bolus vanaf de orofarynx naar de oesofagus, waarbij ruim 25 spiergroepen betrokken zijn, is de meest complexe beweging van de gehele slikbeweging. De laatste fase is de oesofageale fase. Deze begint wanneer de bolus de bovenste oesofageale sfincter heeft gepasseerd. Vanaf hier wordt de bolus middels peristaltische bewegingen van de oesofagus naar de maag verplaatst.[5][6]

1.1.2 Dysfagie

Dysfagie wordt gedefinieerd als een gevoel van obstructie in de passage van vloeistof of vast voedsel door de farynx of oesofagus.[8] Door de complexiteit van de slikbeweging kan dysfagie diverse oorzaken hebben (zie tabel 1). De kenmerken van dysfagie kunnen dus ook sterk verschillen per patiënt.

Tabel 1: *Voorbeelden van oorzaken van dysfagie.[8][1]*

Categorie	Voorbeeld
Ziekte aan mond of tong	Tonsillitis
Aandoeningen die de motiliteit van de oesofagus beperken	Fouten in dwarsgestreept spierweefsel Achalasia [9] Paralyse van de faryngeale constrictoren [10] en spiervezels [10]
Externe druk op de oesofagus	Goitre
Neurologische aandoeningen	Wilson's disease [9]
Intrinsieke laesies	Oesofageale webben Carcinomen en de behandeling ervan

1.2 Diagnostiek en therapie

1.2.1 Huidige diagnostiek en therapie

De gouden standaard voor dysfagiegradatie is videofluoroscopie. Wanneer bij een patiënt een pathologie in het hoofd-hals gebied is vastgesteld, wordt de patiënt doorverwezen naar logopedisten gespecialiseerd in dysfagie. Deze logopedisten nemen een anamnese af, verrichten lichamelijk onderzoek, maken eventuele spraak- en stemopnames en maken vervolgens meerdere videofluoroscopische slikvideo's.[11][2] Terwijl de patiënt een barium mengsel doorslikt, wordt de slikbeweging vastgelegd met een snelheid van

minimaal 30 frames/seconde.[3] De röntgendoorlichting vindt zowel vanaf de zijkant (latero-lateraal), als van voren (posteroanterior) plaats. Per video wordt het volume van de bolus (3cc of 10cc), de consistentie van de bolus (vast of vloeibaar voedsel) of de stand van het hoofd van de patiënt gevarieerd.[11]

Behandelplannen worden in een multidisciplinair team van logopedisten, radiotherapueten, hoofd-halschirurgen en internisten per patiënt opgesteld aan de hand van de dysfagiegradatie en de fysiologische problemen die de dysfagie veroorzaken.[12] Indien er een tumor bevindt die niet in de buurt van een kritisch gebied zit is chirurgie een optie. Als dit wel het geval is, wordt gekozen voor radiotherapie, chemotherapie, immunotherapie, bioradiatie, fotodynamische therapie of een combinatie van deze therapieën.[12] Indien door bestraling de bovenste slokdarmsfincter rigide is geworden, is botox of dilatatie van de slokdarm mogelijk.[11] [13] Soms kan de oorzaak ook medicamenteus behandeld worden.[2] Deze behandelingen leiden echter soms tot verergering van dysfagie. Naast de hiervoor genoemde behandelingen kan een diëtist het dieet aanpassen, kan een logopedist compensatiestrategieën aanbieden voor tijdens het slikken en oefeningen geven om hoofd-halsspieren te versterken.[2] [11][14]

1.2.2 Nadelen videofluoroscopie

Een belangrijk nadeel van videofluoroscopie is dat de beoordeling op dit moment erg subjectief is met een lage *interobserver agreement*.[11] Het geven van een objectieve beoordeling is zeer moeilijk, omdat er geen kwantitatieve metingen gedaan kunnen worden.[15] Een van de weinige objectieve maten is de PAS-score. Deze heeft een hoge *inter-* en *intraobserver agreement*, maar geeft bijvoorbeeld geen informatie over kuchen, hoe lang iemand doet over de complete passage van de bolus of eventueel faryngeaal residu.[16][11] Daarnaast is het beeld van videofluoroscopie een 2D-opeenstapeling van schaduwen, waardoor informatie uit de laterale dimensie mist. Ook zijn weke delen niet zichtbaar. Ten slotte wordt gebruik gemaakt van schadelijke röntgenstraling en kan dus per patiënt maar in beperkte mate onderzoek verricht worden.[12]

1.2.3 Dynamische Magnetic Resonance Imaging

Een veelbelovend alternatief voor videofluoroscopie is dynamische *Magnetic Resonance Imaging* (dMRI). DMRI maakt gebruik van niet-schadelijke magneetvelden.[12] Aangezien MR-beelden driedimensionaal zijn, kan ook de laterale dimensie gebruikt worden om meer informatie te verkrijgen over de oorzaak van dysfagie, de bijbehorende locatie, de afmetingen en de omliggende structuren.[11][12] Zo kan bijvoorbeeld pre-operatief beter afgebeeld worden of een tumor zich naar een kritisch gebied uitbreidt en kan postoperatief littekenweefsel beter bekeken worden. [12][15] 3D-beelden kunnen ook meer inzicht geven in de beweeglijkheid van weke delen.[12] DMRI biedt de mogelijkheid om objectieve aspecten te meten, wat op dit moment lastig is met videofluoroscopie. Een volumemeting zou voor een betere gradatie zorgen.[11][12] Belangrijke weke delen voor de slikbeweging, met name de tongrug en de tongbasis, zijn met MRI goed in beeld te brengen.[15][3] Gebruikmakend van dMRI kan wellicht een andere graduering van slikstoornissen bereikt worden en kan het daarop gebaseerde beleid preciezer gemaakt worden.[12]

Om een dMRI video te maken, worden meerdere MR-beelden achter elkaar gemaakt.[17] Het maken van een volledige MR-afbeelding duurt circa 1 seconde, wat een temporele resolutie van 1 frame/seconde geeft.[18] Wanneer tijdens deze seconde bewogen wordt, wordt het gemiddelde van deze beweging afgebeeld. Dit principe is te vergelijken met *motion blur* bij fotografie. De afbeelding is hierdoor vaag en maakt dysfagiegraduering moeilijk. Om de slikbeweging wel goed vast te kunnen leggen, is een temporele resolutie van minimaal 30 frames/seconde gewenst.[4]

Undersampling is een manier om de temporele resolutie van een MRI scanner te verhogen door delen van de *k-space* niet te vullen tijdens data- acquisitie. Met behulp van *compressive sensing* kunnen deze lege plekken opgevuld worden. Luuk Voskuilen, MSc schrijft momenteel een algoritme dat middels NUiFFT en *compressive sensing* bruikbare MR-beelden kan reconstrueren uit *undersampled k-space*. Dit

reconstrueren duurt ongeveer een halfuur per dMRI-video. Om de functie van dit algoritme over te nemen en de reconstructie van MR-beelden te versnellen en verbeteren, kan mogelijk gebruik gemaakt worden van een *deep learning* (DL) netwerk.

1.3 Doelstelling

Om een geschikt behandelplan op te stellen voor dysfagiepatiënten is het noodzakelijk dat de gradatie van dysfagie verbeterd wordt. Het moet mogelijk worden om kwantitatieve metingen uit te voeren en 3D-informatie te krijgen over structuren in het hoofd-halsgebied, zonder schadelijke röntgenstraling te gebruiken. DMRI in combinatie met DL lijkt een goed alternatief voor videofluoroscopie voor dysfagiegradatie. Voordat dMRI in de kliniek toegepast kan worden, is nog veel onderzoek nodig. In dit onderzoek wordt de eerste stap hierin gezet: testen of een DL-netwerk in staat is MR-beelden met hoge spatiële resolutie te reconstrueren vanuit een *undersampled k-space*. Wanneer deze stap succesvol is afgerond, is de volgende stap om de belangrijke klinische aspecten in kaart te brengen. Deze aspecten, waarmee dysfagie nu gegradueerd wordt, moeten te zien zijn op de gereconstrueerde dMR-beelden. Vervolgens kan gekeken worden naar nieuwe aspecten waarmee dysfagie gegradueerd kan worden met behulp van dMRI en DL. In dit onderzoek is onvoldoende tijd voor de implementatie van deze klinische aspecten in het DL-netwerk, maar door middel van interviews en literatuuronderzoek is een inventarisatie gemaakt van de belangrijkste huidige en toekomstige klinische aspecten voor dysfagiegradatie.

De hoofdvraag van dit onderzoek luidt daarom: 'hoe kunnen dynamische MR-beelden van slikbewegingen met behulp van *deep learning* gereconstrueerd worden vanuit *undersampled k-spacedata* en welke klinische aspecten zijn relevant om in de toekomst dysfagie te gradueren met behulp van dynamische MRI?'

1.4 Leeswijzer

Dit verslag bestaat uit twee verschillende onderdelen: een technisch en een medisch deel. Vanuit technisch oogpunt zal onderzocht worden of het mogelijk is MR-beelden te reconstrueren uit *undersampled k-spacedata*. Hiervoor wordt eerst een **technische achtergrond** gegeven om de begrippen *k-space, (under)sampling* en *compressive sensing* uit te leggen. Daarna wordt uitgelegd wat DL is, hoe een DL-netwerk getraind wordt en wat voor netwerken er zijn. Na deze achtergrond wordt in het hoofdstuk **Methode** het principe van ADMM-net, de data-acquisitie en de verificatiemethoden van het netwerk uitgelegd. Vervolgens worden op basis van literatuur en trainingen de beste instellingen en datasets voor de definitieve trainingen bepaald. In de **Resultaten** worden de resultaten van deze definitieve trainingen gepresenteerd, die in de **Discussie** worden besproken, evenals onverwachte bevindingen en beperkingen van dit onderzoek. Aan de hand hiervan wordt een **conclusie** gegeven en worden aanbevelingen gedaan voor verder onderzoek.

In het tweede deel wordt in het hoofdstuk **Klinische aspecten** vanuit medisch oogpunt onderzocht wat de belangrijkste aspecten zijn waarmee op dit moment dysfagie wordt gegradueerd. Om deze aspecten in kaart te brengen, is literatuuronderzoek uitgevoerd en zijn interviews gehouden met specialisten. Vervolgens zal in een **algehele discussie en conclusie** geëvalueerd worden op de vraag vanuit de kliniek en het aanbod vanuit de techniek. Hier zal het gehele onderzoek besproken, zal de onderzoeksvraag beantwoord worden en worden aanbevelingen gedaan voor het gebruik van dMRI in combinatie met DL voor dysfagiegradatie.

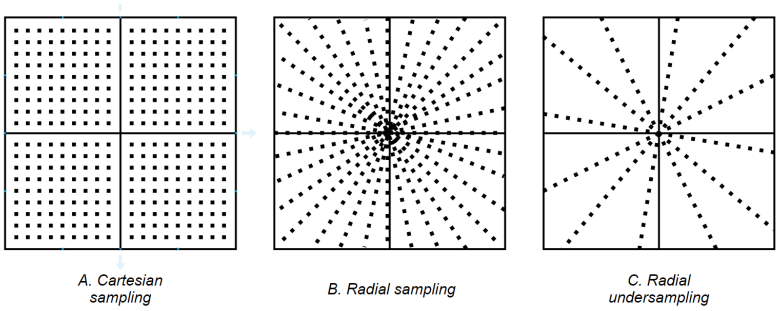
2 Technische Achtergrond

2.1 K-space

Uit elektromagnetische golven die door ontvangstspoelen in de MRI-scanner opgevangen worden, wordt een 3D *k-spacematrix* gevormd. Dit is een rooster van ruwe data in het frequentiedomein in de vorm (k_x, k_y, k_z). Elk punt in de *k-space* correspondeert met een frequentie in het spatiële domein en bevat fase- en amplitude-informatie over alle pixels in de uiteindelijke afbeelding.[19] De *k-space* is geordend op frequentie: het midden van het rooster bevat de lage frequenties en de randen van het rooster bevatten de hoge frequenties. De lage frequenties bevatten informatie over globale aspecten van de afbeelding, zoals het contrast, terwijl de hoge frequenties vooral informatie over lokale aspecten bevatten, zoals details in de afbeelding. Ook bevat het midden van het rooster de hoogste signaalamplitude. Het midden van het rooster is om deze redenen het belangrijkst voor de kwaliteit van de afbeelding.[20]

2.2 Sampling methoden

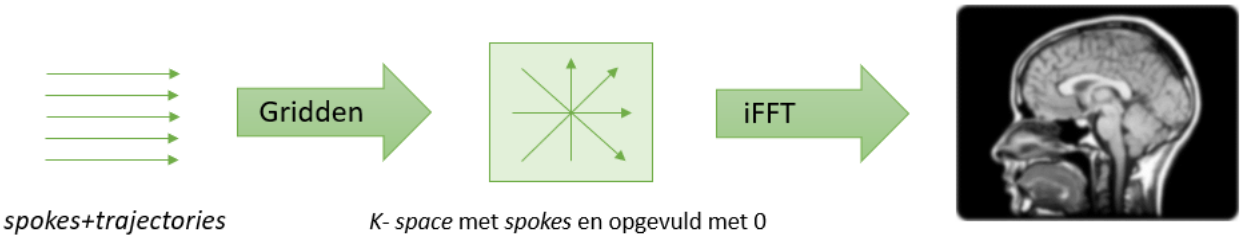
Normaliter wordt de *k-space* gevuld door middel van cartesische *sampling* (zie figuur 3a), wat inhoudt dat de *k-space* rij voor rij wordt gevuld. Een andere manier om de *k-space* te vullen is radiale *sampling* (figuur 3b) waarbij de *k-space* wordt gesampled in zogenaamde *spokes*. Iedere *spoke* heeft een *trajectory* die de locatie van deze *spoke* in de ruimte van de k-space geeft.[21][22] Omdat het midden van de *k-space* de belangrijkste informatie bevat, is het belangrijk om dit te behouden bij *undersampling*. Bij radiale *sampling* gaan er meerdere spokes door het midden, waardoor de informatiedichtheid daar groter is. Bij *undersampling* zal daardoor minder laagfrequente informatie verloren gaan bij radiale dan bij cartesische *sampling*. Daarom is radiale *sampling* meer geschikt als men wil *undersamplen* (zie figuur 3c).



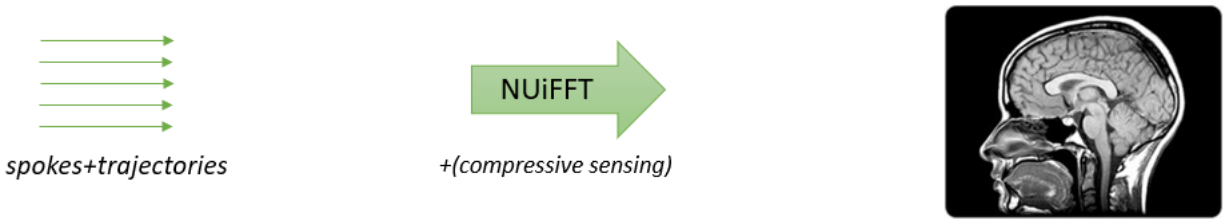
Figuur 3: *Voorbeelden van sampling methoden.*

2.3 Undersampling

Undersampling is een manier om de temporele resolutie van een MRI-scanner te verhogen door minder informatie in de *k-space* te acquireren tijdens MRI-opnames (zie figuur 3c). Een algemene manier om *k-space* om te zetten naar MR-beelden is het uitvoeren van een inverse *fast* fouriertransformatie (iFFT). Een iFFT is echter alleen mogelijk bij een *fully sampled k-space*.[23] Er zijn meerdere manieren om een *undersampled k-space* toch te transformeren naar een MR-beeld, namelijk *gridding* met iFFT (zie figuur 4) of *Non-uniform inverse fast fourier transform* (NUiFFT) met eventueel *compressive sensing* (zie figuur 5). Bij *gridding* worden de *spokes* en *trajectories* geplaatst op een rooster. Vervolgens kunnen de lege plekken in de *k-space* opgevuld worden met bijvoorbeeld nullen (*zero filling*). Dit resulteert in een *fully sampled k-space* met veel nullen. Deze *fully sampled k-space* kan vervolgens door middel van een iFFT omgezet worden naar een MR-beeld. Het nadeel van *gridding* met *zero-filling* is dat er artefacten ontstaan door het invullen van lege delen met nullen. Met NUiFFT kan wel een fouriertransformatie uitgevoerd worden op een *undersampled k-space* bestaande uit alleen *spokes* en *trajectories*. Het voordeel van NUiFFT is dat het opvullen van het rooster gedaan kan worden door *compressive sensing*, wat minder artefacten geeft.



Figuur 4: *Omzetting van spokes en trajectories naar een MR-afbeelding door gridding, zero-filling en een iFFT.*



Figuur 5: *Omzetting van spokes en trajectories in de k-space naar een MR-afbeelding met behulp van NUiFFT en compressive sensing.*

2.4 Compressive sensing

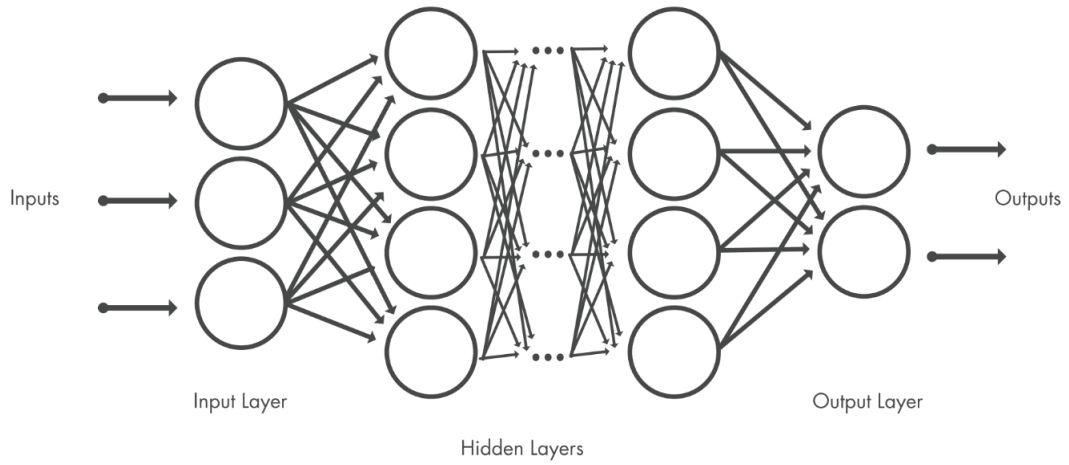
Met behulp van *compressive sensing* kunnen de lege delen van de *k-space* opgevuld worden. Het opvullen gebeurt op basis van informatie die bekend is over een specifiek gebied.[24] L. Voskuilen, MSc, heeft een algoritme geschreven om uit *undersampled k-space* bruikbare MR-beelden te krijgen middels NUiFFT en *compressive sensing*. Reconstrueren met behulp van dit script duurt ongeveer een halfuur per dMR-video. Om dit proces te versnellen kan mogelijk gebruik gemaakt worden van een *deep learning* netwerk. Deze zou de functie van dit algoritme kunnen overnemen door ook een vorm van *compressive sensing* en een fouriertransformatie toe te passen. Zo zou dit DL-netwerk in enkele seconden *undersampled k-spacedata* om kunnen zetten in MR-beelden van hoge kwaliteit.

2.5 Deep learning

Deep learning (DL) is een tak binnen de kunstmatige intelligentie waarin een kunstmatig neuraal netwerk getraind wordt om een bepaalde taak uit te voeren.[25] Een DL-netwerk bevat verschillende lagen tussen de input- en outputlaag, zogenaamde *hidden layers*. Deze layers bestaan uit *nodes*(zie figuur 6). Tijdens de training van het netwerk wordt bij het doorgeven van informatie aan de *nodes* van de daaropvolgende laag aan iedere *node* een bepaalde *weight* gegeven. Op deze manier functioneert het netwerk vergelijkbaar met een biologische neuraal netwerk, waarbij elke neuron verbonden is met omliggende neuronen door middel van axonen en dendrieten.[26] De term *'deep'* slaat op het grote aantal *hidden layers* tussen de input en output.

Training

Het doel van neurale netwerken is het leren herkennen van kenmerken in de input en de correlatie met de output te zien, zodat het bij een nieuwe, nog onbekende input, de juiste output kan genereren.[28] Voordat een netwerk in staat is tot het vinden van deze correlatie moet het getraind worden. Dit trainen kan op drie manieren: *unsupervised, semi-supervised* en *supervised*.[29] Wanneer er geen gelabelde trainingsdata - corresponderende output bij de input[30] - beschikbaar is, heet dit *unsupervised* training. Deze methode is vaak complex en tijdsintensief, omdat het netwerk zelf een relatie tussen de data moet definiëren. Wanneer wel gebruik wordt gemaakt van gelabelde data is er sprake van *supervised* training. Deze manier

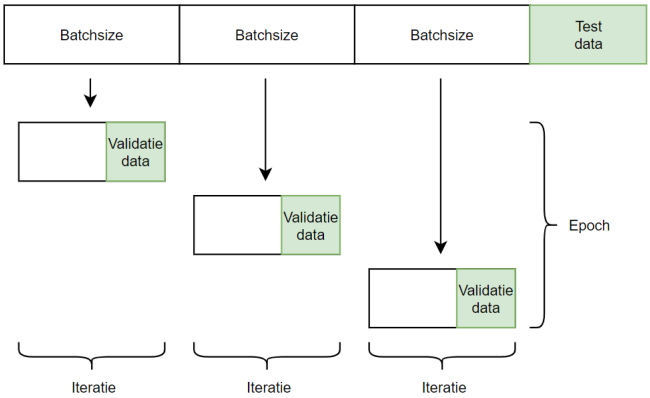


Figuur 6: *De hidden layers en nodes van een artificieel neuraal netwerk.[27]*

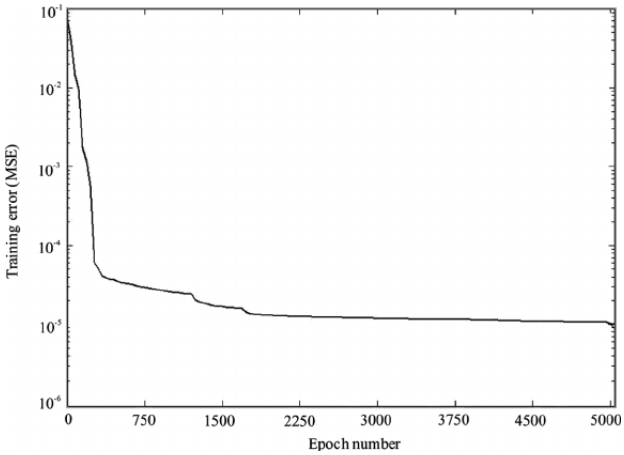
van training wordt vaak gebruikt in klinisch onderzoek, omdat gelabelde data vaak beschikbaar is door de beoordeling van specialisten en omdat training minder tijdsintensief is.[31] *Semi-supervised* trainen is een combinatie van *supervised* en *unsupervised* trainen. Hierbij wordt een klein gedeelte van de trainingsdata gelabeld. Het netwerk deelt vervolgens de overige data in, met de gelabelde data als voorbeeld. Het voordeel van deze methode is dat het netwerk zelf de grenzen tussen verschillende labels kan bepalen en er minder bias optreedt bij dit proces. Ook is deze methode minder arbeidsintensief, omdat er minder data gelabeld hoeft te worden.[26] Naast dat een leeg netwerk getraind kan worden op de hiervoor beschreven manieren, kan ook een al getraind netwerk op deze manier verder getraind worden. Dit heet *transfer learning* en is vaak sneller en makkelijker dan het trainen van een nog ongetraind netwerk.[32]

Een training bestaat uit iteraties, waarin de trainingsdata aan het netwerk wordt gepresenteerd. Een deel van deze data wordt gebruikt voor training van het netwerk en het andere deel wordt gebruikt voor validatie[33](zie figuur 7). Het aanpassen van de *weights* wordt gedaan door middel van *back propagation*: met behulp van optimalisatie-algoritmes voorspelt het netwerk een output, die vervolgens wordt vergeleken met de bekende output. Na elke iteratie worden de *weights* bij de *nodes* aangepast, waardoor het verschil tussen de DL-output en de referentie-output steeds kleiner wordt en het netwerk nauwkeuriger wordt. De mate waarin de *weights* worden aangepast, wordt de *learning rate* genoemd. Als de *learning rate* laag is, zullen de *weights* weinig aangepast worden, terwijl de *weights* bij een hoge *learning rate* in grotere mate aangepast zullen worden.[34] Als alle data één keer aan het netwerk is gepresenteerd, spreekt men van een *epoch*.[26] Bij voldoende trainingsdata zal het verschil tussen de DL-output en de referentie output naderen naar 0 en zal het netwerk een nauwkeurige voorspelling kunnen maken van de output.[35] Als het netwerk vervolgens, nog onbekende testdata als input krijgt, zal het netwerk zelf de juiste output kunnen genereren.

Bij het trainen van een netwerk moet opgelet worden dat er geen *overfitting* optreedt. Bij *overfitting* correspondeert het netwerk zeer nauw of exact met de trainingsdata, waardoor het model niet gegarandeerd een goede voorspelling van de output kan geven als het een nieuwe, onbekende, dataset gepresenteerd krijgt.[26] *Overfitting* is te voorkomen door de training te stoppen wanneer de grafiek van afwijking als functie van epochnummer een plateau bereikt (zie figuur 8). Wanneer *overfitting* optreedt, is verschillend per netwerk en zal dus getest moeten worden.



Figuur 7: *Schematische weergave van de begrippen iteratie, epoch, batchsize, test data en validatie data.*



Figuur 8: *Een voorbeeld van een grafiek van afwijking als functie van epochnummer [36]*

Classificatie en Regressie

Een DL-algoritme kan getraind worden op twee soorten algemene problemen: classificatie- en regressieproblemen.[34] Bij classificatieproblemen wordt het netwerk getraind op inputs van verschillende categorieën en moet het vervolgens de input categoriseren. Een voorbeeld hiervan is het herkennen van voorwerpen in een afbeelding en deze voorwerpen benoemen.[37] Voorbeelden van al bestaande getrainde netwerken zijn AlexNet, GoogleNet , ResNet, CifarNet, Overfeat, VGG en LeNet.[38][39][40][41] Bij regressieproblemen dient het netwerk een kwantitatieve waarde te voorspellen door middel van algoritmes.[34] Voorbeelden van netwerken die dergelijke problemen op kunnen lossen zijn U-net[42], GP-Unet[43] en ADMM-Net[44].

2.6 Deep Learning en dynamische MRI

Het gebruik van DL voor dMRI is een actueel onderwerp binnen verschillende medische specialismen voor dezelfde reden, namelijk het versnellen van MRI-opnames.[45][46][47] Er is hier echter nog weinig onderzoek naar gedaan en het wordt dan ook nog niet geïmplementeerd in de kliniek. Het gebruik van DL bij MRI-reconstructie, met *undersampled k-spacedata* als input en de afbeelding in het tijdsdomein als output, is een regressieprobleem. Er is daarom gezocht op regressienetwerken in Google Scholar, Scopus, GitHub en Mathworks. Omdat onze voorkeur uitging naar een netwerk dat gemaakt is voor *k-spacedata* als input en vervolgens middels een inverse fouriertransformatie en *compressive sensing* een output kan genereren, is gekozen voor ADMM-net.

2.6.1 ADMM-net

ADMM-net is gebaseerd op het ADMM-algoritme (*Alternating Direction Method for Multipliers*).[44] Dit is een variant van het *augmented Lagrangian scheme*, die tweevoudige variabelen opdeelt in subgroepen die afzonderlijk opgelost kunnen worden. Het *augmented Lagrangian scheme* kan gebruikt worden voor bijvoorbeeld het minimaliseren van een functie $f(x)$ onder de randvoorwaarde dat $c_i(x) = 0$ voor $i \in I$. Deze randvoorwaarde wordt in het *augmented Lagrangian scheme* vervangen door twee extra termen in de te minimaliseren functie, zoals gegeven in functie 1.

$$\min_x \left\{ f(x) - \sum_{i \in I} \lambda_i c_i(x) + \frac{\mu_k}{2} \sum_{i \in I} c_i(x)^2 \right\} \tag{1}$$

Functie 1 zoekt naar het minimum van de functie tussen de accolades. De eerste term $f(x)$ is de te minimaliseren functie, de tweede term is de *Lagrangian multiplier term* en de derde term is de *penalty term*, ofwel de boeteterm . Bij elke iteratie wordt λ_i aangepast en wordt μ_k groter, waarbij λ_i afhankelijk is van μ_k. Door de toenemende grootte van μ_k worden de termen $c_i(x)$ naar nul gedreven. Het algemene *compressive sensing* model voor MRI wordt beschreven door functie 2.

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2} \|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(D_l x) \right\} \tag{2}$$

De te minimaliseren functie $f(x)$ is nu een functie die bestaat uit twee termen, beide afhankelijk van x. In het geval van dit onderzoek is x de gewenste output, ofwel de referentie-afbeelding, y de input, ofwel de *undersampled k-space* en x̂ is de output van het DL-netwerk. Voor A geldt $A = P * F$ met P een masker en F een fouriertransformatie. Dit betekent dat het MR-beeld 'x' fouriergetransformeerd wordt en vervolgens vermenigvuldigd wordt met het masker. Bij de minimalisatie wordt y zoveel mogelijk gelijk gemaakt aan Ax. In de ideale situatie is y daarom gelijk aan Ax. Verder geldt dat λ_l een regularisatieparameter is, D_l een transformatiematrix voor een filterbewerking is en g een regularisatiefunctie afgeleid van de voorafgaande gegevens is. Dit optimalisatieprobleem zal het ADMM-net moeten oplossen. Wanneer in plaats van een enkele functie zoals in functie 1 er twee functies geminimaliseerd moeten worden, kan het minimalisatieproces versneld worden door functie 2 om te schrijven naar functie 3.

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(z_l) \tag{3}$$
$$\textit{onderhevig aan } z_l = D_l x, \forall\, l \in\ 1, 2, ..., L$$

In de eerste term rechts van het =-teken moet x geminimaliseerd worden en in de tweede term moet z_l geminimaliseerd worden. $D_l x$ in functie 2 is vervangen door z_l met als randvoorwaarde dat z_l gelijk moet worden aan $D_l x$. Functie 3 is een functie met zowel duale variabelen als randvoorwaarden. Een dergelijke functie is met het *augmented Lagrangian scheme* goed te minimaliseren. Daarom wordt functie 3 omgeschreven in een functie zonder randvoorwaarden; zijn augmented Lagrangian functie:

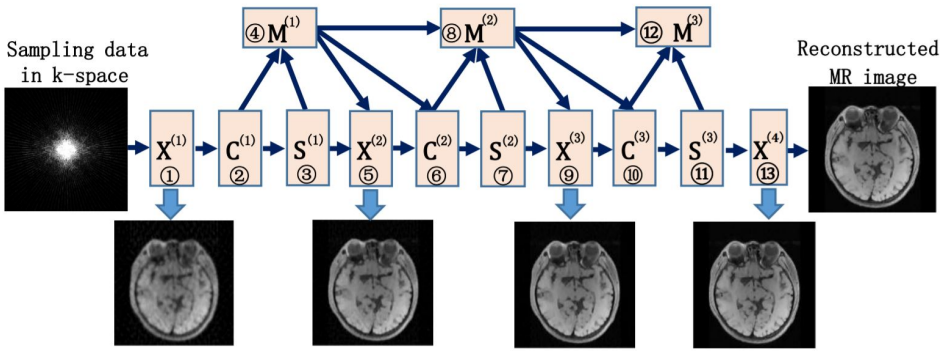
$$\mathcal{L}_\rho(x, z, \alpha) = \frac{1}{2} \|Ax - y\|_2^2 + \sum_{l=1}^{L} [\lambda_l g(z_l) + \langle \alpha_l, D_l x - z_l \rangle + \frac{\rho_l}{2} \|D_l x - z_l\|_2^2] \tag{4}$$

De randvoorwaarde $z_l = D_l x$ voor $i \in I$ is verwerkt in de derde en vierde term rechts van het =-teken. Dit zijn respectievelijk een Lagrangian vermenigvuldigingsterm (met α_l als *Lagrangian multiplier*) en een Lagrangian boeteterm (met ρ_l als boeteparameter). Deze boete term dwingt z_l om dichter naar $D_l x$ te gaan bij iedere iteratie. Na elke iteratie wordt rho_l groter en dus ook de boeteterm. De boeteterm weegt daardoor iedere volgende iteratie zwaarder mee, waardoor z_l meer gedwongen wordt om nog dichter naar $D_l x$ te gaan. Echter, indien alleen de boeteterm in functie 4 zou staan, zou dit proces veel iteraties vergen voor convergentie. Om het proces te versnellen staat in functie 4 als derde term rechts van het =-teken een vermenigvuldigingsterm. Het minimalisatieprobleem heeft nu niet langer een randvoorwaarde en z_l hoeft niet verplicht gelijk te zijn aan $D_l x$, maar na meerdere iteraties gebeurt dit vanzelf. Bij de minimalisatie van functie 4 naar x en z worden niet alleen de eerste en tweede term geminimaliseerd, ook de derde en de vierde term minimaliseren. Hierdoor wordt z_l gelijk aan $D_l x$. Dit geeft dan dezelfde oplossing als de minimalisatie van functie 2 naar x. Het probleem van de minimalisatie van functie 4 kan in het ADMM-algoritme vereenvoudigd worden door het te herschrijven als drie parallelle subproblemen:

$$\begin{cases} \arg\min_x \frac{1}{2} \|Ax - y\|_2^2 + \sum_{l=1}^{L} \frac{\rho l}{2} \|D_l x + \beta_l - z_l\|_2^2], \\ \arg\min_z \sum_{l=1}^{L} [\lambda_l g(z_l) + \frac{\rho l}{2} \|D_l x + \beta_l - z_l\|_2^2], \\ \arg\max_\beta \sum_{l=1}^{L} \langle \beta_l, D_l x - z_l \rangle. \end{cases} \tag{5}$$

Voor de eenvoud is de vermenigvuldigingsterm in de maximalisatie van functie 5 geschaald met $\beta_l = \frac{\alpha_l}{\rho_l}$. In het eerste subprobleem wordt de $Ax - y$ geminimaliseerd naar variabele x met een boetefunctie, het tweede subprobleem minimaliseert $\lambda_l g z_l$ naar variabele z met een boetefunctie en het derde subprobleem maximaliseert de Lagrange vermenigvuldigingsterm naar variabele β_l. Na een heel aantal iteraties zijn de eerste twee functies geminimaliseerd en de derde functie gemaximaliseerd. Dit geeft dezelfde oplossing als functie 4.

ADMM minimaliseert in plaats van twee functies met een randvoorwaarde een aantal functies zonder randvoorwaarde. Een groot voordeel van ADMM is dat deze tweevoudige variabelen tegelijkertijd opgelost kunnen worden en dat daardoor het algortime zeer snel is.[48][49]



Figuur 9: *Schematische weergave van een netwerk met 3 stages.[49]*

Met dit algoritme als basis is het ADMM-net gecreëerd. Het ADMM-net 'leert' hoe het van *k-space* een afbeelding moet maken door tijdens de training de *weights* in zijn *layers* aan te passen. Het trainen gebeurt middels een *stochastic gradient descent optimization* algoritme, specifiek L-BFGS, en de *weights* worden aangepast door middel van *back propagation*. Het netwerk bestaat uit verschillende *stages*: een eerste *stage*, een aantal midden *stages* en een laatste *stage*. Eén *stage* bestaat uit 4 verschillende lagen: een *reconstruction layer* X(n), een *convolution layer* C(n), een *nonlinear transform layer* S (n) en een *multiplier update layer* M(n).[44] Een voorbeeld van het netwerk met 3 *stages* is te zien in figuur 9. Naast deze lagen zijn er ook een *input*- en een *outputlayer* aanwezig. Het totale aantal lagen waaruit het netwerk bestaat is dus afhankelijk van het aantal *stages* en is gelijk aan

$$aantal\ lagen\ van\ het\ netwerk = 4 * aantal\ stages + 2.$$

3 ADMM-net: Een proof-of-concept

3.1 Methode

Om een DL-netwerk MR-beelden met hoge spatiële resolutie te laten reconstrueren uit *undersampled k-space*, zullen de hierna beschreven stappen gezet worden. De proefpersonendata van Voskuilen zal eerst verwerkt worden tot data waarmee het initieel netwerk getraind kan worden, namelijk *undersampled k-space* data verkregen middels een masker en *undersampled k-space* verkregen uit *spokes en trajectories*. Het netwerk dat voor deze DL-toepassing getraind zal worden is het ADMM-net[49], geschreven in Matlab (R2019a versie 9.6.0). De scripts en bestanden die bij dit netwerk horen, zijn gedownload via GitHub.com[50]. Voor het trainen van het netwerk zal gebruik gemaakt worden van een GPU (GeForce GTX 960) en het script *main_netTrain.m* (zie bijlage B). Vervolgens zal een aantal trainingen uitgevoerd worden om de juiste instellingen voor de definitieve training te bepalen. Er zal bovendien bepaald worden welk netwerk met ADMM-net getraind zal worden en welke input beter is(*undersampled k-space* of *fully sampled k-space*). Ten slotte zal ook bepaald worden of variaties aangebracht moeten worden in de trainingsopties, zoals de *learning rate* en de hoeveelheid trainingsdata. Na iedere iteratie zal een netwerk met aangepaste *weights* opgeslagen worden. De resultaten van deze netwerken zullen met elkaar vergeleken worden om de beste instellingen vast te stellen en zo de definitieve training samen te stellen.

3.1.1 Data-acquisitie

De data die nodig is voor het trainen, is afkomstig van zes gezonde proefpersonen en is verzameld door Voskuilen. De data betreft 315 tijdframes met 2D MRI-beelden (Philips, 3 tesla) per proefpersoon. Deze data is *undersampled* geacquireerd met acht *spokes* per frame. Voskuilen heeft met deze *undersampled* data en een door hem geschreven algoritme MR-beelden gereconstrueerd. Deze MR-beelden zullen in dit onderzoek gebruikt worden als referentie-afbeeldingen. Deze referentie-afbeeldingen kunnen door middel van een fouriertransformatie omgezet worden naar gereconstrueerde *fully sampled k-space* data.

Undersampled k-space data

Om het gewenste DL-netwerk te krijgen, moet het initiële netwerk getraind worden met *undersampled k-spacedata*. Er zal getraind worden met twee soorten *undersampled k-spacedata*: *undersampled k-spacedata* verkregen met behulp van een masker en *undersampled k-space* data verkregen uit *spokes* en *trajectories*.

Undersampled k-space data door middel van een ADMM-net masker

Om *undersampled k-space* te simuleren kan de gereconstrueerde *fully sampled k-space* vermenigvuldigd worden met een masker: een matrix bestaande uit enen en nullen. Zoals beschreven in de inleiding is radiale *undersampling* een betere methode dan cartesische *undersampling*, omdat hierbij minder belangrijke informatie verloren gaat. Het doel is dus om een masker te creëren waarmee een radiale *undersampled k-space* gesimuleerd kan worden. In de bestanden van het ADMM-net zijn pseudoradiale maskers beschikbaar met *sampling rates* van 10%, 20%, 30%, 40% of 50%, zodat respectievelijk 10%, 20%, 30%, 40% of 50% van de oorspronkelijke data behouden blijft. Het begrip pseudoradiaal houdt in dat het masker radiaal lijkt, maar cartesisch is geacquireerd. Hoe deze data is verkregen is te zien in bijlage A: Train data van undersampled k-space met 20% masker.

Undersampled k-space data door middel van spokes en trajectories

De oorspronkelijke *undersampled k-space* data is beschikbaar in de vorm van *spokes* en *trajectories*. Deze *spokes* en *trajectories* vormen samen een radiaal *sampling* patroon, dat met behulp van de BART-toolbox© (Berkeley, versie 0.4.04[51][52][53]) door *gridding* en NUiFFT omgezet kan worden in *k-space* data en een corresponderende MR-afbeelding (zie MakenDataSpokTraj.m in bijlage A). Zowel de *fully sampled*

k-space als de corresponderende afbeeldingen zijn opgeslagen als 2D-data met de tijd als derde dimensie. Hoe deze data is verkregen is te zien in bijlage A: Train data van spokes en trajectories middels masker.

Met de Matlab scripts in bijlage A, wordt in een *loop* ieder tijdframe opgeslagen als een data *struct* waarin een *k-spaceframe* met bijbehorende MR-beeld staat. Deze *structs* dienen tijdens het trainen als traindata.

3.1.2 Verificatie van het netwerk

Een netwerk kan getest worden met *main_ ADMM_Net_test.m* (zie bijlage B). Hierin kan de gewenste input ingeladen worden om aan het netwerk te presenteren, in ons geval *undersampled k-space* data. Het netwerk zal vervolgens een output genereren bij deze input. Daarna worden in dit script de *loss* en de *peak-signal-to-noise ratio* (PSNR) van de output ten opzichte van de de referentie-afbeelding bepaald. Ook zal er genoemd worden hoe lang het netwerk er over deed om een output te genereren. Dit is belangrijk omdat het sneller moet zijn dan het algoritme van Voskuilen, namelijk 30 minuten voor 315 afbeeldingen.

Loss

De *loss* kwantificeert het verschil tussen de door het netwerk gegenereerde output en de referentie afbeelding. De mate waarin iedere output van het model afwijkt van de bijbehorende referentie afbeelding wordt berekend, dan worden de afwijkingen van de gehele dataset samengevoegd, het gemiddelde wordt genomen en zo blijft er één getal over, de *loss*. Hoe lager de *loss*, hoe beter het netwerk in staat is met de *weights* een output te genereren die lijkt op de referentie afbeelding.[54][55]

PSNR

De *peak-signal-to-noise ratio* (PSNR) is de ratio in decibels tussen het maximale vermogen van een signaal en het vermogen van de storende ruis, zoals te zien in vergelijking 6.

$$PSNR = 10\ log_{10}(\frac{p}{RMS})\ dB \tag{6}$$

Met p de maximum waarde van het signaal en RMS de wortel van de *mean square error* (MSE) tussen de originele afbeelding en de output van het DL-netwerk. Een hogere PSNR indiceert een hogere kwaliteit van de afbeelding.[56] De MSE wordt uitgerekend door het gemiddelde te nemen van de gekwadrateerde intensiteitsverschillen tussen de pixels van de originele afbeelding en de output van het DL-netwerk, zoals te zien in vergelijking 7).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \tag{7}$$

De PSNR is gemakkelijk te berekenen en heeft een duidelijke betekenis[57], maar er zijn meer kwaliteitsmaten nodig om de kwaliteit van een afbeelding te kwantificeren. Twee afbeeldingen kunnen namelijk dezelfde PSNR hebben, maar verschillende kenmerken hebben die in meer of minder mate zichtbaar zijn voor het blote oog.[58]

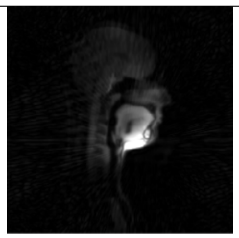
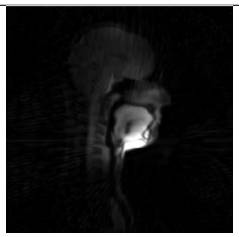
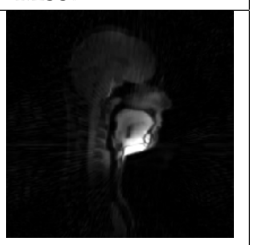
3.1.3 Experimentele opstelling

In het script *main_ADMM_Net_test.m* kunnen er verschillende netwerken getraind worden. Zo is InitNet beschikbaar. Dit netwerk is het initiële netwerk, dat nog ongetraind is en dus nog onaangepaste weights heeft. Naast InitNet zijn nog 3 netwerken aanwezig: net-stage7, net-stage14 en net-stage15. Deze drie netwerken zijn verkregen door InitNet te trainen met als input *undersampled k-spacedata* die verkregen

is met behulp van een 20% pseudoradiaal masker. De netwerken zijn getraind met 100 afbeeldingen. Het verschil tussen de drie netwerken is het gekozen aantal midden stages: 7, 14 en 15 midden stages respectievelijk.

Om te beginnen is nagegaan welke van de drie netwerken, net-stage7, net-stage14 en net-stage15, zonder verdere training het beste resultaat geeft voor de MRI-data. Hiervoor zijn de verschillende netwerken in *main_ADMM_Net_test.m* geladen en zijn ze getest met de *undersampled k-spacedata* van tijdframe 100 uit 20190501.mat. De *undersampled k-spacedata* is verkregen door middel van het 20% pseudoradiaal masker. Op basis van de PSNR en *loss* van deze testen kan geconcludeerd worden dat net-stage15, vóór training met de MRI data van het hoofd- halsgebied, de beste resultaten geeft (zie tabel 2). Bij de volgende testtraining zal daarom gebruik gemaakt worden van net-stage15.

Tabel 2: *Resultaten van het ongetraind testen van de netwerken net-stage7, -14 en -15*

Netwerk	net-stage7	net-stage14	net-stage15
Loss	0.20058	0.15716	0.15077
PSNR in dB	-6.6885	-4.5926	-4.2357
Output			

Vervolgens is zowel net-stage15 als InitNet getraind met *fully sampled k-spacedata* en door het 20% procent masker verkregen *undersampled k-spacedata*. Zie tabel 3 voor de opzet van het onderzoek. Beide zijn hiervoor getraind met 100 frames uit 20190501.mat. Voor dit aantal is gekozen omdat dit aantal groot genoeg is om inzicht te krijgen in de training van de netwerken, maar het zal niet al te veel tijd innemen.

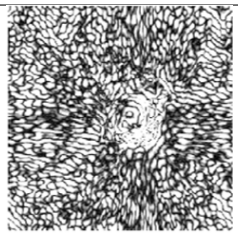
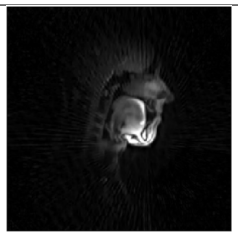
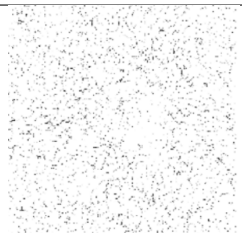
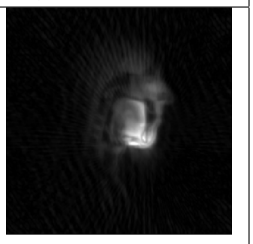
Tabel 3: *Opzet trainingen vóór definitieve training.*

N.	**Netwerk**	**Data.train**	**Data.label**
1.	InitNet	Fully sampled k-space	Referentie afbeelding
2.	InitNet	K-space door masker verkregen	Referentie afbeelding
3.	net-stage15	Fully sampled k-space	Referentie afbeelding
4.	net-stage15	K-space door masker verkregen	Referentie afbeelding

Net-stage15 bestaat uit 70 *layers*. Om uit InitNet een netwerk te maken dat ook 70 *layers* heeft, is voor de variabele *stage* een waarde van 17 gekozen. De variabele *InitialLearnRate* is ingesteld op 0.01 in plaats van 0.001. Daar is voor gekozen zodat de *weights* in grote mate aangepast zullen worden. Voor de variabelen *FilterNumber, FilterSize, Padding* en *LinearLabel* zijn de waarden niet veranderd, omdat deze al optimaal waren volgens de literatuur[50].

Uit de *loss*, PSNR en output van deze testen (zie tabel 4) blijkt duidelijk dat het trainen met *undersampled k-spacedata* betere resultaten geeft dan trainen met *fully sampled k-spacedata*. Dit kan verklaard worden doordat het netwerk waarschijnlijk ontworpen is om lege delen in te vullen door middel van *compressive sensing*. *Fully sampled k-space* heeft geen lege plekken om in te vullen, dus zal het netwerk meer moeite hebben met het vormen van een MR-afbeelding. Verder is te zien aan de *loss*, PSNR en output, dat het trainen van InitNet een beter resultaat geeft dan het trainen van net-stage15.

Tabel 4: *Resultaten van het onderzoek vóór het definitieve trainen.*

Netwerk	InitNet	InitNet	net-stage15	net-stage15
Aantal iteraties	6	9	8	5
Input	Fully sampled k-space	K-space door masker verkregen	Fully sampled k-space	K-space door masker verkregen
Loss	7,4162	0,20521	102,2289	0,34991
PSNR	-37,7173	-6,6229	-60,7678	-11,3912
Output				

3.1.4 Definitieve training

Op basis van de resultaten van het onder zoek vóór het trainen, zal het definitieve trainen van het netwerk gebeuren met InitNet als initieel netwerk en zal *undersampled* data gebruikt worden. Er zal met verschillende trainingsdata gewerkt worden. Zo zal er getraind worden met *undersampled* data die verkregen is door het 20% masker. Als dit een goed resultaat geeft zal er getraind worden met de originele *undersampled* data uit de MRI-scanner, deze bevat 8 *spokes* aan informatie. Er zal, net zoals bij het onderzoek voorafgaand aan het definitieve trainen, getraind worden aan de hand van het script *main_netTrain.m*, dat te zien is in bijlage B. De instellingen zijn, op *TrainNumber* na, het zelfde gebleven.

Met de *undersampled k-spacedata*, verkregen met behulp van het 20% masker, zal twee keer getraind worden. De eerste keer wordt getraind met een *TrainNumber* van 360 en de andere keer zal InitNet getraind worden met een*TrainNumber* van 150. Per proefpersoon is bekeken tijdens welke frames de persoon aan het slikken was. Door deze frames te gebruiken als input, zal er relatief veel variatie zijn tussen de frames. De gekozen frames zijn te zien in tabel 5.

Tabel 5: *De gekozen frames voor het trainen met het 20% ADMM-net masker*

	Tijdframes	
Persoon	**Training 360 afbeeldingen**	**Training 260 afbeeldingen**
20190220	1 t/m 60	11 t/m 40
20190213	1 t/m 60	21 t/m 50
20190501	1 t/m 60	11 t/m 40
20181101	11 t/m 70	71 t/m 100
20181025	61 t/m 120	51 t/m 80
20181023	31 t/m 90	–

Ook bij het trainen van Initnet met *undersampled k-space* met 8 *spokes* is *TrainNumber* aangepast. Omdat het netwerk nog niet eerder getraind is met zo weinig informatie in de *k-space* is er gekozen om het *TrainNumber* te verhogen naar 1850. Met een totale beschikking tot 1890 afbeeldingen blijven daardoor nog 40 afbeeldingen over om het netwerk mee te testen.

De resulterende netwerken zullen vergeleken worden met het netwerk *net-stage15.mat*. Met het script *main_ADMM_Net_test.m* zal gekeken worden naar de output die het DL-netwerk heeft gegenereerd, de *loss*, de PSNR en de tijd die het kostte voor het DL-netwerk om de output te genereren. Daarnaast zal

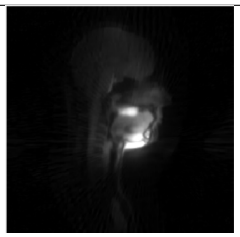
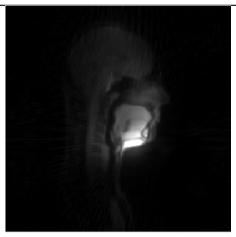
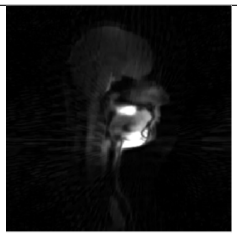
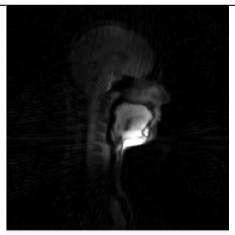
gekeken worden naar de output van het DL-netwerk met zowel een bekende als onbekende *k-space* als input.

3.2 Resultaten

3.2.1 Training met 360 beelden met het 20% masker

Het resultaat van de training met 360 beelden van *undersampled k-spacedata*, verkregen middels het 20% masker, als input, is te zien in tabel 6. De training is na één iteratie opgehouden en heeft het netwerk 'netinitmask20def-001' opgeleverd. Netinitmask20def-001 en net-stage15 zijn getest met als bekende input de *k-space* van frame 10 en als onbekende input de *k-space* van frame 150 uit '20190501.mat'.

Tabel 6: *Resultaten van het trainen met 360 afbeeldingen.*

Netwerk	netinitmask20def-001	netinitmask20def-001	net-stage15	net-stage15
tijdframe uit 20190501.mat	frame 10	frame 150	frame 10	frame 150
Loss	0.2995	0.29627	0.14824	0.14723
PSNR in dB	-10.996	-10.3106	-4.7028	-4.032
Tijd in seconden	1.411734	1.416931	1.435609	1.456482
Output				

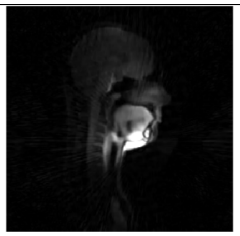
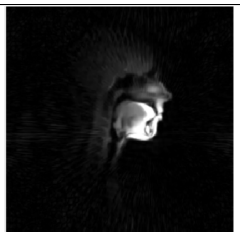
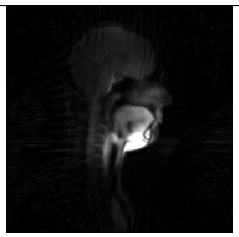
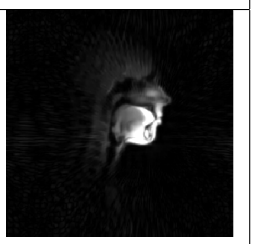
3.2.2 Training met 150 beelden met het 20% masker

Het resultaat van de training met 150 beelden van *undersampled k-spacedata*, verkregen middels het 20% masker, als input, is te zien in tabel 7. Het trainen is na 4 iteraties opgehouden en heeft het netwerk 'netmask30pp-004' opgeleverd. Netinitmask30pp-004 en net-stage15 zijn getest met als bekende input de *k-space* van frame 30 van *'20190501.mat'* en als onbekende input de *k-space* van frame 30 van *'20181025.mat'*.

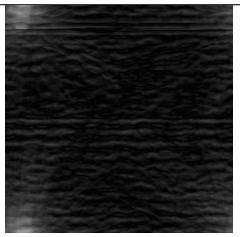
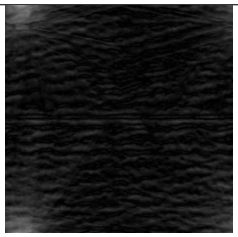
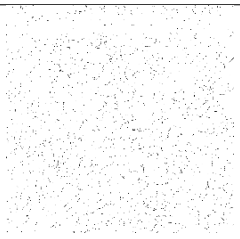
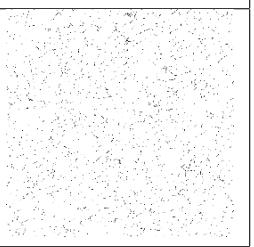
3.2.3 Training met 8 spokes

Het resultaat van de training met 1850 beelden van *undersampled k-spaces* met 8 *spokes* is te zien in figuur 8. Het trainen is na 11 iteraties opgehouden en heeft het netwerk 'net8spokes-011' opgeleverd. Net8spokes-011 en net-stage15 zijn getest met bekende input genaamd *rawdata1800* en een onbekende input genaamd *rawdata1851*. Het resultaat is te zien in figuur 8.

Tabel 7: *Resultaten van het trainen met 150 afbeeldingen.*

Netwerk	netmask30pp-004	netmask30pp-004	net-stage15	net-stage15
referentie output	20190501: frame 30	20181025: frame 30	20190501: frame 30	20181025: frame 30
Loss	0.1201	0.1604	0.1461	0.19089
PSNR	-2.8960	-5.8899	-4.6572	-7.4923
Tijd in seconden	1.537569	1.796873	1.58317	1.502009
Output				

Tabel 8: *Resultaten van het trainen met 1850 afbeeldingen met 8 spokes.*

Netwerk	net8spokes-011	net8spokes-011	net-stage15	net-stage15
referentie output	rawdata1800	rawdata1851	rawdata1800	rawdata1851
Loss	1	1	2.9916	2.3472
PSNR	-20.0754	-19.9856	-28.8728	-26.4752
Output				

3.3 Discussie

3.3.1 Bespreking Resultaten

De resultaten van het netwerk dat met 360 20% beelden is getraind (zie tabel 6) laten een *loss* van 0,3 en 0,3 en een PSNR van -11 en -10 zien. In vergelijking met het niet in dit onderzoek getrainde net-stage15, heeft Netinitmask20def-001 een hogere *loss* en een lagere PSNR. Dit betekent dat net-stage15 beter in staat is MR-beelden te reconstrueren vanuit *undersampled k-space* dan Netinitmask20def-001. Er zit maar een klein verschil tussen de *loss* en PSNR van de bekende en de onbekende afbeeldingen in beide netwerken. Dit kan betekenen dat er geen *overfitting* heeft plaatsgevonden.

De resultaten van 150 20% beelden (zie tabel 7) laten een *loss* van 0,12 en 0,16 en een PSNR van -2,9 en -5,9 zien voor tijdframe 30 uit 20190501.mat en 20181025 respectievelijk. De *loss* van netmask30pp-004 is lager en de PSNR hoger dan die van net-stage15. Dit betekent dat netmask30pp-004 beter dan net-stage15 MR-beelden kan reconstrueren vanuit een *undersampled k-space*. Deze training is dus succesvol geweest. In tegenstelling tot de training met 360 afbeeldingen, verschillen de *loss* en PSNR nu wel tussen de bekende en de onbekende afbeelding bij netmask30pp-004. Hoewel voor deze training van iedere proefpersoon slechts 30 tijdframes zijn gebruikt, waarin bovendien de slikbeweging plaatsvond zo is geprobeerd *overfitting* tegen te gaan, lijkt deze bevinding toch te duiden op *overfitting*. Opvallend is dat hetzelfde fenomeen te zien is in de resultaten van net-stage15, terwijl voor dit netwerk de *undersampled k-space* data

van beide tijdframes nog onbekend is. Deze bevinding lijkt tegen te spreken dat het verschil in PSNR tussen de verschillende tijdframes te wijten is aan *overfitting*.

De resultaten van de training met 1850 beelden van *undersampled k-space* met 8 *spokes* als input, laten een zeer hoge *loss* van 1 en een zeer lage PSNR van -20 zien (zie tabel 8). Omdat de *loss* van net-stage15 nog hoger (3,0 en 2,3) en de PSNR nog lager is (-29 en -26), valt te concluderen dat net8spokes-011 beter in staat is MR-beelden te reconstrueren uit de *undersampled k-space* dan net-stage15. Echter, dysfagiegradatie is niet mogelijk aan de hand van beide outputs. De training met 1850 beelden is dus niet succesvol geweest.

Een DL-netwerk zou pas een toevoeging zijn in de kliniek als het netwerk in staat is de orginele *undersampled k-spacedata* uit de MRI om te zetten naar MR-afbeeldingen. Zoals te zien is in de resultaten van het trainen met 1850 afbeeldingen met 8 *spokes* is dat op dit moment nog zeker niet mogelijk. Deze *undersampled k-space* vult slechts 6,58% van de 256x256 matrix, wat waarschijnlijk te weinig informatie is voor het netwerk om succesvol *compressive sensing* toe te passen.

De netwerken die getraind zijn op 150 en 360 *undersampled k-space* beelden, doen allemaal ongeveer 1,5 seconden over het reconstrueren van één MR-beeld uit *undersampled k-space*. Het reconstrueren van een dMR-video van 315 frames zal daarom in 7,8 minuten gereconstrueerd kunnen worden vanuit *k-space*. Het algoritme van Voskuilen doet over het reconstrueren van een dergelijke video 30 minuten. Deze afname in tijd is van grote klinische betekenis. 30 Minuten wachttijd na het maken van de dMR-beelden is een lange tijd voor zowel de clinicus als de patiënt. Een wachttijd van slechts 7,8 minuten is een goede stap in de richting van implementatie van dMRI voor dysfagiegradatie in de kliniek.

3.3.2 Onverwachte bevindingen

Een onverwachte bevinding tijdens het experimenteren met de training van het netwerk, was dat het netwerk na training met *fully sampled k-space* een grotere *loss* en een lagere PSNR gaf dan het netwerk na training met *undersampled* data. Van tevoren werd gedacht dat het reconstrueren van MR-beelden uit *fully sampled k-space* data makkelijker zou zijn voor het netwerk, aangezien er in dat geval meer informatie beschikbaar is over hoe de output er uit moet komen te zien. Deze onverwachte bevinding zou verklaard kunnen worden door het feit dat het ADMM-net gebaseerd is op *compressive sensing* en dus gemaakt is om lege delen op te vullen. Bij *fully sampled k-space* zijn er geen lege plekken om op te vullen en zal het netwerk dus niet zijn functie kunnen uitvoeren.

3.3.3 Beperkingen

Tijd te kort

De grootste limitatie van dit onderzoek was de tijdsplanning. Het totale onderzoek duurde 11 weken. Door het opzetten van het onderzoek, het uitvoeren van literatuuronderzoek, het testen van het ADMM-net en het schrijven van het verslag, bleven effectief drie weken over om de netwerken te trainen. Aangezien de meeste trainingen acht uur tot vier dagen duurden, is slechts een begin gemaakt met de trainingen. Indien er meer tijd was geweest, hadden verscheidene instellingen en datasets getest en vergeleken kunnen worden.

Trainingsdata

Voor de training heeft het DL-netwerk een afbeelding nodig als referentie voor de output. In dit onderzoek zijn deze referentie-beeldingen gereconstrueerd vanuit het algoritme van Voskuilen. Deze afbeeldingen zijn daarom van mindere kwaliteit dan afbeeldingen die *fully sampled* opgenomen worden door een MR-scanner. Aangezien het algoritme nog in de onderzoeksfase zit, is er weinig bekend over de nauwkeurigheid van het algoritme dat is gebruikt voor deze reconstructie. Het gebruik van deze

gesimuleerde afbeeldingen kan daarom een negatieve invloed hebben op de prestaties van het netwerk.

De afbeeldingen die de DL als output geeft zijn nog niet optimaal. Zelfs het beste netwerk geeft nog een negatieve PSNR van -2.8960 en een *loss* van 0.1201. Dit zou kunnen komen doordat het netwerk is getraind op *fully sampled k-spacedata* die door middel van een pseudoradiaal masker is *undersampled*. Deze *fully sampled k-spacedata* is namelijk de inverse fouriergetransformeerde van een gesimuleerd MR-beeld dat is verkregen met het algoritme van Voskuilen. Het zou dus kunnen dat de *undersampled* data net niet de 'echte' data is, maar een deel van de gesimuleerde data. Dit zou ertoe kunnen leiden dat de output nog veel artefacten bevat.

Ook is er op dit moment nog te weinig data beschikbaar. Het netwerk kon maar getraind worden met data van 6 proefpersonen. Per proefpersoon zijn er 315 beelden beschikbaar. Van deze 315 beelden werd er maar in ongeveer 60 van deze frames een slikbeweging uitgevoerd. In de overige 255 afbeeldingen ligt de proefpersoon haast stil. Aangezien het om gezonde proefpersonen gaat, is de anatomie en fysiologie van de proefpersonen redelijk vergelijkbaar. Het is waarschijnlijk dat deze vergelijkbare afbeeldingen kunnen zorgen voor *overfitting*. Er zal getraind moeten worden met data van meer verschillende proefpersonen en patiënten om zo een netwerk te kunnen creëren dat ook een juiste output kan genereren bij onbekende inputdata.

3.3.4 Aanbevelingen

Op basis van de hiervoor beschreven beperkingen van dit onderzoek, zijn er een aantal aanbevelingen opgesteld voor vervolgonderzoeken.
Om te beginnen zijn de netwerken getraind om elk frame in de *k-space* om te zetten naar het bijbehorende MR-beeld in datzelfde tijdframe. Op deze manier wordt geen gebruik gemaakt van het feit dat de tijdframes elkaar opvolgen in de tijd en kan het netwerk niet leren van de veranderingen in de tijd. Daarom wordt aangeraden om in een vervolgonderzoek netwerken niet alleen te trainen met enkele frames, maar ook met complete dMRI-video's van slikbewegingen, zodat het netwerk vanuit de *k-space* een dMRI-video kan reconstrueren.

Ten tweede zal meer onderzoek gedaan moeten worden naar andere netwerkarchitecturen voor het oplossen van het specifieke regressieprobleem waar dit onderzoek zich op richt. Wegens tijdsgebrek is na literatuuronderzoek gekozen om ADMM-net te trainen. Wellicht dat andere netwerken, zoals U-Net[42], na training beter presteren dan ADMM-net.

Tot slot is gebruik gemaakt van dMRI-video's van gezonde proefpersonen voor de trainingen. Dysfagie kent echter vele vormen en oorzaken. Voordat DL in de kliniek toegepast kan worden voor dysfagiegradatie, zal het netwerk dus eerst getraind moeten worden op afbeeldingen van patiënten met verschillende vormen van dysfagie, zodat het in staat is om bij een nieuwe patiënt de vorm van dysfagie en verschillende kenmerken ervan te herkennen. Deze kenmerken komen aan bod in het hoofdstuk 'Klinische aspecten' .

3.4 Conclusie

Het DL-netwerk ADMM-net kan na training undersampled k-spacedata omzetten naar MR-beelden, echter is de spatiële resolutie van deze MR-beelden nog niet hoog genoeg. Er zal nog meer getraind moeten worden, maar de opzet van het ADMM-net lijkt geschikt om MR-beelden te vormen uit *undersampled k-space*. Ook is dit is enkel nog een proof-of-concept, dus voordat klinische implementatie mogelijk is zal eerst vervolgonderzoek gedaan moeten worden.

4 Klinische aspecten

Voordat dMRI met het bijbehorend DL-netwerk gebruikt kan worden voor het gradueren van dysfagie in de kliniek, is het belangrijk om in beeld te krijgen wat belangrijk is voor deze graduering en of de belangrijke aspecten goed zichtbaar zijn in de nieuwe beelden. Om hier informatie over te verkrijgen is literatuuronderzoek uitgevoerd en zijn interviews[11][12] [15] gehouden met specialisten. In dit hoofdstuk zullen de resultaten van dit literatuuronderzoek en de interviews beschreven worden, als aanvulling op de informatie beschreven in de inleiding. Ten slotte zullen er aanbevelingen gedaan worden voor de gewenste klinische output van een DL-netwerk in de toekomst.

4.1 Klinische parameters uit videofluoroscopie

De slikvideo's worden in het Antonie van Leeuwenhoekziekenhuis bekeken van lippen tot maag.[11] Dit is gebaseerd op het Logemann's model[59] en komt overeen met de informatie gevonden tijdens de literatuurstudie[60][61][13]. Tabel 9 geeft een overzicht van de kenmerken waarop gelet wordt bij het gradueren van dysfagie.

Tabel 9: *Aspecten voor dysfagie gradatie [11]*

Locatie	Aspect
Lippen	Gaat een deel van de bolus weer via de mond naar buiten?
Mondholte	Gaat het kauwen goed en snel genoeg? Is het transport naar de achterkant van de mondholte vertraagd? Hoe is de orale inzet en de slikinzet? Is er sprake van residu?
Orofarynx	Raakt de achterwand van de tong de achterwand van de keel?
Vallecula en luchtpijp	Kantelt de vallecula goed? Indien de vallecula niet goed functioneert, is het dan rigide, verkort of verdikt, of sluit het niet goed de luchtpijp af?
Farynx	Is er sprake van residu? Indien er sprake is van residu, zit dit dan anterior van de vallecula of boven de slokdarmingang en wordt dit dan eventueel veroorzaakt door een slechte larynxheffing[15]? Helpt naslikken om het residu in de slokdarm te laten gaan?
Slokdarm	Is er verminderde opening van de oesofageale sfincter?
Maag	In de passage naar de maag goed?

Na het onderzoeken van deze kenmerken wordt de penetratie- en aspiratiegradatie (PAS) toegepast[11]; een schaal die een waarde hangt aan de mate van penetratie en aspiratie. Deze schaal is gebaseerd op de hoeveelheid bolus die in de luchtweg komt en of deze bolus de stembanden raakt of zelfs passeert.[62] Bij uitgebreidere onderzoeken wordt ook de *oropharyngeal swallow efficiency* (OPSE) van Logemann en het aantal keren slikken genoteerd.[11] De OPSE is een maat voor de efficiëntie van het bolustransport en is te berekenen aan de hand van de duur van het transport en een benadering van het percentage bolus dat achterblijft als residu en aspiratie.[63] Ten slotte kunnen ook de heffing en de kanteling van het strottenhoofd berekend worden door middel van kalibratie met een 10 eurocentmunt, die onder de kin van de patiënt is geplakt en op videofluoroscopiebeelden goed zichtbaar is. Vervolgens wordt een lijn getrokken tussen nekwervel C3 en het tongbeen.[11]

4.2 De output van een DL-netwerk in de kliniek

Door een DL-netwerk op de juiste manier toe te passen in de kliniek, kan het geven van objectieve maten, geautomatiseerd worden, waardoor tijd wordt gewonnen en de werkdruk verlaagd kan worden.

Een voorbeeld van een automatische objectieve beoordeling is wanneer met behulp van DL gemeten kan worden hoe lang de achterwand van de tong in contact staat met de achterwand van de keel. Door het DL-netwerk te leren wat de achterwand van de tong en wat de achterwand van de keel is, kan de duur van deze aanraking automatisch bepaald worden. Als de tong de achterwand van de keel niet aanraakt, kan dit een aanwijzing zijn voor verminderde kracht van de tong. Een andere wenselijke output van een DL-netwerk is het automatisch invullen van bijvoorbeeld de PAS en OPSE-score.[11]

Wanneer het DL-netwerk gebruikt wordt voor classificatie, moet samen met de specialisten bepaald worden wat de gewenste output is. Zo zou het voor de specialisten handig zijn als het DL-netwerk bijvoorbeeld kan aangeven in welk gedeelte van de keel het probleem zich bevindt en ook hoe lang iedere fase van de slikbeweging is. Hierdoor kan een aantal mogelijke oorzaken voor de slikproblemen uitgesloten worden. De logopedist zou hierdoor niet meer naar alle delen van de keel hoeven te kijken. Een andere handige output zouden slikoefeningen kunnen zijn, die voor de patiënt tot verbetering van de slikproblemen kunnen leiden. Als dit goed werkt, kunnen veel onnodige oefeningen overgeslagen worden. Echter, deze output is erg subjectief en het effect van dergelijke oefeningen is erg patiëntafhankelijk, waardoor het resultaat van te voren moeilijk in te schatten is.[15]

Meer tijdwinst en verlaging van de werkdruk zou bereikt kunnen worden door het automatisch gradueren of classificeren met behulp van DL. Voor automatische gradatie zou het netwerk getraind kunnen worden met beelden die gelabeld zijn door logopedisten (*supervised learning*). Dit betekent dat de output subjectief blijft, maar wel dezelfde output geeft als de logopedist - de huidige gouden standaard - en dat het minder arbeidsintensief zal zijn. Een andere mogelijkheid voor het trainen is *semi-supervised learning*. Een deel van de data zal dan beoordeeld worden door een logopedist als referentie en het andere deel zal beoordeeld worden door het netwerk zelf. Het netwerk zou dan kenmerken uit de beelden kunnen herkennen die de door de logopedist over het hoofd zijn gezien.

Een belangrijk voordeel van DL is dat het de voorgeschiedenis van een patiënt en van vorige patiënten mee kan nemen in het bepalen van zijn output.[11] Het DL-netwerk zou op deze manier een betere voorspelling kunnen maken van de uitkomst van een specifieke behandeling.

5 Algehele discussie

Dit onderzoek had als doel de volgende vraag te beantwoorden: 'Hoe kunnen dynamische MR-beelden van slikbewegingen met *deep learning* gereconstrueerd worden vanuit *undersampled k-space* data en welke klinische aspecten zijn relevant om in de toekomst dysfagie te gradueren met behulp van dynamische MRI?'

5.1 Deep learning en dynamische MRI

Het DL-netwerk ADMM-net kan na training *undersampled k-spacedata* omzetten in MR-beelden, maar de spatiële resolutie van deze afbeeldingen is nog niet hoog genoeg om dit netwerk in de kliniek te gebruiekn. Wel is het getrainde netwerk in staat circa 5 keer zo snel MR-beelden te reconstrueren dan het algoritme van Voskuilen. Dit is enkel nog een proof-of-concept, dus voordat klinische implementatie mogelijk is zal eerst vervolgonderzoek gedaan moeten worden. Dit vervolgonderzoek zal zich met name moeten richten op data die gebruikt wordt voor de training van het netwerk; dat wil zeggen input in de vorm van *spokes* en *trajectories* en daadwerkelijk *fully sampled* outputdata. Deze data zal afkomstig moeten zijn van een grote groep patiënten die lijden aan dysfagie en deze groep zal gevarieerd moeten zijn.

5.2 Toekomstperspectief

Het netwerk dat getraind is voor dit onderzoek maakt het mogelijk om sneller MR-beelden te maken dan nu wordt gedaan in de kliniek. Hiermee wordt tijd gewonnen, wordt de werkdruk van specialisten verlaagd en wordt het mogelijk om videofluoroscopie, dat vele nadelen heeft, te vervangen. Zo is het met dMRI mogelijk om kwantitatieve informatie te krijgen, zoals over penetratie en aspiratie. Een mogelijke vervolgstap zou zijn om een classificatie aan te brengen in het netwerk, waardoor het in staat is een patiënt te classificeren naar een bepaalde oorzaak of aspect van dysfagie. Ook dit kan mogelijk de werkdruk bij specialisten verlagen.

DMRI brengt vele voordelen met zich mee voor dysfagiegradatie, zoals visualisatie van weke delen, 3D-beelden en de mogelijkheid om kwantitatieve metingen te verrichten. Een beperking van het onderzoek is echter dat het netwerk is getraind op 2D-beelden, waardoor veel cruciale informatie uit de laterale dimensie niet beschikbaar is. Dit kan een beperking zijn wanneer belangrijke anatomische en pathologische structuren, die de oorzaak kunnen zijn van de dysfagie, in dit vlak niet zichtbaar zijn. Ook is het bij 3D-beelden mogelijk om absolute metingen te verrichten, wat van pas kan komen bij het gradueren van dysfagie. Er wordt daarom aanbevolen om in vervolgonderzoek te bekijken of het mogelijk is een netwerk met 3D MR-beelden te trainen. Hoewel dMRI vele voordelen heeft ten opzichte van videofluoroscopie, heeft het ook enkele nadelen. In de meeste MRI-scanners in de kliniek moeten patiënten liggen in plaats van zitten. Liggend slikken kan gevaarlijk zijn voor mensen die slecht kunnen slikken. Daarnaast is de vergelijking tussen videofluoroscopie en dMRI niet zuiver vanwege zwaartekracht. Wellicht dat een verticaal gepositioneerde MRI-scanner uitkomst kan bieden, maar op dit moment is de sterkte van het magneetveld van deze MRI-scanners te laag.[11][12] Tot slot kan een MRI-onderzoek, ondanks de versnelling van het proces met DL, langer duren dan een videofluoroscpie-onderzoek, omdat bij MRI de patiënt zorgvuldiger voorbereid moet worden op het onderzoek. Een MRI-onderzoek kan dus alsnog belastend zijn voor de patiënt en arbeidsintensiever zijn voor de specialist.

De toepassing van dMRI in combinatie met DL hoeft zeker niet beperkt te zijn tot de hoofd-halschirurgie en kan gebruikt worden binnen andere specialismen. Voorbeelden hiervan zijn ortopedie, waarbij dynamische beelden gemaakt kunnen worden van het bewegingsapparaat, en de thoraxchirurgie, waarbij deze toepassing geschikt is in verband met de bewegende borstkas tijdens de ademhaling. Binnen deze specialismen zal de mogelijkheid van het maken van dMR-beelden met weinig artefacten grote voordelen hebben bij diagnostiek.

5.3 Conclusie

Het ADMM-net in combinatie met dMRI geeft veelbelovende resultaten en kan in de toekomst, indien verder getraind op gevarieerde patiëntendata verkregen, een goede methode zijn voor dysfagiegradatie.

Referenties

[1] Patrick Zuercher, Céline S. Moret, Rainer Dziewas, and Joerg C. Schefold. Dysphagia in the intensive care unit: epidemiology, mechanisms, and clinical management. *Critical Care*, 23(1):103, dec 2019.

[2] Federatie Medisch Specialisten. *Orofaryngeale dysfagie - Richtlijn 2017*. Federatie Medische Specialisten, 2017.

[3] A Olthoff, P.-O. Carstens, S Zhang, E Von Fintel, T Friede, J Lotz, J Frahm, and J Schmidt. Evaluation of dysphagia by novel real-time MRI. *Neurology*, 87(20):2132–2138, 2016.

[4] Lindsay East, Kimberly Nettles, Ann Vansant, and Stephanie K. Daniels. Evaluation of Oropharyngeal Dysphagia With the Videofluoroscopic Swallowing Study. *Journal of Radiology Nursing*, 33(1):9–13, mar 2014.

[5] Giuseppe Cicala, Maria Antonietta Barbieri, Edoardo Spina, and Jose de Leon. A comprehensive review of swallowing difficulties and dysphagia associated with antipsychotics in adults. *Expert Review of Clinical Pharmacology*, 12(3):219–234, mar 2019.

[6] Jenny Walton and Priyamal Silva. Physiology of swallowing. *Surgery (Oxford)*, 36(10):529–534, oct 2018.

[7] Sharon Theimer. Los síntomas del cáncer de garganta dependen de las estructuras afectadas, 2017.

[8] P. Kumar and M. Clark. The pharynx and oesophagus. In *Kumar & Clark's Clinical Medicine*, chapter 6, pages 236–246. Saunders Elsevier, 8 edition, 2012.

[9] Emanuel Rubin and Howard M. Reisner. The Gastrointestinal Tract. In *Essentials of Rubin's pathology*, chapter 13, pages 351–357. Wolters Kluwer Health | Lippincott Williams &Wilkins, Baltimore, MD, 6 edition, 2014.

[10] M. J. T. Fitzgerald, G. Gruener, and E. Mtui. *Clinical Neuroanatomy and Neuroscience*. Saunders Elsevier, 6 edition, 2012.

[11] L. van der Molen. Persoonlijk interview, 2019.

[12] F. J. M. Balm. Persoonlijk interview, 2019.

[13] Olle Ekberg. *Dysphagia: Diagnosis and Treatment*. Springer Science & Business, 2012.

[14] John E. Morley. Dysphagia and Aspiration. *Journal of the American Medical Directors Association*, 16(8):631–634, aug 2015.

[15] M. van Alphen. Persoonlijk interview, 2019.

[16] Paul D. Neubauer, Alfred W. Rademaker, and Steven B. Leder. The Yale Pharyngeal Residue Severity Rating Scale: An Anatomically Defined and Image-Based Tool. *Dysphagia*, 30(5):521–528, 2015.

[17] Daniels S Easterling C. Continued Relevance of Videofluoroscopy in the Evaluation of Oropharyngeal Dysphagia. *Current Radiology Reports*, 5(2):6, 2017.

[18] Shenasa M. MRI Anatomy for Cardiac Mapping and Ablation. In *Cardiac mapping*, chapter 9, page 524. Wiley-Blackwell, 3 edition, 2009.

[19] MRI-questions.com. K-space - Questions and Answers âĂŃin MRI.

[20] R. Mezrich. A perspective on k-space. *Radiology*, 195:297–315, 1995.

[21] Wikipedia. Trajectory - Wikipedia, 2019.

[22] L. Feng, R. Grimm, K. T. Block, and Et al. Golden-angle radial sparse parallel MRI: combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI. *Magn Reson Med*, 72:707–717, 2014.

[23] L. Greengard and J. Y. Lee. Accelerating the Nonuniform Fast Fourier Transform. *SIAM Review*, 46(3):444–454, 2004.

[24] Wikipedia. Compressed sensing, 2019.

[25] Allison Gray. Practical Deep Learning Examples. Technical report, MathWorks, 2018.

[26] Charu C. Aggarwal. *Neural networks and deep learning : a textbook*. Springer, 2018.

[27] Mathworks. Introducing Deep Learning with MATLAB. *Introducing Deep Learning with MATLAB*, page 15, 2018.

[28] E. Alpaydin. *Introduction to machine learning*. MIT Press, 2014.

[29] Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul A. S. Awwal, Vijayan K. Asari, Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3):292, mar 2019.

[30] MathWorks. What Is Deep Learning? | 3 things you need to know.

[31] Justin Ker, Lipo Wang, Jai Rao, and Tchoyoson Lim. Deep Learning Applications in Medical Image Analysis. *IEEE Access*, 6:9375–9389, 2018.

[32] MathWorks. Transfer Learning Using AlexNet, 2019.

[33] Markt Hodnett and Joshua F. Wiley. *R Deep Learning Essentials: A step-by-step guide to building deep learning models using TensorFlow, Keras, and MXNet*. Packt Publishing Ltd, 2018.

[34] Ethem Alpaydin. *Introduction to Machine Learning Ethem Alpaydin.* Cambridge, Massachusetts: The MIT Press, 2014.

[35] Sergey Demyanov, Rajib Chakravorty, Zongyuan Ge, SeyedBehzad Bozorgtabar, Michelle Pablo, Adrian Bowling, and Rahil Garnavi. Tree-loss function for training neural networks on weakly-labelled datasets. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 287–291. IEEE, apr 2017.

[36] M. Bilgehan and P. Turgut. Artificial Neural Network Approach to Predict Compressive Strength of Concrete through Ultrasonic Pulse Velocity. *Research in Nondestructive Evaluation*, 21(1):1–17, jan 2010.

[37] Frauke Wouda, Henk Hutink, and Jacqueline Nell. Artificial Intelligence in de zorg begrippen, praktijkvoorbeelden en vraagstukken. Technical report, Nictiz, 2019.

[38] Sameer Khan and Suet-Peng Yong. A deep learning architecture for classifying medical images of anatomy object. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1661–1668. IEEE, dec 2017.

[39] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep Learning in Medical Image Analysis. *Annual review of biomedical engineering*, 19:221, 2017.

[40] Yuhai Yu, Hongfei Lin, Jiana Meng, Xiaocong Wei, Hai Guo, and Zhehuan Zhao. Deep Transfer Learning for Modality Classification of Medical Images. *Information*, 8(3):91, 2017.

[41] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *âĂİ IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

[42] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo. Deep learning for undersampled MRI reconstruction. *Physics in Medicine & Biology*, 63(13), 2018.

[43] Florian Dubost, Gerda Bortsova, Hieab Adams, Arfan Ikram, Wiro Niessen, Meike Vernooij, and Marleen De Bruijne. GP-Unet: Lesion Detection from Weak Labels with a 3D Regression Network. *CoRR*, may 2017.

[44] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. ADMM-Net: A Deep Learning Approach for Compressive Sensing MRI. *CoRR*, may 2017.

[45] Sampurna Biswas, Hemant K. Aggarwal, and Mathews Jacob. Dynamic MRI using modelâĂŘbased deep learning and SToRM priors: MoDLâĂŘSToRM. *Magnetic Resonance in Medicine*, 82(1):485–494, jul 2019.

[46] Yudai Suzuki, Keigo Kawaji, Amit R. Patel, Satoshi Tamura, and Satoru Hayamizu. Toward effective noise reduction for sub-Nyquist high-frame-rate MRI techniques with deep learning. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1136–1139. IEEE, dec 2017.

[47] J Schlemper, O Oktay, W Bai, D C Castro, J Duan, C Qin, J V Hajnal, and D Rueckert. Cardiac MR segmentation from undersampled k-space using deep latent representation learning, 2018.

[48] From Wikipedia. Augmented Lagrangian method, 2015.

[49] yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for Compressive Sensing MRI. In D D Lee, M Sugiyama, U V Luxburg, I Guyon, and R Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 10–18. Curran Associates, Inc., 2016.

[50] Z. yang, yan, Sun, J., Li, H., & Xu. Github, 2016.

[51] Tamir J.I., Ong F., Cheng J.Y., Uecker M., and Lustig M. Generalized Magnetic Resonance Image Reconstruction using The Berkeley Advanced Reconstruction Toolbox. In *ISMRM Workshop on Data Sampling and Image Reconstruction*, Sedona, 2016.

[52] Uecker M., Ong F., Tamir J.I., Bahri D., Virtue P., Cheng J.Y., Zhang T., and Lustig M. Berkeley Advanced Reconstruction Toolbox. *Proc Intl Soc Mag Reson Med*, 23:2486, 2015.

[53] Uecker M., Virtue P., Ong F., Murphy M.J., Alley M.T., Vasanawala S.S., and Lustig M. Software Toolbox and Programming Library for Compressed Sensing and Parallel Imaging. In *ISMRM Workshop on Data Sampling and Image Reconstruction*. Sedona, 2013.

[54] Russel Reed and Robert J Marks. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks - Russell Reed, Robert J MarksII - Google Boeken*. Cambridge, Mass. : MIT Press, 1999., 1999.

[55] A. Gibson and J. Patterson. *Deep Learning: A Practitioner's Approach*. O'Reilly Media, 2017.

[56] MathWorks. Single Image Super-Resolution Using Deep Learning - MATLAB & Simulink Example, 2019.

[57] Wang Z, Bovik A, Sheikh H, and Simoncelli E. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[58] Shehrzad Qureshi. *Embedded image processing on the TMS320C6000 DSP : examples in code composer studio and MATLAB*. Springer Science+Business Media, 2005.

[59] Jeri A Logemann. *Evaluation and treatment of swallowing disorders*. Pro-Ed, Austin, TX, 2 edition, 1998.

[60] Michael E. Groher and Michael A. Crary. Imaging Swallowing Examinations: Videofluoroscopy and Endoscopy. In *Dysphagia : clinical management in adults and children*, chapter 18, page 386. Elsevier, 2 edition, 2016.

[61] Julie. Cichero and B. E. Murdoch. *Dysphagia : foundation, theory and practice*. John Wiley & Sons, 2006.

[62] Chia-Wei Kuo, Clint Tanner Allen, Chu-Chun Huang, and Chia-Jung Lee. Murray secretion scale and fiberoptic endoscopic evaluation of swallowing in predicting aspiration in dysphagic patients. *European Archives of Oto-Rhino-Laryngology*, 274(6):2513–2519, 2017.

[63] A. W. Rademaker, B. R. Pauloski, J. A. Logemann, and T. K. Shanahan. Oropharyngeal Swallow Efficiency as a Representative Measure of Swallowing Function. *Journal of Speech and Hearing Research*, 37:314–325, 1997.

Bijlage A: zelf geschreven Matlab scripts

Train data van fully sampled k-space

Dit Matlab script is gebruikt om *fully sampled k-space* trainingsdata te genereren.

```
clearvars
close all
load('20190220.mat')

for i = 1:1:315 %dus van 1 tot ND (315 bij 315 afbeeldingen) met stapjes van 1
img=squeeze(img(:,:,i));
img=imresize(img, [256 256]);
img=im2double(img); %Afbeelding transformeren naar goede formaat
kspace_full= fft2(img);
data.train = kspace_full;
data.label = img;
save(strcat('./data_luuk/data_proef/dataproef', saveName(i, 2), '.mat'), 'data'); % Gegenereerde afbeeldingen opslaan
end
```

Train data van undersampled k-space met 20% masker

Dit script is gebruikt om *undersampled k-space* data te genereren.

```
clearvars
close all
load('20190220.mat')

for i = 1:1:315
img=squeeze(img(:,:,i));
img=imresize(img, [256 256]);
img=im2double(img); %Afbeelding transformeren naar goede formaat

kspace_und = (double(kspace_full)) .* (ifftshift(mask)); %Het toepassen van het masker

data.train = kspace_und;
data.label = img;
save(strcat('./data_luuk/data_proef/dataproef', saveName(i, 2), '.mat'), 'data');
end
```

Train data van spokes and trajectories

Met dit script kan aan de hand van de *spokes* en *trajectories undersampled k-space* data gegenereerd worden.

```
clearvars
close all
load('20190220.mat')

%% Loop om trainingsdata aan te passen en op te slaan.
```

```
for i = 1:1:315
afb = squeeze(img(:,:,i)); %original image
afb_resize = imresize(afb, [256 256]);
afb_double = im2double(afb_resize);

reco_test = bart('nufft -a -d 256:256:1 -t',(traj(:,:,:,i)),(spokes(:,:,:,i)
    )); %Grid de spokes: 'reco_test' wordt een matrix van 3x256x8x1.

kspace = bart('fft -u 7', reco_test); %Zet de grid met trajectories en
    spokes om naar het frequentie domein.
kspace=im2double(kspace);
data.train = kspace; %ruwe kspace met 8 spokes
data.label = afb_double; %originele afbeelding
save(strcat('./rawdata_map/rawdata', saveName(i, 2), '.mat'), 'data');
end
```

Train data van spokes en trajectories middels masker

Met dit script wordt aan de hand van de *spokes* en *trajectories* een masker gemaakt. Met dit masker wordt vervolgens de *undersampled k-space* data gegenereerd.

```
threshold = 4;

for i = 1:1:315
afb = squeeze(img(:,:,i)); %Originele afbeelding
afb_resize = imresize(afb, [256 256]);
afb_double = im2double (afb_resize);

reco_test = bart('nufft -a -d 256:256:1 -t',(traj(:,:,:,i)),(spokes(:,:,:,i)
    ));%Grid de spokes: 'reco_test' wordt een matrix van 3x256x8x1, waarbij 3
     de dimensies inhoudt, 256 de grote van de matrix, 8 de aantal spokes en
    1 het aantal tijdframe

kspace = bart('fft -u 7', reco_test); %Deze zet de grid met trajectories en
    spokes om naar het frequentiedomein.
kspaceabs = abs(kspace);
masker = kspaceabs > threshold;
kspace = (double(kspace)) .* (masker);
data.train = kspace; %raw ksp met alleen 8 spokes
data.label = afb_double; %originele afbeelding
save(strcat('./rawdata_map/rawdata', saveName(i, 2), '.mat'), 'data');
end
```

Bijlage B: aangepaste ADMM-net Matlab scripts

De scripts van het ADMM-net zijn verkregen van `http://github.com/yangyan92/Deep-ADMM-Net`. Hieronder staan de scripts die voor dit onderzoek zijn aangepast.

Aangepast script Main_ADMM_Net_test.m

```
%% This is a cpu test code demo for ADMM_Net_v1 reconstruction.
%% Output: the average NMSE and PSNR over the test images.

clc;
clear all;
addpath('./layersfunction/');
addpath('./util');

%% Load trained network
%load('./Train_output/net/netInitmask20def-001.mat'); %zie workspace: net.
load('./net/network_20/net-stage15.mat')
% klikt zie je de layers. The models are trained from 100 rea MR training
% images with 20% sampling rate
%load('./Train_output/net/netINITstage5mask-014.mat');
%% Load data
%load('./data_proef_mask_test/maskertest01.mat');
load('./mask/mask_20.mat') %the mask is a pseudo radial sampling pattern,
% with 20% sampling rate

 %% Undersampling in the k-space
 load('20190501.mat')
 img_n = squeeze (abs(img(:,:,10)));
 img_resize = imresize(img_n, [256 256]); %juiste grootte
 img_double = im2double (img_resize);

 kspace_full = fft2(img_double); %originele afbeelding fouriertransformeren

% kspace_full = fft2(im_ori);
 y = (double(kspace_full)) .* (ifftshift(mask));
% hoeft niet nog eens double, want we hebben van de img al double gemaakt en daar
% nemen we vervolgens de fft van, dus die is ook al double.
% Waarom ifftshift? Waarom ifft en waarom shiften?

data.train = y;
data.label = img_double;

%% reconstrction by ADMM-Net
tic
[re_LOss, rec_image] = loss_with_gradient_single_before(data, net);
% dit trekt je data door je netwerk heen denken we...
% je rec_image is je deep learning output
% je data.label is je originele

```

```
re_PSnr = psnr(abs(rec_image) , abs(data.label)); %calculate the peak
%signal to noise ratio for rec_image, with data.label als referentie.
%rec_image and data.label must be of the same size and class

re_LOss
toc
Zero_filling_rec = ifft2(data.train); %undersampled afbeelding, waarbij je alles
% opvult met 0
%%
figure; sgtitle(['net-stage15, 20190501-10, reLOss = ' num2str(re_LOss) ', rePSnr = ' num2str(re_PSnr)])
subplot(1,4,1); imshow ((abs(data.train)), [0 10]); title('k-space');
subplot(1,4,2); imshow ((abs(data.label)), [0 80]); title ('Originele afbeelding');
subplot(1,4,3); imshow ((abs(Zero_filling_rec)), [0 80]); title('Zero-filling');
subplot(1,4,4); imshow ((abs(rec_image)), [0 80]); title('ADMM-Net result');

% imwrite(abs(rec_image),'rec_image.png')
```

Aangepast script Main_netTrain.m

```
%% This is a traning code for ADMM-Net by L-BFGS optimizing.
%% Copyright (c) 2017 Yan Yang
%% All rights reserved.

clear all ;
clc;
addpath('./Train_LBFGS/')
addpath('./Train_LBFGS/lb')
addpath('./Train_LBFGS/Matlab')
addpath('./layersfunction/')
addpath('./util')
addpath('./data_proef_mask/')
tic

%% Network initialization
% net = InitNet ( );
load('./net/network_20/net-stage15.mat')
% Te vinden onder 'Train_LBFGS'

%% Initial loss
wei0 = netTOwei(net); % transforms the network to weight vector
l0 = loss_with_gradient_total(wei0)

%% L-BFGS optimiztion
fun = @loss_with_gradient_total;
% Parameters in the L-BFGS algorithm
low = -inf*ones(length(wei0),1);
upp = inf*ones(length(wei0),1);
opts.x0 = double(gather(wei0));
```

```
opts.m = 5;
opts.maxIts = 7200;
opts.maxTotalIts = 7200;
opts.printEvery = 1;
opts.InitialLearnRate = 0.001;

[wei1, l1, info] = lbfgsb(fun, low, upp, opts);
wei1=single(wei1);
net1 = weiTOnet(wei1);
fprintf('Before training, error is %f; after training, error is %f.\n', l0,
    l1);
toc
```

Aangepast script getMData.m

```
function data = getMData (n)
config;
size = nnconfig.ImageSize;
ND = nnconfig.DataNmber;
data.train = single(zeros(size));
data.label = single (zeros(size));
dir = './Datamask30pp/20190220mask-';
load(strcat(dir , saveName(n,floor(log10(ND)))));
```

Aangepast script config.m

```
%% network setting
global nnconfig;
nnconfig.FilterNumber =8;
nnconfig.FilterSize = 3;
nnconfig.Stage = 17;
nnconfig.Padding = 1;
nnconfig.LinearLabel = double(-1:0.02:1);

%% training and testing setting
nnconfig.EnableGPU = 1;    %first run on GPU always takes longer than
    subsequent runs
nnconfig.WeightDecay = 0;

%% data
nnconfig.TrainNumber = 150; %komt voor in Loss_with_gradient_total
nnconfig.ValDataNumber = 20;
nnconfig.TesDataNumber = 10;
nnconfig.ImageSize = [256,256];
nnconfig.DataNmber = 150; %het aantal afbeeldingen gemaakt met gen_traindata
    .m
```

Aangepast script lbfgsb.m

```
function [x,f,info] = lbfgsb( fcn, l, u, opts )
% x = lbfgsb( fcn, l, u )
%   uses the lbfgsb v.3.0 library (fortran files must be installed;
%       see compile_mex.m ) which is the L-BFGS-B algorithm.
```

```
%   The algorithm is similar to the L-BFGS quasi-Newton algorithm,
%   but also handles bound constraints via an active-set type iteration.
%   This version is based on the modified C code L-BFGS-B-C, and so has
%   a slightly different calling syntax than previous versions.
%
%  The minimization problem that it solves is:
%       min_x  f(x)     subject to   l <= x <= u
%
% 'fcn' is a function handle that accepts an input, 'x',
%   and returns two outputs, 'f' (function value), and 'g' (function gradient).
%
% 'l' and 'u' are column-vectors of constraints. Set their values to Inf
%   if you want to ignore them. (You can set some values to Inf, but keep
%   others enforced).
%
% The full format of the function is:
% [x,f,info] = lbfgsb( fcn, l, u, opts )
%   where the output 'f' has the value of the function f at the final iterate
%   and 'info' is a structure with useful information
%       (self-explanatory, except for info.err. The first column of info.err
%        is the history of the function values f, and the second column
%        is the history of norm( gradient, Inf ).  )
%
%   The 'opts' structure allows you to pass further options.
%   Possible field name values:
%
%       opts.x0     The starting value (default: all zeros)
%       opts.m      Number of limited-memory vectors to use in the algorithm
%                       Try 3 <= m <= 20. (default: 5 )
%       opts.factr  Tolerance setting (see this source code for more info)
%                       (default: 1e7 ). This is later multiplied by machine epsilon
%       opts.pgtol  Another tolerance setting, relating to norm(gradient,Inf)
%                       (default: 1e-5)
%       opts.maxIts         How many iterations to allow (default: 100)
%       opts.maxTotalIts    How many iterations to allow, including linesearch iterations
%                       (default: 5000)
%       opts.printEvery     How often to display information (default: 1)
%       opts.errFcn         A function handle (or cell array of several function handles)
%                       that computes whatever you want. The output will be printed
%                       to the screen every 'printEvery' iterations. (default: [] )
%                       Results saved in columns 3 and higher of info.err variable
%
```

```
% Stephen Becker, srbecker@alumni.caltech.edu
% Feb 14, 2012
% Updated Feb 21 2015, Stephen Becker, stephen.becker@colorado.edu

narginchk(3, 4)
if nargin < 4, opts = struct([]); end

% Matlab doesn't let you use the .name convention with structures
%   if they are empty, so in that case, make the structure non-empty:
if isempty(opts), opts=struct('a',1) ; end

function out = setOpts( field, default, mn, mx )
    if ~isfield( opts, field )
        opts.(field)    = default;
    end
    out = opts.(field);
    if nargin >= 3 && ~isempty(mn) && any(out < mn), error('Value is too small'); end
    if nargin >= 4 && ~isempty(mx) && any(out > mx), error('Value is too large'); end
    opts    = rmfield( opts, field ); % so we can do a check later
end

% [f,g] = callF( x );
if iscell(fcn)
    % the user has given us separate functions to compute
    %   f (function) and g (gradient)
    callF   = @(x) fminunc_wrapper(x,fcn{1},fcn{2} );
else
    callF   = fcn;
end

n   = length(l);
if length(u) ~= length(l), error('l and u must be same length'); end
x0  = setOpts( 'x0', zeros(n,1) );
x   = x0 + 0; % important: we want Matlab to make a copy of this.
              %  just in case 'x' will be modified in-place
              % (Feb 2015 version of code, it should not be modified,
              %  but just-in-case, may as well leave this )

if size(x0,2) ~= 1, error('x0 must be a column vector'); end
if size(l,2) ~= 1, error('l must be a column vector'); end
if size(u,2) ~= 1, error('u must be a column vector'); end
if size(x,1) ~= n, error('x0 and l have mismatchig sizes'); end
if size(u,1) ~= n, error('u and l have mismatchig sizes'); end

% Number of L-BFGS memory vectors
% From the fortran driver file:
% "Values of m < 3  are not recommended, and
%  large values of m can result in excessive computing time.
%  The range  3 <= m <= 20 is recommended.  "
```

```
m    = setOpts( 'm', 5, 0 );

% 'nbd' is 0 if no bounds, 1 if lower bound only,
%       2 if both upper and lower bounds, and 3 if upper bound only.
% This .m file assumes l=-Inf and u=+Inf imply that there are no constraints
    .
% So, convert this to the fortran convention:
nbd     = isfinite(l) + isfinite(u) + 2*isinf(l).*isfinite(u);
if ispc
    nbd = int32(nbd);
else
    nbd = int64(nbd);
end

% Some scalar settings, "factr" and "pgtol"
% Their descriptions, from the fortran file:

%     factr is a DOUBLE PRECISION variable that must be set by the user.
%       It is a tolerance in the termination test for the algorithm.
%       The iteration will stop when
%
%        (f^k - f^{k+1})/max{|f^k|,|f^{k+1}|,1} <= factr*epsmch
%
%       where epsmch is the machine precision which is automatically
%       generated by the code. Typical values for factr on a computer
%       with 15 digits of accuracy in double precision are:
%       factr=1.d+12 for low accuracy;
%             1.d+7  for moderate accuracy;
%             1.d+1  for extremely high accuracy.
%       The user can suppress this termination test by setting factr=0.
factr   = setOpts( 'factr', 1e7, 0 );

%     pgtol is a double precision variable.
%       On entry pgtol >= 0 is specified by the user.  The iteration
%         will stop when
%
%                 max{|proj g_i | i = 1, ..., n} <= pgtol
%
%         where pg_i is the ith component of the projected gradient.
%       The user can suppress this termination test by setting pgtol=0.
pgtol   = setOpts( 'pgtol', 1e-5, 0 ); % may crash if < 0

% Maximum number of outer iterations
maxIts  = setOpts( 'maxIts', 100, 1 );

% Maximum number of total iterations
%   (this includes the line search steps )
maxTotalIts     = setOpts( 'maxTotalIts', 5e3 );

% Print out information this often (and set to Inf to suppress)
printEvery  = setOpts( 'printEvery', 1 );
```

```
errFcn       = setOpts( 'errFcn', [] );

iprint  = setOpts('verbose',1);
% <0 for no output, 0 for some, 1 for more, 99 for more, 100 for more
% I recommend you set this -1 and use the Matlab print features
% (e.g., set printEvery )

fcn_wrapper(); % initialized persistent variables
callF_wrapped = @(x,varargin) fcn_wrapper( callF, errFcn, maxIts, ...
    printEvery, x, varargin{:} );
% callF_wrapped = @(x,varargin)callF(x); % also valid, but simpler

% Call the mex file
[f,x,taskInteger,outer_count, k] = lbfgsb_wrapper( m, x, l, u, nbd, ...
    callF_wrapped, factr, pgtol, ...
    iprint, maxIts, maxTotalIts);

info.iterations      = outer_count;
info.totalIterations = k;
info.lbfgs_message1  = findTaskString( taskInteger );
errHist = fcn_wrapper();
info.err = errHist;
end % end of main function

function [f,g] = fcn_wrapper( callF, errFcn, maxIts, printEvery, x, varargin
    )
persistent k history
if isempty(k), k = 1; end
if nargin==0
    % reset persistent variables and return information
    if ~isempty(history) && ~isempty(k)
        printFcn(k,history);
        f = history(1:k,:);
    end
    history = [];
    k = [];
    return;
end
if isempty( history )
    width       = 0;
    if iscell( errFcn ), width = length(errFcn);
    elseif ~isempty(errFcn), width = 1; end
    width       = width + 2; % include fcn and norm(grad) as well
    history     = zeros( maxIts, width );
end

% Find function value and gradient:
[f,g] = callF(x);

if nargin > 5
```

```
    outerIter = varargin{1}+1;

    history(outerIter,1)     = f;
    history(outerIter,2)     = norm(g,Inf); % g is not projected
    if isa( errFcn, 'function_handle' )
        history(outerIter,3) = errFcn(x);
    elseif iscell( errFcn )
        for j = 1:length(errFcn)
            history(outer_count,j+2) = errFcn{j}(x);
        end
    end

    if outerIter > k
        % Display info from *previous* input
        % Since this may be called several times before outerIter
        % is actually updated
%         fprintf('At iterate %5d, f(x)= %.2e, ||grad||_infty = %.2e [MATLAB]\n',...
%             k,history(k,1),history(k,2) );
        if ~isinf(printEvery) && ~mod(k,printEvery)
            printFcn(k,history);
            net = weiTOnet(x);
            err = history(k, 1);
            save(strcat('./Train_output/net/netwerkengoed21-6/netmask30pp-',saveName(k, 3), '.mat'), 'net');
            save(strcat('./Train_output/error/errorgoed21-6/errormask30pp-',saveName(k, 3), '.mat'), 'err');
        end
        k = outerIter;
    end

end

end

function printFcn(k,history)
fprintf('Iter %5d, loss = %f, ||grad||_infty = %f', ...
    k, history(k,1),  history(k,2) );
for col = 3:size(history,2)
    fprintf(', %.2e', history(k,col) );
end
fprintf('\n');
end

function [f,g] = fminunc_wrapper(x,F,G)
% [f,g] = fminunc_wrapper( x, F, G )
%   for use with Matlab's "fminunc"
f = F(x);
if nargin > 2 && nargout > 1
    g = G(x);
end
```

```
end

function str = findTaskString( taskInteger )
% See the #define statements in lbfgsb.h
switch taskInteger
case 209
    str = 'ERROR: N .LE. 0';
case 210
    str = 'ERROR: M .LE. 0';
case 211
    str = 'ERROR: FACTR .LT. 0';
case 3
        str = 'ABNORMAL_TERMINATION_IN_LNSRCH.';
case 4
        str = 'RESTART_FROM_LNSRCH.';
case 21
        str = 'CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL.';
case 22
        str = 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH.';
case 31
        str = 'STOP: CPU EXCEEDING THE TIME LIMIT.';
case 32
        str = 'STOP: TOTAL NO. of f AND g EVALUATIONS EXCEEDS LIM.';
case 33
        str = 'STOP: THE PROJECTED GRADIENT IS SUFFICIENTLY SMALL.';
case 101
        str = 'WARNING: ROUNDING ERRORS PREVENT PROGRESS';
case 102
        str = 'WARNING: XTOL TEST SATISIED';
case 103
        str = 'WARNING: STP = STPMAX';
case 104
        str = 'WARNING: STP = STPMIN';
case 201
        str = 'ERROR: STP .LT. STPMIN';
case 202
        str = 'ERROR: STP .GT. STPMAX';
case 203
        str = 'ERROR: INITIAL G .GE. ZERO ';
case 204
        str = 'ERROR: FTOL .LT. ZERO';
case 205
        str = 'ERROR: GTOL .LT. ZERO';
case 206
        str = 'ERROR: XTOL .LT. ZERO';
case 207
        str = 'ERROR: STPMIN .LT. ZERO';
case 208
        str = 'ERROR: STPMAX .LT. STPMIN';
case 212
```

```
        str = 'ERROR: INVALID NBD';
case 213
        str = 'ERROR: NO FEASIBLE SOLUTION';
    otherwise
        str = 'UNRECOGNIZED EXIT FLAG';
end
end
```

Bijlage C: Interviewvragen voor de klinische aspecten

Diagnostiek

1. Voordat een patiënt een videofluoroscopie onderzoek krijgt, heeft hij al meerdere klinische stappen moeten doorlopen. Kunt u ons meer vertellen over dit klinische traject? Wat voor informatie halen jullie uit dit traject?
2. Zijn er ook patiÃńnten met dysfagie van wie geen videofluoroscopie gemaakt worden? Op welke informatie wordt dit gebaseerd?
3. Zijn er andere medische beeldvorming die worden gebruikt naast videofluorscopie voor dysfagie patiÃńnten?
4. Welke kenmerken/aspecten/informatie halen jullie uit videofluoroscopie beelden?
5. Wat voor indeling maken jullie in deze informatie? Bijvoorbeeld:
   (a) Maken jullie gebruik van bepaalde groeperingen tijdens jullie diagnose?
   (b) Gaat dit aan de hand van getallen/percentages of eerder mild/erg?
   (c) Wat zijn de kenmerken per graad?

Behandeling

6. Wat wordt er met de informatie uit de videofluoroscopie onderzoeken gedaan om tot een behandeling te komen? Bijvoorbeeld:
   (a) Multidisciplinair overleg
   (b) Extra onderzoeken
7. Welke behandelmethoden zijn er en wat is het voornaamste doel van deze behandelingen?
8. Waarop wordt de keuze voor een bepaalde behandeling gebaseerd?

dMRI en Deep Learning

9. Wat ziet u als voordelen en beperkingen van de huidige gradatie methode (videofluoroscopie)?
10. Ziet u ook nadelen van dMRI t.o.v. videofluoroscopie? Welk verschil maken deze nadelen voor de diagnose en gradering?
11. Indien de DL in de toekomst in plaats van alleen een beeld ook andere uitkomsten zou kunnen geven, welke uitkomsten zou u dan willen zien? Denkt u aan:
    (a) Kwantitatieve gegevens, zoals afstanden, tijden en volumes
    (b) Kenmerken of pathologiën markeren

(c) Diverse aspecten al automatisch invullen, zoals de PAS-score
(d) Een voorspelling voor het soort pathologie