



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Informasjonsteknologi, kybernetikk/signalbehandling	Vårsemesteret, 2011 Åpen
Forfatter: Roald Klingsheim (signatur forfatter)
Fagansvarlig: Kjersti Engan	
Veileder(e): Kjersti Engan og Trygve Eftestøl	
Tittel på masteroppgaven: Deteksjon av lesjoner i MR-bilder ved bruk av watershedsegmentering	
Studiepoeng:	
Emneord: Bildebehandling, watershedsegmentering, watershed, segmentering, MRI, MR- bilder, lesjoner, lesjondetektering	Sidetall: 41 + vedlegg/annet: 8 Stavanger, 15 juni, 2011 dato/år
1	

15. juni 2011

Sammendrag

Denne rapporten tar for seg deteksjon av lesjoner i MR-bilder av hjernen til demenspasienter ved bruk av watershedsegmentering. Rapporten vil først gi en kort innføring om demens og magnetresonanstomografi (MR) for at leseren skal få en bedre forståelse av hvorfor det vil være gunstig å ha automatisk deteksjon av lesjoner i hjernen. Videre vil det gis en generell innføring rundt segmentering, som er hovedverktøyet brukt for å detektere interessante områder i bilder, før teorien bak morfologisk bildebehandling og watershedalgoritmen blir sett nærmere på. Til slutt vil den foreslåtte segmenteringsalgoritmen for MRI-bildene og programmene som blir brukt for å løse problemet bli gjennomgått, før resultater, konklusjon og forslag til videre arbeid blir presentert.

Forord

Denne rapporten ble skrevet som hovedoppgave etter masterstudie i kybernetikk/signalbehandling ved Universitetet i Stavanger, og jeg vil med dette rette en takk til alle som har hjulpet og støttet meg underveis; både med denne oppgaven og gjennom hele studiet.

En spesiell takk til Kjersti Engan og Trygve Eftestøl ved Universitetet i Stavanger for god hjelp og veiledning, og til Kjell Kristian Grane Torgersen for et høyt antall kaffekopper og gode samtaler.

Stavanger, Juni 2011

Roald Klingsheim

Innhold

1 Innledning	5
1.1 Om demens og magnetresonanstomografi (MR)	5
1.2 Segmentering	5
1.3 Relatert arbeid	6
1.4 Organisering av rapporten	6
2 Teori	7
2.1 Morfologiske operatører	7
2.1.1 Dilasjon	7
2.1.2 Erosjon	9
2.1.3 Åpning og lukking	11
2.2 Morfologisk gradient:	11
2.3 Morfologisk rekonstruksjon	12
2.4 Watershed	14
2.4.1 Watershed med markører	14
2.5 Histogramterskling	15
3 Algoritme og brukte programmer	17
3.1 FSL med Brain Extraction Tool	17
3.2 Algoritme	18
4 Eksperimentelt	22
4.1 Parametre	22
4.1.1 Strukturelementer	22
4.1.2 Terskelverdier	23
4.2 Resultater	25
4.3 Bedømming av resultatene	32
5 Konklusjon og videre arbeid.	39
A MATLAB-kode	42
A.1 Kjøring av algoritmen	43
A.2 loadniftispm.m	45
A.3 finnterskel.m	45
A.4 watershedfunksjon.m	46
A.5 morphgrad.m	47
A.6 fmofr.m	47
A.7 finn_tp_fp_tn_fn.m	48
A.8 finn_grupper.m	49

1 Innledning

Hensikten med arbeidet som gjengis i denne rapporten har vært å produsere en algoritme for automatisk deteksjon av lesjoner i MR-bilder av hjernen til demenspasienter. I dette innledningskapittelet blir det først gitt litt generell informasjon om demens og magnetresonanstomografi som motivasjon for oppgaven før begrepet segmentering forklares nærmere. Videre vil det gis en kort innføring i tidligere, beslektede arbeider, før det til slutt i kapittelet kommer en forklaring på hvordan resten av rapporten er lagt opp.

1.1 Om demens og magnetresonanstomografi (MR)

Demens er en tilstand relatert til dysfunksjon i hjernen, som kan skyldes en rekke ulike sykdommer samt traumatiske skader. Avhengig av hvilken sykdom som forårsaker demensen, eller hvilke hjerneområder som rammes, vil forskjellige funksjoner som for eksempel språk, oppmerksomhet og motorikk kunne rammes. Den nedsatte hjernefunksjonen er ikke-reverserbar, og pasienten blir ofte gradvis verre. Dersom tilstanden opptrer i høy alder, brukes gjerne begrepet aldersdemens.

Magnetresonanstomografi, ofte forkortet MRI (engelsk: magnetic resonance imaging) eller MR, er en teknikk for å fremstille bilder av kroppsvev hos mennesker eller dyr. Ved bruk av MR-teknologi kan det oppnås god avbildning av såkalte bløtdelsstrukturer, som fettvev eller fettholdige organer, eller mer vannholdig vev som blod, muskler og sener. MR-bilder er derfor et veldig nyttig hjelpemiddel ved kartlegging av sykdom i hjerne og sentralnervesystemet. En manuell gjennomgang av MR-bilder ved diagnosestilling kan derimot være ugunstig og tidkrevende siden det genereres store mengder data.

Nasjonalforeningen for folkehelsen opplyser om at rundt 10 000 personer rammes av demens i Norge årlig[7], og det anslås at 30% av de over 90 år har en eller annen form for demens[10]. Med en kommende befolkningsvekst og stadig økende gjennomsnittlig levealder vil demens bli enda mer utbredt, og da tilstanden er ikke-reversibel, vil tidlig deteksjon av sykdommen være viktig. En algoritme som automatisk detekterer demensrelaterte lesjoner i MR-bilder vil dermed forhåpentligvis komme til nytte.

1.2 Segmentering

Innen digital bildebehandling brukes segmentering for å dele et bilde opp i forskjellige regioner, ved det mål å forenkle bildet eller gjøre det lettere å analysere [8]. Bildesegmentering blir typisk brukt for å detektere objekter og forskjellige grenser i bilder. Segmenteringsalgoritmer baseres ofte på et bildes intensitetsverdier, enten ved at en finner brå endringer i intensitet, og dermed kanter og linjer, eller ved at en deler et bilde opp i regioner som har like egenskaper ut fra definerte kriterier. Alle etterfølgende operasjoner, som gjenkjenning og klassifisering av objekter avhenger mye av kvaliteten på segmenteringsprosessen[9]. I

denne oppgaven skal det sees nærmere på watershedsegmentering, som er en attraktiv metode da den blant annet gir lukkede og godt definerte regionsgrenser.

1.3 Relatert arbeid

Segmentering av medisinsk bildemateriell har lenge vært utprøvd på grunn av den åpenbare nyttegraden gode resultater kan ha. Det kan for eksempel hjelpe ved diagnosestilling av sykdom eller som hjelpemiddel ved operasjoner, men å få tilfredsstillende resultater har vært vanskelig. I 1995 ble forskjellige segmenteringsmetoder gjennomgått av L. P. Clarke med flere[11]. Konklusjonen var da at segmentering av medisinske bilder fortsatt var en vanskelig oppgave, og helautomatisk segmentering ga langt fra tilfredsstillende resultater. I 2005 skrev R. A. Heckemann med flere at “funksjonelle og strukturelle avbildninger av hjernen spiller en større og større rolle innen nevrovitenskap og eksperimentell medisin, og den store mengden data fra slike avbildninger gir et økende behov for automatiserte bildeanalyseringsmetoder. Spesielt vil nøyaktige og pålitelige segmenteringsmetoder være viktig for å hente informasjon fra bildene.”[12]. I boken “Computational Surgery and Dual Training” utgitt i 2010 skriver M. Garbey med flere at “segmentering på MR-bilder er ikke åpenbart, og fremdeles en vanskelig oppgave”, før de presenterer en segmenteringsmetode basert på Hidden Markov Chains.[13]I 2003 skrev P. Anbeek med flere “Probabilistic segmentation of white matter lesions in MR imaging” der de presenterer en helautomatisk metode for å segmentere lesjoner i MR-bilder av hjernen, basert på K-nærmeste-nabo klassifisering.[20]

Det finnes med andre ord mange forsøk på segmentering av medisinske bilder ved bruk av forskjellige metoder, men det har ikke blitt funnet publikasjoner der watershedsegmentering har blitt brukt på MR-bilder av hjernen. J. Sijbers med flere skrev i 1997 “Watershed-based segmentation of 3D MR data for volume quantization”, men dette er en semiautomatisk teknikk for volumkvantifisering.[14]

1.4 Organisering av rapporten

Neste kapittel vil gi en grundigere innføring i teorien bak morfologisk bildebehandling og watershedalgoritmen, før algoritmen og programmer som har blitt brukt blir presentert i kapittel 3. I kapittel 4 ses det nærmere på resultater og testing av parametre, før det i kapittel 5 avsluttes med konklusjoner og forslag til videre arbeid.

2 Teori

I dette kapitlet blir en del bakgrunnsteori presentert. De tema og metoder som forklares blir benyttet i algoritmen som presenteres i kapittel 3. Morforlogisk bildebehandling og watershedalgoritmen er viktige bestanddeler av den foreslåtte segmenteringsalgoritmen for MR-bildene og blir derfor relativt grundig forklart. Det er lagt vekt på å presentere bakgrunnsteori slik at det skal være mulig for en medstudent å lese rapporten.

2.1 Morfologiske operasjoner

Ordet morfologi stammer fra de greske ordene $\mu\sigma\phi\eta$, *morphé*, som betyr form; og $\lambda\acute{o}\gamma\omicron\varsigma$, *lógos*, som betyr ord, studie eller læren om. Direkte oversatt betyr det altså “læren om former”. Ordet brukes vanligvis som navn på den grenen innen biologi som har med formen og strukturen til dyr og planter å gjøre. Innen bildebehandling brukes morfologi som et verktøy for å hente ut bildekomponenter som er nyttige for å representere og beskrive en regions form, som for eksempel grenser og skjeletter.[1]I dette delkapitlet vil en del av de morfologiske operatorene en finner innen bildebehandling bli gjennomgått. For å utføre disse operasjonene trengs et strukturelement, en matrise bestående av enere og nuller; 1 og 0. Enerne i matrisen kan settes opp på forskjellige former (for eksempel diamant eller tilnærmet sirkel) avhengig av hvilken effekt man ønsker at de morfologiske operatorene skal ha. Dilasjon med et rundt strukturelement vil for eksempel gi avrundede hjørner på de utvidede objektene.

2.1.1 Dilasjon

Dilasjon, også kalt utvidelse på norsk, er en operasjon som gjør objekter i bilder større, eller tykkere. Strukturelementet føres rundt hele bildet og vil øke lysstyrken til bildeelementer omgitt av naboer med høyere intensitet.. Størrelsen og formen på strukturelementet avgjør hvor mye bildeelementene blir endret, og dermed hvor mye objektene i bildet utvides.

Utvidelse på et gråtonebilde f med strukturelement b , $(f \oplus b)$ er definert som [1]:

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\}$$

der D_b er området til b , og $f(x, y)$ antas å være lik $-\infty$ utenfor området til f .

Videre har vi som oftest at gråtoneutvidelse blir utført med et flatt strukturelement, der verdien, eller høyden, til b er 0 for alle koordinater som omfattes av området D_b , altså:

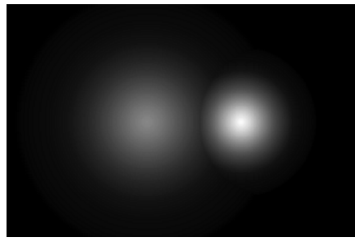
$$b(x', y') = 0 \text{ for } (x', y') \in D_b$$

I dette tilfellet vil max-operatoren kun være spesifisert ved enerne og nullene i D_b -matrisen, slik at vi får forenklet uttrykket for gråtoneutvidelse til:

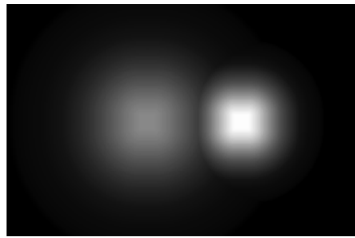
$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') \mid (x', y') \in D_b\}$$

Gråtoneutvidelse blir altså en operator som finner lokale maksima, der maksimumsverdiene blir tatt over et sett med pikselnaboer bestemt av formen på elementene i D_b -matrisen med verdien 1.

Figurene under viser eksempel på gråtonedilatasjon:

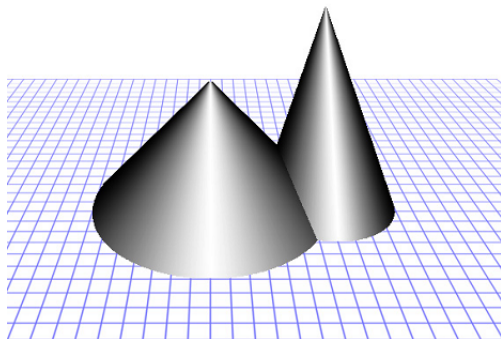


Figur 2.1. Figuren viser et gråtonebilde som skal dilateres.

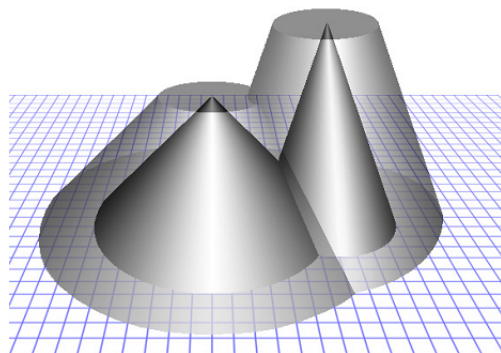


Figur 2.2. Figuren viser den dilaterte versjonen av figur 2.1.

Eller som 3D-objekter:



Figur 2.3. Figuren viser en 3D-versjon av figur 2.1.



Figur 2.4. Figuren viser en 3D-versjon av figur 2.2 som eksempel på dilasjon..

2.1.2 Erosjon

Erosjon er en operasjon som gjør objekter i bilder mindre, eller tynnere. Strukturelementet føres rundt hele bildet og vil redusere lysstyrken til bildeelementer omgitt av naboer med lavere intensitet. Størrelsen og formen på strukturelementet avgjør og hvor mye bildeelementene blir endret, og dermed altså hvor mye objektene i bildet eroderes.

Erosjon på et gråtonebilde f med strukturelement b , $(f \ominus b)$ er definert som [1]:

$$(f \ominus b)(x, y) = \min\{f(x + x', y + y') - b(x', y') \mid (x', y') \in D_b\}$$

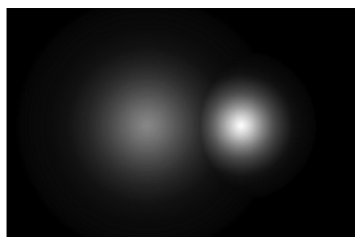
der D_b er området til b , og $f(x, y)$ antas å være lik $+\infty$ utenfor området til f .

Videre kan det forenkles på samme måte som for utvidelse, og for flatt strukturelement får en da:

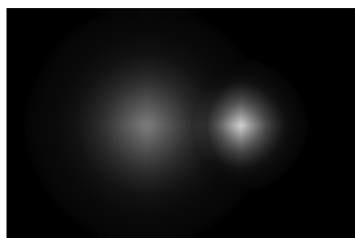
$$(f \ominus b)(x, y) = \min\{f(x + x', y + y') \mid (x', y') \in D_b\}$$

Gråtoneerosjon blir altså en operator som finner lokale minima, der minimumsverdiene blir tatt over et sett med pikselnaboer bestemt av formen på elementene i D_b -matrisen med verdien 1.

Figurene under viser eksempel på gråtoneerosjon:

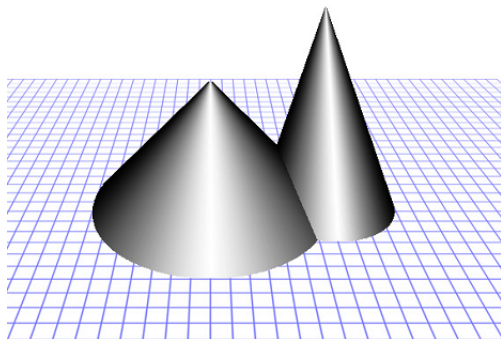


Figur 2.5. Figuren viser et gråtonebilde som skal eroderes.

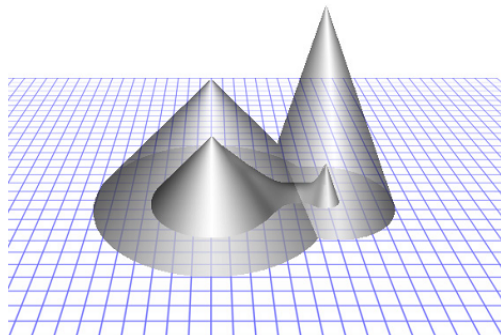


Figur 2.6. Figuren viser den eroderte versjonen av figur 2.5.

Eller som 3D-objekter:



Figur 2.7. Figuren viser en 3D-versjon av figur 2.5.



Figur 2.8. Figuren viser en 3D-versjon av figur 2.6 som eksempel på erosjon..

2.1.3 Åpning og lukking

Morfologisk åpning av et gråtonebilde f med strukturelement b , $f \circ b$ er definert som [1]:

$$f \circ b = (f \ominus b) \oplus b$$

altså en erosjon etterfulgt av en utvidelse. Morfologisk lukking er definert som en utvidelse etterfulgt av en erosjon:

$$f \bullet b = (f \oplus b) \ominus b$$

Generelt blir åpning brukt for å fjerne små, lyse detaljer i bilder, mens alt annet forblir relativt uforandret. På samme måte vil lukking fjerne små mørke detaljer fra bilder. På grunn av dette brukes ofte åpning og lukking i kombinasjon for å glatte ut bilder og fjerne støy.

2.2 Morfologisk gradient:

Innen bildebehandling brukes forskjellige typer gradienter for å detektere kanter, ved at en rask endring fra mørke til lyse verdier (eller omvendt) gir en høy gradientverdi. Ved å kombinere de morfologiske operatorene dilasjon og erosjon, kan vi finne den morfologiske gradienten som [2]:

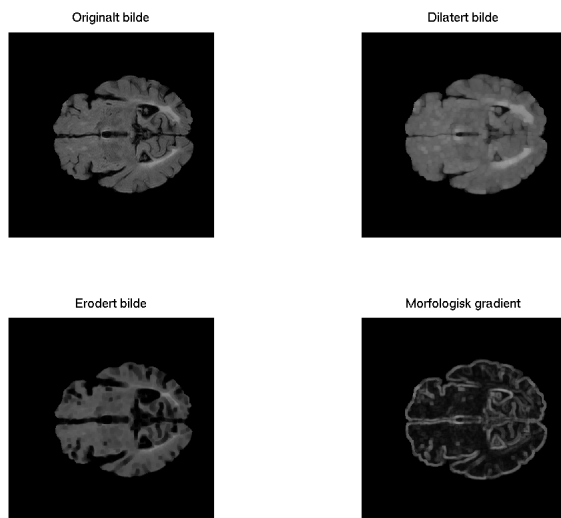
$$grad_{b_e, b_i}(f) = (f \oplus b_e) - (f \ominus b_i)$$

der b_e og b_i er strukturelementer. Den morfologiske gradienten finnes altså ved å trekke det eroderte bildet fra det utvidede bildet, se *figur 2.9* under. Gradienten består forøvrig av to del-gradienter, en ekstern gradient definert som:

$$grad_{ext, b_e}(f) = (f \oplus b_e) - f$$

og en intern gradient definert som:

$$grad_{int,b_i}(f) = f - (f \ominus b_i)$$



Figur 2.9. Figuren viser hvordan en morfologisk gradient finnes ved å trekke den eroderte versjonen av et bilde fra den dilaterte versjonen av samme bilde.

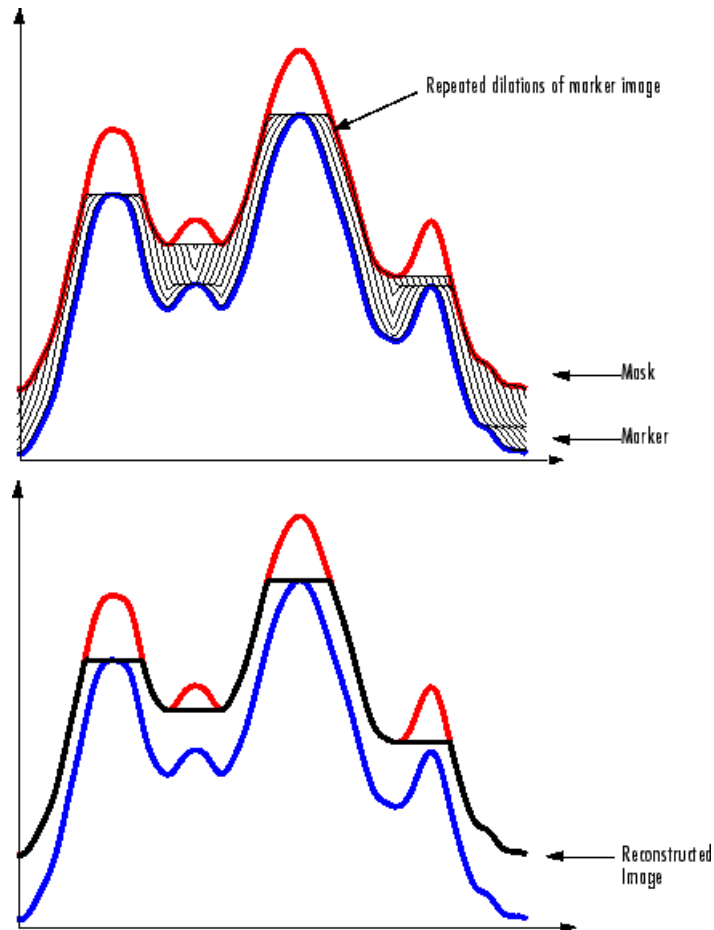
Det finnes også gradienter basert på vanlig filtrering. Originalbildet konvolveres da med et filter, som for eksempel Sobel-filteret, for å få et gradientbilde. Det som gjør at vi kaller det en morfologisk gradient er at det bygger på morfologiske operasjoner, som igjen er avhengig av formen til områder i bildene.

2.3 Morfologisk rekonstruksjon

Rekonstruksjon er en morfologisk transformasjon som bruker to bilder og et strukturelement, i stedet for ett bilde og strukturelement. Det ene bildet er startpunktet for transformasjonen og kalles *markøren*, mens det andre bildet, kalt *masken* begrenser transformasjonen. Hvis G er masken og F er markøren, vil rekonstruksjonen av G fra F , $R_G(F)$, være definert ut fra følgende prosedyre[1]:

1. Initialiser h_1 til å være markørbildet F
2. Lag et strukturelement b
3. Gjenta: $h_{k+1} = (h_k \oplus b) \cap G$ til $h_{k+1} = h_k$
4. $R_G(F) = h_{k+1}$

Morfologisk rekonstruksjon kan altså tenkes som gjentatte utvidelser (dilasjoner) av markørbildet F , inntil konturene av markørbildet “passer inn under” maskebildet G . Se figur 2.10 under.

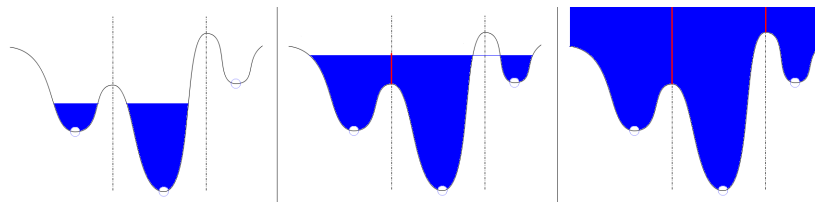


Figur 2.10. Denne figuren er hentet fra [3] og viser morfologisk rekonstruksjon i 1-D. Det foretas utvidelse på markørbildet til det “treffer” maske-bildet.

To andre nyttige rekonstruksjonsteknikker er åpning-ved-rekonstruksjon og lukking-ved-rekonstruksjon. Ved bruk av åpning-ved-rekonstruksjon eroderes først bildet som ved vanlig morfologisk åpning, men i stedet for å deretter ta en lukking, blir det eroderte bildet brukt som markørbilde i en rekonstruksjon med det originale bildet som maske. For å finne lukking-ved-rekonstruksjon tar en først åpning-ved-rekonstruksjon på et invertert bilde, for deretter å igjen invertere resultatet.

2.4 Watershed

Watershed er en segmenteringsmetode først foreslått av Digabel og Lantuejoul-der [4][5] der en ser på et gråtonebilde som en overflate med gråtoneverdier som høyden; mørke bildeelementer har lave verdier og får altså lav høyde, mens hvite bildeelementer har høye verdier, og får dermed større høyde. I litteraturen finnes det flere watershed-algoritmer, men den mest intuitive kan kanskje sies å være den som baserer seg på oversvømmelsessimulering. Hvis en ser for seg at hvert lokale minimumspunkt i bildet blir stukket hull i og en deretter senker overflaten ned i vann, vil vann strømme inn fra forskjellige minima. Når vann fra to forskjellige minimumspunkter møtes bygges det demning for å holde disse adskilte. Til slutt vil oversvømmelsen nå et punkt der kun toppen av alle demningene er synlige, og disse vil tilsvare watershed-linjene. De endelige regionene som oppstår med hvert sitt regionale minimum kalles på engelsk “catchment basins”, eller bare dammer på norsk.

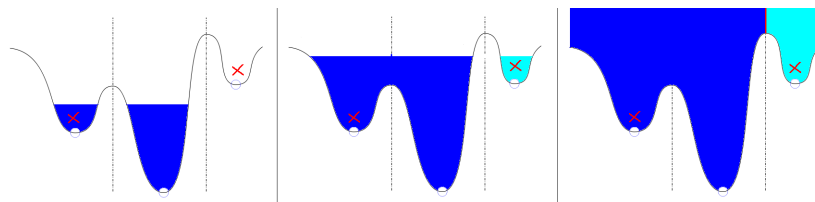


Figur 2.11. Figuren viser intuitivt hvordan watershedmetoden finner regioner basert på minimumspunkter.

For bildesegmentering blir watershed oftest brukt på et gradientbilde, men slike kan inneholde svært mange lokale minimumspunkter som dessverre vil føre til oversegmentering. En effektiv måte å forhindre dette på er å bruke markører[2].

2.4.1 Watershed med markører

Segmentering ved bruk av watershedmetoden vil fungere bedre hvis en får markert forgrunns- og bakgrunnsobjekter. Ved bruk av markører vil det kun bygges demninger dersom vann fra to minimumspunkter med markører møtes, en vil slik ikke lenger ta hensyn til mange av de lokale minimumspunktene, og en slipper ofte oversegmentering.



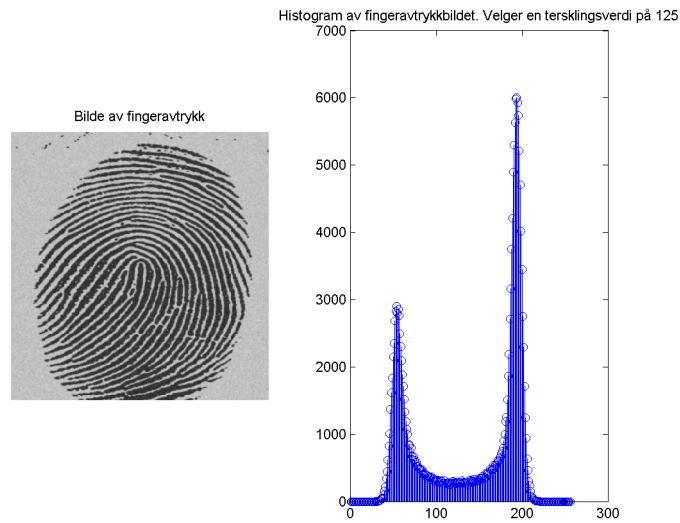
Figur 2.12. Figuren viser hvordan regioner uten markører slås sammen med nærmeste markerte region.

Fra[6] har vi at watershedsegmentering med markører bruker følgende prosedyre:

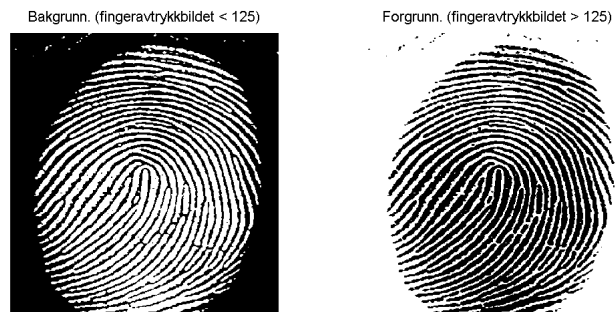
1. Finn en segmenteringsfunksjon. Dette er et bilde der de mørke regionene er objektene det ønskes å segmentere
2. Finn forgrunnsmarkører. Dette er sammenhengende bildeelementer inni hvert objekt
3. Finn bakgrunnsmarkører. Dette er bildeelementer som ikke er en del av noen objekter en ønsker å segmentere
4. Endre segmenteringsfunksjonen slik at den kun har minimumspunkter i forgrunns- og bakgrunnsmarkører
5. Finn watershedtransformen

2.5 Histogramterskling

For å finne forgrunnsmarkørene brukt av watershedmetoden brukes det i den foreslåtte segmenteringsalgoritmen histogramterskling. Ved å finne histogrammet av et gråtonebilde får en en oversikt over hvordan de forskjellige gråtoneverdiene er fordelt. Dette kan videre brukes til å sette en tersklingsverdi, for å kunne trekke ut interessante områder i et bilde. Et enkelt eksempel er vist i *figurene 2.13* og *2.14* under. Eksempelbildet viser et fingeravtrykk med en lysegrå bakgrunn. Av histogrammet til bildet ser en at gråtoneverdiene i bildet klart skiller seg inn i to grupper. Velger så en tersklingsverdi mellom disse toppene, og får med dette delt fingeravtrykkbildet inn i forgrunn og bakgrunn.



Figur 2.13. Figuren viser hvordan histogrammet av et bilde kan brukes for å sette en tersklingsverdi.

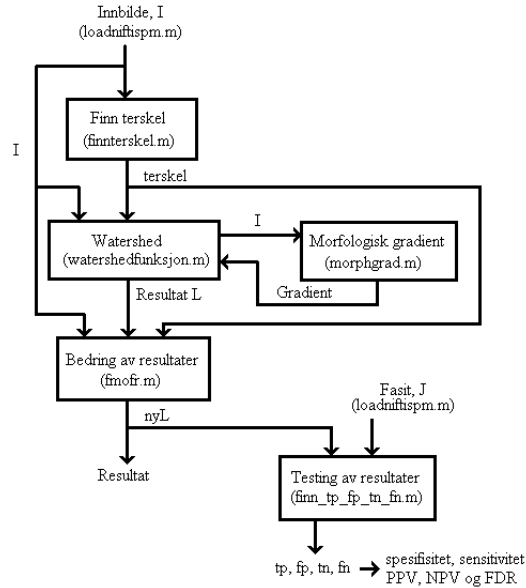


Figur 2.14. Figuren viser bakgrunn og forgrunn av eksempelbildet etter at en tersklingsverdi har blitt utført basert på det opprinnelige bildets histogram.

Med bakgrunn i denne teorien vil algoritmen gjennomgås i neste kapittel.

3 Algoritme og brukte programmer

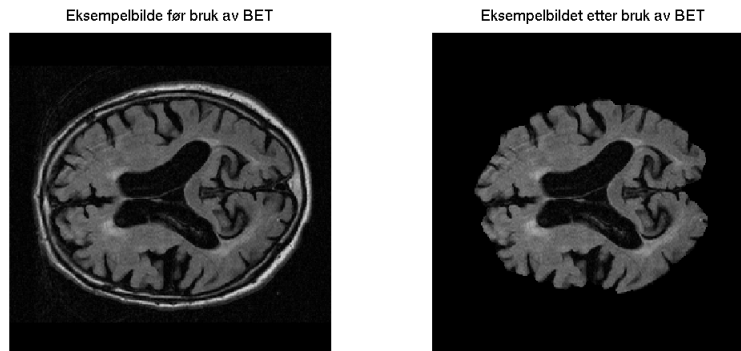
I dette kapittelet blir algoritmen samt andre brukte programmer presentert. Teorien fra kapittel to vil henvises til der den er relevant. *Figur 3.1* under viser en lettfattelig algoritmeoversikt.



Figur 3.1. Figuren viser en flytskjemaoversikt av algoritmen. Matlabfunksjonene som blir brukt er gitt i parentes.

3.1 FSL med Brain Extraction Tool

FSL er et software-bibliotek som inneholder verktøy til bruk ved bildeanalyse av MRI-bilder av hjernen. FSL står for FMRIB Software Library, der FMRIB igjen er en forkortelse av Functional Magnetic Resonance Imaging of the Brain. FSL er hovedsaklig skrevet av medlemmer fra analysegruppen FMRIB Analysis Group, ved Universitetet i Oxford, England[15, 16]. Et av verktøyene som tilbys av FSL er Brain Extraction Tool (BET), som blant annet brukes for å segmentere ut hjernen fra et MR-bilde av hodet[17]. Slik slipper en da ved senere bearbeiding av bildet at potensielt forstyrrende elementer som skallen eller annet vev er tilstede. Et eksempelbilde før og etter at BET har blitt brukt er vist i *figur 3.2* under.



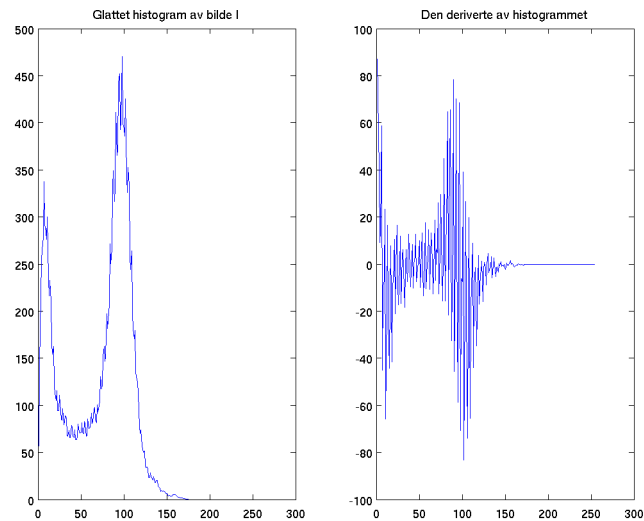
Figur 3.2. Figuren viser et eksempelbilde før og etter bruk av BET (Brain Extraction Tool). En ser at skallen og annet vev fjernes, mens hjernen segmenteres ut.

BET ble brukt på alle pasientbildene før arbeidet med segmenteringsalgoritmen ble startet

3.2 Algoritme

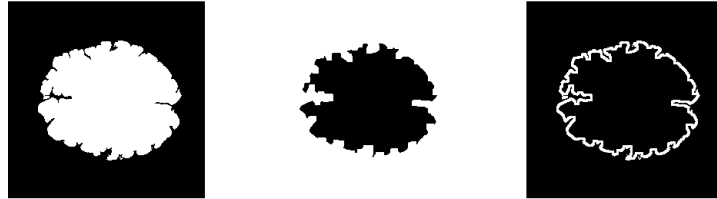
I dette delkapittelet gis en forklaring på hva som blir gjort i segmenteringsalgoritmen. For selve MATLAB-koden, se appendix.

Etter at BET har blitt brukt for å segmentere ut selve hjernen, blir algoritmen brukt på bildemateriellet. Hele pasientvolumet lastes først inn, før ett og ett 2D-bilde deretter blir hentet ut og gitt navnet I . Bildet I brukes deretter som input i funksjonen `finnterskel.m`, der en tersklingsverdi senere brukt for å finne forgrunnsmarkører settes. Dette gjøres ved hjelp av histogramtersking, som forklart i kapittel 2.5. Først finnes en glattet versjon av I s histogram, som deretter deriveres, se *figur 3.3* under. Minimumspunktet av denne deriverte brukes så som utgangspunkt for å finne den endelige tersklingsverdien. Foreløpige forgrunnsmarkører settes så til $I > grense$.

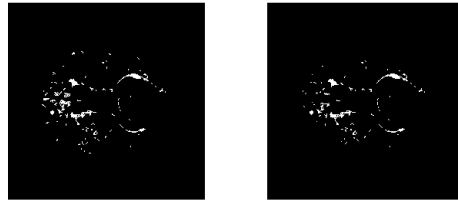


Figur 3.3. Figuren viser det glattede histogrammet av inngangsbildet I , og den deriverte av dette. Minimumspunktet i den deriverte brukes som utgangspunkt for å finne den endelige tersklingsverdien.

Videre foretas det en åpning-lukning-ved-rekonstruksjon på I som forklart i kapittel 2.3. Dette gjøres for å glatte ut I , før en finner bakgrunnsmarkører i bildet, bw . Det glattede bildet, I_{ocrec} , brukes også for å lage en gradient kun rundt hjernen i I , for å forbedre forgrunnsmarkørene. Denne gradienten fjerner punkter i forgrunnsmarkørene som er helt i ytterkanten av hjernen, se *figur 3.4.*



Bildene viser hvordan en enkel gradient finnes. Denne trekkes fra forgrunnsmarkøren for å fjerne ytterpunkter



Figur 3.4. Figuren viser hvordan en enkel gradient om hjernen finnes, ved at hjernen først fylles, og en omvendt, erodert versjon av samme bilde trekkes fra. Denne gradienten trekkes så fra forgrunnsmarkørene for å fjerne ytterpunkter.

Bakgrunnsmarkørene utbedres så ved hjelp av de morfologiske operatorene forklart i kapittel 2.1. Disse bakgrunnsmarkørene brukes så for å finne watershedlinjene, de såkalte watershed ridge lines, wrl , ved å gjøre følgende:

```
D = bwdist(bw4);
DL = watershed(D);
wrl = DL == 0;
```

Den morfologiske gradienten blir så funnet, som forklart i kapittel 2.2, og forsterket ved hjelp av den inverse til wrl . Videre finnes den endelige gradienten ved å bruke MATLAB-kommandoen `imimposemin` på den forsterkede morfologiske gradienten; wrl og forgrunnsmarkørene. Til slutt finnes resultatet av segmenteringsalgoritmen, L .

```
gradmag3 = imimposemin(gradmag2, wrlinv|fgm3);
L = watershed(gradmag3);
```

Etter at resultatene L har blitt funnet lastes disse inn av en ny funksjon, `fmofr`[A.6], for å sjekke hvert enkelt segmenterte område opp mot original-

bildet I . Hvis det viser seg at et resultatsegment har blitt funnet der gjennomsnittsverdien av tilsvarende område i I har en gjennomsnittlig verdi lavere enn en valgt grense, forkastes dette resultatet, og L oppdateres.

Videre sjekkes resultatene opp mot den gitte fasiten, og verdiene tp , fp , tn og fn regnes ut. Dette er henholdsvis verdiene “true positive”, “false positive”, “true negative” og “false negative”. Disse verdiene kan blant annet brukes til å finne sensitivitets- og spesifisitetstall, se kapittel 4.3..

Kort algoritmeoversikt:

```
.
for {et antall pasientvolum}
  x = pasientbilder;
  y = fasitbilder;
  for {alle enkeltbilder i et volum}
    I = bilde;
    J = fasit;
    finnterskel: {input: I. output: terskel}
    grense = terskel + 0.06;
    watershedfunksjon: {input: I, grense. output: resultater L,
markører bw og fgm}
    forbedring av resultater: (input: L, I og terskel. output:
bedrede resultater nyL)
    finn_tp_fp_tn_fn: (input: nyL, J. output: tp, fp, tn, fn)
  end
  finn total tp, fp, tn og fn pr volum
  lagre resultater pr volum
end
.
```

4 Eksperimentelt

Dette kapitlet inneholder en gjennomgang av de forskjellige parametrene som settes i algoritmen, samt resultater og kommentarer til disse. I tillegg gis det først en kort innføring rundt de tallene som bedømmer resultatene.

4.1 Parametre

De forskjellige parametrene som settes i algoritmen, kan deles inn i to grupper. Den ene er *strukturelementer* som brukes for å utføre morfologiske operasjoner, som forklart i kapittel 2.1. Den andre er *grenseverdier*, som settes i forbindelse med terskling av bilder.

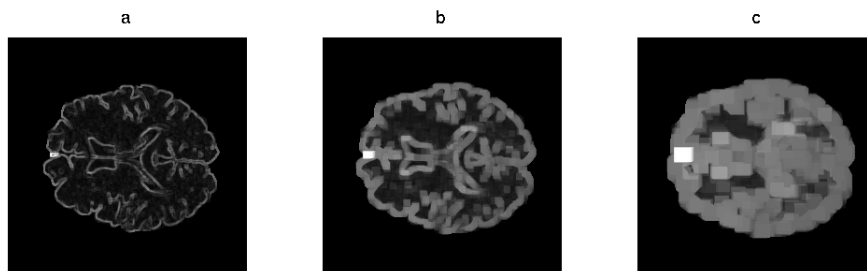
4.1.1 Strukturelementer

I algoritmen settes tre forskjellige strukturelementer, *se*; *se2*; og *se_morph*. Disse tre er alle satt etter subjektive meninger om resultatene de gav.

se Strukturelementet *se* settes i forbindelse med den morfologiske operasjonen åpning-lukking-ved-rekonstruksjon som gjøres på bildet *I*. Verdien settes til $se = strel('square', 7)$; , altså er dette et kvadratisk strukturelement bestående av 7x7 enere. Dette er et relativt stort strukturelement, som ble valgt fordi åpning-lukking-ved-rekonstruksjon gjøres for å glatte bilde *I*. Det gis da god glatting av bildet, uten at alle detaljer blir visket ut. *se* brukes også i forbindelse med gradienten om hjernen, som brukes for å forbedre forgrunnsmarkørene. Etter liten test viste at særlig større strukturelementer gav en for bred gradient, slik at relevant data kunne blitt fjernet, men mindre strukturelementer gav en for tynn gradient, slik at fortgrunnsmarkørene ikke ble forbedret.

se2 Strukturelementet *se2* settes i forbindelse med morfologiske operasjoner på bakgrunnsmarkørene, som brukes for å finne watershedlinjene *wrl*. Strukturelementet *se2* ble satt til $se2 = strel('square', 3)$; , som er relativt lite, dette fordi det ikke ønsktes særlig store endringer.

se_morph Det siste strukturelementet som settes i algoritmen er $se_morph = strel('square', 3)$; . Dette brukes for å finne den morfologiske gradienten. Størrelsen er relativt liten, og den samme som *se2*, men den settes inne i *morphgrad*-funksjonen, og nevnes derfor her. Et lite strukturelement gir skarpere kanter i den morfologiske gradienten og har derfor blitt valgt, noe *figur 4.1*. tydelig viser. De tre bildene viser morfologiske gradienter funnet med strukturelementer av størrelse på henholdsvis 3x3, 7x7 og 15x15.



Figur 4.1. Figuren viser tre morfologiske gradienter, funnet med strukturelementer av størrelse på henholdsvis a) 3×3 , b) 7×7 , c) 15×15 .

4.1.2 Terskelverdier

I algoritmen brukes grenseverdier etter terskling til å finne forgrunnsmarkører, samt som en ekstra test av resultatet ved å sjekke hvert enkelt resultatsegment opp mot originalbildet I . Hvis det viser seg at et resultatsegment har blitt funnet der gjennomsnittsverdien av tilsvarende område i I har en gjennomsnittlig verdi lavere enn en valgt grense, forkastes dette resultatet, og L oppdateres.

Disse grenseverdiene har blitt satt ut fra testing, og blitt valgt ut fra resultatene de gir i spesifisitet og sensitivitet. Dette er verdier som gir et mål på hvor bra algoritmen fungerer[18]. Testingen har tatt utgangspunkt i at terskelverdien som ble funnet av funksjonen *finnterskel* [A.3] erfaringsmessig ble satt for lavt, slik at forgrunnsmarkørene dekket et for stort område. Denne terskelverdien har derfor blitt økt med en rekke verdier og blitt brukt ved gjennomkjøring av algoritmen for et sett av pasientvolumene, kalt “Gruppe 1”. Dette er en gruppe bestående av 33 av de 102 tilgjengelige pasientvolumene, for mer info se appendiks A.8.

Videre har verdien som settes når resultatet sjekkes blitt prøvd satt til flere forskjellige verdier. Disse verdiene har blant annet vært basert på tersklingsverdien *finnterskel* gir ut, men også konstante verdier har blitt satt og prøvd ut. Tabellene nedenfor viser resultatene i spesifisitet og sensitivitet for fjorten forskjellige tersklingsverdier, med to forskjellige grenser satt ved sjekkingen av resultatene; $\text{grense}/4$ og *finnterskel*.

grense/4:

	Spesifisitet	Sensitivitet
finn+075	0.3192	0.9606
finn+050	0.3409	0.9416
finn+025	0.4118	0.8858
finn+020	0.439	0.8114
finn+015	0.465	0.7136
finn+010	0.6003	0.6079
finn+009	0.6639	0.6135
finn+008	0.7471	0.636
finn+007	0.8015	0.6717
finn+006	0.8369	0.7146
finn+005	0.8786	0.7079
finn+004	0.9085	0.6426
finn+002	0.9214	0.3767
finn+0	0.9444	0.1412

Tabell 1. Viser sensitivitets- og spesifisitetsverdier ved forskjellige tersklingsverdier, med grense/4 satt i fmofr [A.6]

finnterskel:

	Spesifisitet	Sensitivitet
finn+075	0,9529	0,0753
finn+050	0,962	0,0672
finn+025	0,9783	0,0575
finn+020	0,9852	0,0723
finn+015	0,9884	0,1916
finn+010	0,9947	0,4155
finn+009	0,9958	0,4807
finn+008	0,9953	0,5516
finn+007	0,9932	0,6291
finn+006	0,9897	0,6908
finn+005	0,9845	0,6946
finn+004	0,9782	0,6364
finn+002	0,9709	0,3749
finn+0	0,9862	0,1401

Tabell 2. Viser sensitivitets- og spesifisitetsverdier ved forskjellige tersklingsverdier, med finnterskel satt i fmofr [A.6]

Ser av tabellene at de beste sensitivitets- og spesifisitetsverdiene finnes med tersklingsverdiene “finn+005” og “finn+006”. I neste delkapittel ses det nærmere på tersklingsverdien “finn+006”.

4.2 Resultater

I dette delkapittelet vil resultatene for alle pasientvolumene for grensen “*finn+006*” bli gjennomgått. Grensen satt ved sjekkingen av resultatene i funksjonen *fmo_{fr}*, se appendiks A.6, er her satt til $\text{grense}/4$ og *finnterskel*, da disse gir noen av de beste resultatene. Resultatene vil bli presentert i tabeller ordnet etter gruppenummer, se appendiks A.8. Tallene i tabellen viser spesifisitet, sensitivitet, positiv prediktiv verdi (PPV), negativ prediktiv verdi (NPV) og FDR (false discovery rate). I tillegg vises gjennomsnittlig grense satt for hvert enkelt pasientvolum.

006 grense/4 gruppe1

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
2	0.8595	0.9645	0.1025	0.9993	0.8975	0.4488
11	0.8975	0.8485	0.0486	0.9990	0.9514	0.3472
13	0.9284	0.6914	0.0802	0.9970	0.9198	0.3788
16	0.8935	0.5008	0.0486	0.9940	0.9514	0.4265
20	0.5849	0.9645	0.1424	0.9957	0.8576	0.3631
21	0.8444	0.9105	0.0582	0.9989	0.9418	0.3650
25	0.9224	0.5503	0.0905	0.9932	0.9095	0.4111
26	0.8620	0.2548	0.0787	0.9616	0.9213	0.3936
27	0.9167	0.7513	0.0425	0.9987	0.9575	0.4136
34	0.9103	0.9459	0.0849	0.9995	0.9151	0.3986
35	0.9109	0.6056	0.1259	0.9909	0.8741	0.3486
44	0.9216	0.8491	0.2013	0.9962	0.7987	0.3673
48	0.8207	0.8902	0.0149	0.9996	0.9851	0.4158
51	0.5848	0.5871	0.0239	0.9879	0.9761	0.3549
52	0.8922	0.7767	0.0338	0.9988	0.9662	0.3251
58	0.9494	0.4950	0.0713	0.9958	0.9287	0.3403
61	0.9261	0.6113	0.1448	0.9915	0.8552	0.4383
62	0.8454	0.2580	0.0644	0.9651	0.9356	0.3605
63	0.8627	0.6897	0.0130	0.9991	0.9870	0.3723
65	0.4057	0.7999	0.0382	0.9856	0.9618	0.3974
66	0.7760	0.7956	0.0215	0.9984	0.9785	0.4110
68	0.9093	0.7594	0.0509	0.9983	0.9491	0.3859
69	0.9253	0.8181	0.0739	0.9986	0.9261	0.3954
75	0.9109	0.9224	0.0346	0.9997	0.9654	0.3641
76	0.6669	0.5218	0.0232	0.9892	0.9768	0.3166
77	0.8041	0.8676	0.0594	0.9977	0.9406	0.4149
84	0.7194	0.7306	0.0188	0.9972	0.9812	0.4055
85	0.8865	0.4424	0.1469	0.9730	0.8531	0.4020
88	0.8113	0.4507	0.0456	0.9866	0.9544	0.4376
94	0.8789	0.7481	0.0905	0.9954	0.9095	0.3618
99	0.7741	0.8246	0.0678	0.9955	0.9322	0.3861
100	0.9076	0.8912	0.0190	0.9998	0.9810	0.3533
102	0.9086	0.8641	0.1304	0.9976	0.8696	0.3386
Avg.	0,8369	0,7145	0,0694	0,9931	0,9305	0,3830

Tabell 3. Viser resultatene for pasientvolumene i gruppe 1 ved tersklingsverdi "finnterskel+0.06", med grense/4 satt i fmofr [A.6]

006 grense/4 gruppe2

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
1	0.7374	0.7799	0.1573	0.9816	0.8427	0.3456
3	0.9221	0.2835	0.0606	0.9864	0.9394	0.4703
5	0.8464	0.8197	0.0196	0.9992	0.9804	0.4430
7	0.8509	0.8964	0.0351	0.9993	0.9649	0.4186
14	0.9334	0.8790	0.0426	0.9996	0.9574	0.3893
15	0.5033	0.7730	0.0158	0.9954	0.9842	0.3825
19	0.9123	0.8623	0.1113	0.9981	0.8887	0.3886
22	0.9221	0.3948	0.2960	0.9484	0.7040	0.3409
23	0.7437	0.5453	0.0118	0.9966	0.9882	0.3597
24	0.9178	0.9117	0.0954	0.9991	0.9046	0.3810
28	0.8477	0.7601	0.1792	0.9878	0.8208	0.3765
29	0.9205	0.6857	0.0723	0.9969	0.9277	0.3492
30	0.8968	0.2981	0.0214	0.9941	0.9786	0.3880
32	0.9635	0.8763	0.5270	0.9941	0.4730	0.4593
36	0.9487	0.7588	0.0776	0.9986	0.9224	0.3805
47	0.7502	0.7639	0.0673	0.9926	0.9327	0.3931
49	0.8158	0.8554	0.0633	0.9974	0.9367	0.3882
50	0.7889	0.7425	0.0595	0.9942	0.9405	0.4239
54	0.2495	0.7580	0.0077	0.9926	0.9923	0.3518
55	0.6777	0.8473	0.0235	0.9979	0.9765	0.4044
56	0.9278	0.8593	0.1161	0.9983	0.8839	0.3459
59	0.8011	0.8338	0.1195	0.9933	0.8805	0.3331
60	0.8464	0.5132	0.0291	0.9949	0.9709	0.3102
64	0.9043	0.8858	0.0240	0.9997	0.9760	0.3944
74	0.9100	0.4436	0.2837	0.9532	0.7163	0.4182
78	0.9380	0.6292	0.0812	0.9966	0.9188	0.4417
82	0.7986	0.8792	0.0183	0.9994	0.9817	0.5077
83	0.9275	0.8040	0.0751	0.9985	0.9249	0.3579
87	0.8501	0.3870	0.0327	0.9906	0.9673	0.3443
91	0.5176	0.8025	0.0742	0.9820	0.9258	0.3243
93	0.9433	0.2301	0.1986	0.9525	0.8014	0.5427
98	0.9888	0.2231	0.1002	0.9956	0.8998	0.4001
101	0.7917	0.8457	0.0821	0.9957	0.9179	0.3784
Avg.	0,8270	0,6917	0,0963	0,9909	0,9036	0,3919

Tabell 4. Viser resultatene for pasientvolumene i gruppe 2 ved tersklingsverdi "finnterskel+0.06", med grense/4 satt i fmojr [A.6]

006 grense/4 gruppe3

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
4	0.9280	0.3835	0.2577	0.9585	0.7423	0.4464
6	0.9417	0.5322	0.0644	0.9963	0.9356	0.4235
8	0.8889	0.9498	0.0367	0.9997	0.9633	0.3786
10	0.8660	0.1619	0.0064	0.9949	0.9936	0.4681
12	0.9317	0.9216	0.0497	0.9997	0.9503	0.3544
17	0.6018	0.9284	0.2102	0.9866	0.7898	0.3838
18	0.9400	0.7142	0.1010	0.9971	0.8990	0.4308
31	0.8729	0.3269	0.1243	0.9592	0.8757	0.3546
33	0.8655	0.9524	0.0239	0.9998	0.9761	0.3912
37	0.7941	0.7833	0.0343	0.9975	0.9657	0.4030
38	0.8158	0.6464	0.0861	0.9885	0.9139	0.3357
41	0.9374	0.7954	0.0640	0.9988	0.9360	0.5017
43	0.9602	0.8284	0.0982	0.9991	0.9018	0.4588
46	0.8990	0.9125	0.1732	0.9977	0.8268	0.3714
53	0.6594	0.8483	0.0589	0.9942	0.9411	0.3576
57	0.6716	0.5696	0.0085	0.9969	0.9915	0.3850
70	0.9821	0.5067	0.3448	0.9907	0.6552	0.3950
72	0.7894	0.9196	0.0162	0.9996	0.9838	0.4003
73	0.9147	0.8609	0.1696	0.9969	0.8304	0.3877
79	0.8999	0.6298	0.0763	0.9946	0.9237	0.3889
80	0.8976	0.8664	0.0504	0.9991	0.9496	0.4395
86	0.9343	0.6025	0.2169	0.9873	0.7831	0.4710
89	0.8842	0.8414	0.0865	0.9977	0.9135	0.3868
90	0.9528	0.8468	0.0881	0.9991	0.9119	0.3773
95	0.9310	0.7622	0.0142	0.9997	0.9858	0.3383
97	0.9535	0.9308	0.3010	0.9984	0.6990	0.3523
Avg.	0,8735	0,7316	0,1062	0,9933	0,8937	0,3992

Tabell 5. Viser resultatene for pasientvolumene i gruppe 3 ved tersklingsverdi “finnterskel+0.06”, med grense/4 satt i fmofr [A.6]

006 finnterskel gruppe1

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
2	0.9847	0.9579	0.5108	0.9993	0.4892	0.4488
11	0.9799	0.8479	0.2061	0.9990	0.7939	0.3472
13	0.9863	0.6816	0.3101	0.9971	0.6899	0.3788
16	0.9863	0.6816	0.3101	0.9971	0.6899	0.3788
20	0.9918	0.9599	0.8934	0.9971	0.1066	0.3631
21	0.9895	0.9049	0.4776	0.9990	0.5224	0.3650
25	0.9632	0.5499	0.1732	0.9935	0.8268	0.4111
26	0.9858	0.2548	0.4530	0.9662	0.5470	0.3936
27	0.9962	0.7506	0.4930	0.9988	0.5070	0.4136
34	0.9891	0.9459	0.4317	0.9995	0.5683	0.3986
35	0.9920	0.6056	0.6158	0.9916	0.3842	0.3486
44	0.9828	0.8472	0.5344	0.9964	0.4656	0.3673
48	0.9959	0.8740	0.3902	0.9996	0.6098	0.4158
51	0.9970	0.4743	0.7321	0.9910	0.2679	0.3549
52	0.9898	0.7767	0.2703	0.9989	0.7297	0.3251
58	0.9864	0.4946	0.2221	0.9960	0.7779	0.3403
61	0.9840	0.6113	0.4391	0.9920	0.5609	0.4383
62	0.9925	0.1340	0.4231	0.9652	0.5769	0.3605
63	0.9957	0.6826	0.2938	0.9992	0.7062	0.3723
65	0.9971	0.6901	0.8742	0.9909	0.1258	0.3974
66	0.9847	0.7908	0.2415	0.9987	0.7585	0.4110
68	0.9807	0.7513	0.2001	0.9984	0.7999	0.3859
69	0.9761	0.8144	0.1987	0.9986	0.8013	0.3954
75	0.9895	0.9224	0.2338	0.9997	0.7662	0.3641
76	0.9982	0.2735	0.7021	0.9891	0.2979	0.3166
77	0.9909	0.8676	0.5769	0.9981	0.4231	0.4149
84	0.9958	0.6605	0.5355	0.9975	0.4645	0.4055
85	0.9871	0.4424	0.6014	0.9757	0.3986	0.4020
88	0.9783	0.4380	0.2874	0.9886	0.7126	0.4376
94	0.9928	0.7455	0.6262	0.9959	0.3738	0.3618
99	0.9881	0.7905	0.5685	0.9958	0.4315	0.3861
100	0.9926	0.8912	0.1937	0.9998	0.8063	0.3533
102	0.9909	0.8633	0.6017	0.9978	0.3983	0.3386
Avg.	0.9882	0.6962	0.4431	0.9939	0.5569	0.3815

Tabell 6. Viser resultatene for pasientvolumene i gruppe 1 ved tersklingsverdi "finnterskel+0.06", med finnterskel satt i fmofr [A.6]

006 finnterskel gruppe2

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
1	0.9913	0.7075	0.8356	0.9818	0.1644	0.3456
3	0.9812	0.2835	0.2110	0.9872	0.7890	0.4703
5	0.9960	0.8182	0.4319	0.9993	0.5681	0.4430
7	0.9861	0.8964	0.2810	0.9994	0.7190	0.4186
14	0.9879	0.8790	0.1970	0.9996	0.8030	0.3893
15	0.9948	0.6361	0.5577	0.9962	0.4423	0.3825
19	0.9910	0.8623	0.5494	0.9982	0.4506	0.3886
22	0.9922	0.3915	0.8072	0.9516	0.1928	0.3409
23	0.9933	0.5407	0.3137	0.9974	0.6863	0.3597
24	0.9855	0.9114	0.3737	0.9991	0.6263	0.3810
28	0.9903	0.7595	0.7739	0.9895	0.2261	0.3765
29	0.9895	0.6857	0.3716	0.9971	0.6284	0.3492
30	0.9871	0.2981	0.1488	0.9947	0.8512	0.3880
32	0.9759	0.8763	0.6282	0.9942	0.3718	0.4593
36	0.9866	0.7588	0.2431	0.9986	0.7569	0.3805
47	0.9899	0.6887	0.6161	0.9926	0.3839	0.3931
49	0.9915	0.8542	0.5940	0.9979	0.4060	0.3882
50	0.9858	0.6832	0.4634	0.9943	0.5366	0.4239
54	0.9986	0.4165	0.6888	0.9955	0.3112	0.3518
55	0.9969	0.8210	0.7046	0.9984	0.2954	0.4044
56	0.9880	0.8593	0.4404	0.9984	0.5596	0.3459
59	0.9956	0.7928	0.8549	0.9933	0.1451	0.3331
60	0.9950	0.5104	0.4762	0.9956	0.5238	0.3102
64	0.9923	0.8858	0.2349	0.9997	0.7651	0.3944
74	0.9831	0.4436	0.6786	0.9565	0.3214	0.4182
78	0.9736	0.6292	0.1719	0.9967	0.8281	0.4417
82	0.9936	0.8615	0.3645	0.9994	0.6355	0.5077
83	0.9912	0.8036	0.3993	0.9986	0.6007	0.3579
87	0.9896	0.3854	0.3264	0.9919	0.6736	0.3443
91	0.9928	0.7272	0.8291	0.9869	0.1709	0.3243
93	0.9812	0.2301	0.4276	0.9543	0.5724	0.5427
98	0.9901	0.2231	0.1118	0.9957	0.8882	0.4001
101	0.9905	0.8407	0.6605	0.9965	0.3395	0.3784
Avg.	0,9893	0,6654	0,4777	0,9917	0,5222	0,3919

Tabell 7. Viser resultatene for pasientvolumene i gruppe 2 ved tersklingsverdi "finnterskel+0.06", med finnterskel satt i fmofr [A.6]

006 finnterskel gruppe3

Volum nr.	Spesifisitet	Sensitivitet	PPV	NPV	FDR	Gj.snittlig grense
4	0.9834	0.3694	0.5914	0.9599	0.4086	0.4464
6	0.9788	0.5306	0.1590	0.9964	0.8410	0.4235
8	0.9923	0.9453	0.3526	0.9998	0.6474	0.3786
10	0.9751	0.1402	0.0291	0.9953	0.9709	0.4681
12	0.9890	0.9216	0.2447	0.9997	0.7553	0.3544
17	0.9490	0.8021	0.6422	0.9767	0.3578	0.3838
18	0.9657	0.7142	0.1646	0.9972	0.8354	0.4308
31	0.9943	0.2123	0.6708	0.9581	0.3292	0.3546
33	0.9946	0.9524	0.3776	0.9998	0.6224	0.3912
37	0.9691	0.7727	0.1893	0.9978	0.8107	0.4030
38	0.9940	0.6219	0.7350	0.9899	0.2650	0.3357
41	0.9855	0.7856	0.2256	0.9988	0.7744	0.5017
43	0.9630	0.8273	0.1049	0.9991	0.8951	0.4588
46	0.9818	0.9125	0.5372	0.9979	0.4628	0.3714
53	0.9929	0.8342	0.7466	0.9958	0.2534	0.3576
57	0.9989	0.2398	0.5078	0.9963	0.4922	0.3850
70	0.9849	0.5067	0.3849	0.9908	0.6151	0.3950
72	0.9898	0.9086	0.2505	0.9997	0.7495	0.4003
73	0.9816	0.8609	0.4865	0.9971	0.5135	0.3877
79	0.9878	0.6261	0.4027	0.9951	0.5973	0.3889
80	0.9913	0.8664	0.3857	0.9992	0.6143	0.4395
86	0.9768	0.6025	0.4394	0.9879	0.5606	0.4710
89	0.9904	0.8389	0.5334	0.9979	0.4666	0.3868
90	0.9863	0.8468	0.2505	0.9992	0.7495	0.3773
95	0.9876	0.7622	0.0744	0.9997	0.9256	0.3383
97	0.9868	0.9308	0.6019	0.9985	0.3981	0.3523
Avg.	0,9834	0,7050	0,3880	0,9932	0,6119	0,3992

Tabell 8. Viser resultatene for pasientvolumene i gruppe 1 ved tersklingsverdi "finnterskel+0.06", med finnterskel satt i fmofr [A.6]

4.3 Bedømming av resultatene

I dette delkapittelet vil resultatene gjennomgås og kommenteres. Først vil det gis en kort innføring i de verdiene som blir brukt for å bedømme resultatene. Som forklart i kapittel 3.2. finnes verdiene tp , fp , tn og fn i algoritmen. Dette er de fire forskjellige utfallene en kan ha når en prøver å klassifisere noe; tp er de sanne positive resultatene, altså i denne oppgavens tilfelle riktig klassifisert lesjon. tn er tilsvarende sanne negative, og dermed her riktig klassifisert hjerne-masse, eller bakgrunn. Videre kan en detektere lesjoner der det ikke faktisk er dette - dette er verdien fp , eller falske positive. Til slutt har en også verdien fn , falske negative, som her vil si at et lesjonsområde ble klassifisert som bakgrunn. Disse verdiene brukes videre for å finne de fem verdiene spesifisitet, sensitivitet, positiv prediktiv verdi, negativ prediktiv verdi og FDR som følger[19]:

$$\text{Sensitivitet} = \frac{tp}{(tp+fn)}$$

$$\text{Spesifisitet} = \frac{tn}{(fp+tn)}$$

$$\text{PPV} = \frac{tp}{(tp+fp)}$$

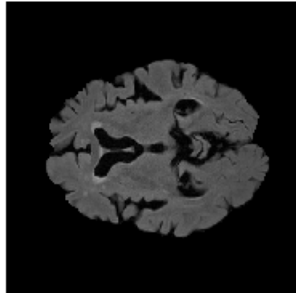
$$\text{NPV} = \frac{tn}{(tn+fn)}$$

$$\text{FDR} = \frac{fp}{(fp+tp)}$$

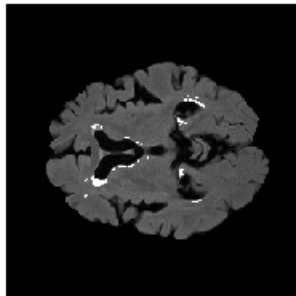
Sensitivitet kalles gjerne også for treff-rate, og er et mål på hvor mange av de faktiske positive som ble riktig klassifisert. Spesifisitet er tilsvarende et mål på hvor mange av de faktiske negative som ble klassifisert som dette. Begge disse verdiene ønskes så nær 1 som mulig. PPV og FDR er rater med den sammenheng at den ene er lik 1 minus den andre. PPV gir et mål på hvor stor del de sanne positive utgjør av alt som ble klassifisert som treff. Dette er også en verdi som ønskes så nær 1 som mulig. Tilsvarende viser FDR antall falske positive i alt som ble klassifisert som treff; denne verdien ønskes nær null. Videre har en NPV som gir en rate på hvor mange av de negative elementene (her: bakgrunn) som ble riktig klassifisert.

Med bakgrunn i dette, samt gjennomsnittsverdiene nederst i hver tabell i kapittel 4.2, ser en at resultatene - i alle fall gjennomsnittlig sett - er ganske dårlige. Dette på tross av at tersklingsverdiene ble valgt på grunnlag av at de gav så gode spesifisitets- og spesifisitetsverdier som mulig. En gjennomsnittlig FDR-verdi i tabell 3 på 0,9305 vil si at mer enn 93% av de detekterte positive verdiene, ikke skulle ha vært det. En ser av tabellene 3, 4 og 5 at grense/4 som tersklingsverdi gir veldig høye FDR-verdier, og tilsvarende lave PPV-verdier. Hvis en ser nærmere på pasientvolum 97 i tabell 5, ser en at spesifisitets- og sensitivitetsverdiene er veldig gode, henholdsvis 0.9535 og 0.9308. Allikevel viser PPV-raten at bare 30% av de detekterte positive faktisk er dette. Pasientvolum 97 med fasit og resultater er vist i *figur 4.2* under.

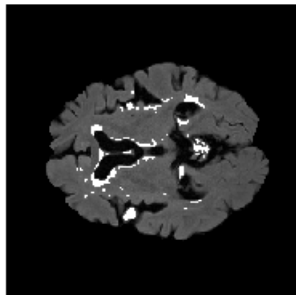
Bilde 15 i pasientvolum 97



Bilde 15 i pasientvolum 97, med fasit



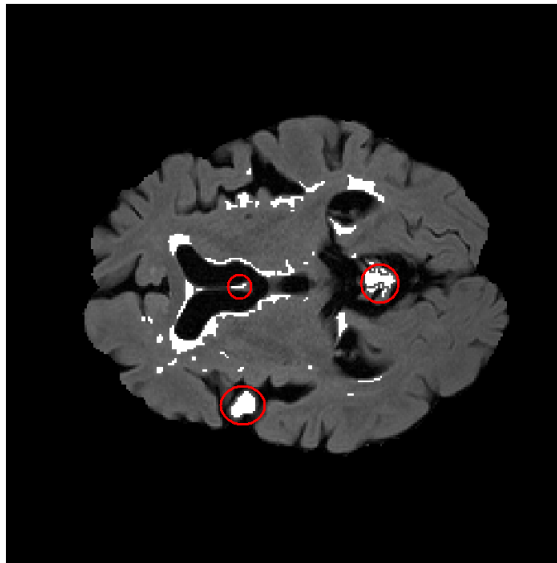
Bilde 15 i pasientvolum 97 med resultater



Figur 4.2. Figuren viser bilde 15 fra pasientvolum 97, tabell 4. De tre bildene er henholdsvis det originale bildet, det originale bildet med fasit overlagt og det originale bildet med resultater overlagt.

I *figur 4.3* under ser en nærmere på resultatbildet for pasientvolum 97. Detekterte områder en kan se bort fra er markert med rødt. De midterste områdene er ikke lesjonsområder, og kan ses bort fra, videre er det nederste området tydelig hjernemasse, og ikke lesjon. Dette vil bedre resultatene noe, men en har fortsatt for mange detekterte positive - i tillegg trengs en manuell gjennomgang av resultatene for å utelukke disse områdene.

Bilde 15 i pasientvolum 97 med resultater



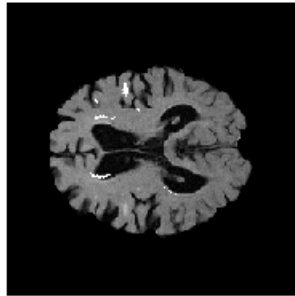
Figur 4.3. Figuren viser bilde 15 fra pasientvolum 97, med resultater overlatt. Områdene markert med rødt er feildetekterte lesjoner en enkelt kan se bort fra grunnet plassering.

Hvis en ser på resultatene i tabellene 6, 7 og 8, ser en at resultatene med tersklingsverdien finnterskel er, gjennomsnittlig, noe bedre. FDR-ratene er ikke like høye som for grense/4, men de er fortsatt for høye. I *figur 4.4* ser en nærmere på pasientvolum 10 fra tabell 8. Dette volumet har en FDR-verdi på hele 97%, som vil si at bare 3% av de detekterte positive områdene faktisk er det. Resultatene kan til en viss grad klandres på en dårlig fasit, men de detekterte områdene er uansett feil, og resultatet særs dårlig.

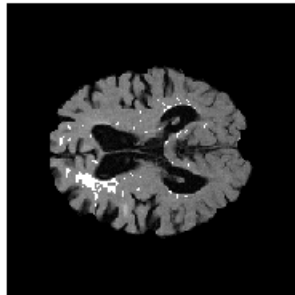
Bilde 15 i pasientvolum 10



Bilde 15 i pasientvolum 10, med fasit



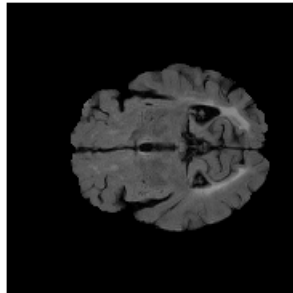
Bilde 15 i pasientvolum 10 med resultater



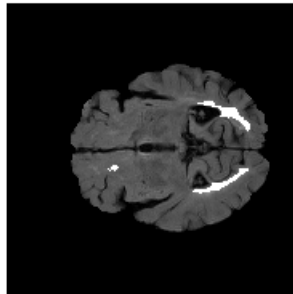
Figur 4.4. Figuren viser bilde 15 fra pasientvolum 10, tabell 8. De tre bildene er henholdsvis det originale bildet, det originale bildet med fasit overlagt og det originale bildet med resultater overlagt.

En har også noen oppløftende resultater, som for pasientvolum 20 gjengitt i tabell 6. Volumet har en spesifisitet på 0.9918, en sensitivitet på 0.9599 og en FDR-verdi på 10,66%. Resultatene er vist under, i *figur 4.5*. En vil se at flere små områder som fasiten ikke inneholder. Disse blir feil i forhold til fasit, men områdene er såpass lyse at det her kan være snakk om lesjoner, og at fasiten i dette tilfellet er ufullstendig.

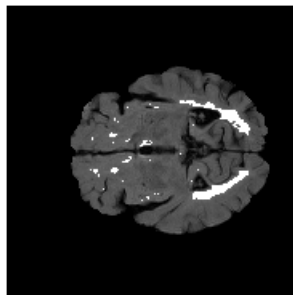
Bilde 15 i pasientvolum 20



Bilde 15 i pasientvolum 20, med fasit



Bilde 15 i pasientvolum 20 med resultater



Figur 4.5. Figuren viser bilde 15 fra pasientvolum 20, tabell 6. De tre bildene er henholdsvis det originale bildet, det originale bildet med fasit overlatt og det originale bildet med resultater overlatt

Gjennomsnittlig sett er altså ikke resultatene spesielt gode, men for enkelte pasientvolum gis gode resultater. For andre volum, som vist i *figur 4.3*, vil det være enkelt å utelukke store deler av resultatene, men bare etter å manuelt ha sett gjennom bildene. Det må også bemerkes at spesifisitetsverdiene kan virke kunstig høye, da lesjonene som detekteres er av såpass mye mindre størrelse enn resten av hjernemassen. De falske positive verdiene får derfor ikke så mye og si totalt sett.

5 Konklusjon og videre arbeid.

Innledningsvis i rapporten ble det nevnt at hensikten med arbeidet som har blitt gjort var å produsere en algoritme for automatisk deteksjon av lesjoner i MR-bilder av hjernen til demenspasienter. Etter å ha gjennomgått resultatene i kapittel 4.3 ser en at målet bare til en viss grad har blitt nådd. FDR-verdiene som fås er gjennomsnittlig uakseptabelt høye, men for enkelte pasientvolum er resultatene gode.

Som tidligere nevnt ble det i litteraturstudiet ikke funnet artikler om bruk av watershedsegmentering for å detektere lesjoner i MR-bilder av hjernen, noe som gjorde dette til en veldig spennende oppgave å jobbe med. Resultatene gir dog en indikasjon på at gradientmetoder som watershed kan være vanskelige å bruke på MRI-data; men det finnes forbedringspotensiale.

Forslag til videre arbeid kan blant annet være å få *fmofr*-algoritmen til å også fjerne resultater sentrert i bildet som en vet ikke er lesjoner, som vist i *figur 4.3*. Det kan også være fordelaktig å bruke en tersklingsverdi per pasientvolum, i stedet for å sette en ny tersklingsverdi for hvert enkelt bilde i alle pasientvolum. Videre kan 3-D-informasjonen som finnes i bildene utnyttes. Ved å sammenligne resultatene for et enkeltbilde med bildene over og under i samme pasientvolum vil feildetekterte resultater forhåpentligvis kunne fjernes.

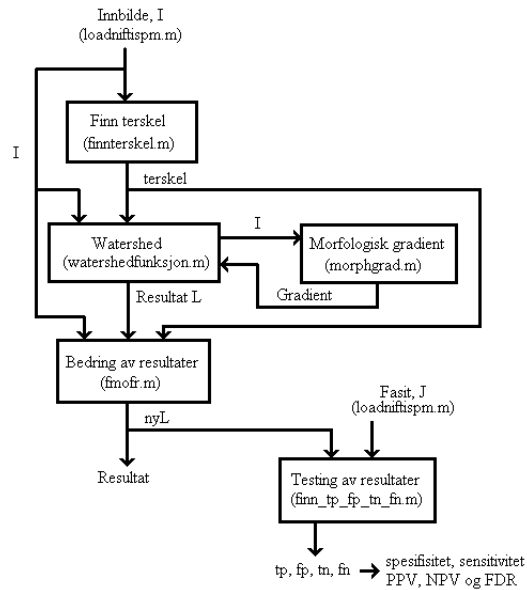
Referanser

- [1] R. C. Gonzalez, R. E. Woods og S. L. Eddins. Digital Image Processing Using MATLAB. Gatesmark Publishing. ISBN 9780982085400
- [2] E. R. Dougherty, R. A. Lotufo. Hands-on Morphological Image Processing. SPIE, 2003 ISBN: 978-0-8194-4720-3.
- [3] <http://www.mathworks.com/help/toolbox/images/morph5.gif>
- [4] Digabel, H., and Lantuejoul, C. Iterative Algorithms, Actes du Second Symposium Europeen d'Analyse Quantitative des Microstructures en Sciences des Materiaux, Biologie et Medecine, Caen, 4-7 1977, J.-L. Chermant, Ed., Riederer Forlag, Stuttgart, 1978.
- [5] Lantuejoul, C. La Squelettisation et Son Application Aux Mesures Topologiques Des Mosaïques Polycristallines. PhD thesis, Ecole des Mines, Paris, 1978.
- [6] <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexw>
- [7] http://www.nasjonalforeningen.no/no/Demens/Om_demens/
- [8] Linda G. Shapiro and George C. Stockman (2001): "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3
- [9] Beucher, S. "The Watershed Transform Applied to Image Segmentation", Proceedings of the Pfefferkorn Conference on Signal and Image Processing in Microscopy and Microanalysis, pp. 299-314, September 1991.
- [10] Lobo A, Launer LJ, Fratiglioni L, et al. Prevalence of dementia and major subtypes in Europe: a collaborative study of population-based cohorts. *Neurology* 2000;54:S4-S9.
- [11] Clarke, L.P.; Velthuizen, R.P.; Camacho, M.A.; Heine, J.J.; Vaidyanathan, M.; Hall, L.O.; Thatcher, R.W.; Silbiger, M.L. MRI segmentation Methods and applications. *Magn. Reson. Imaging* 13:343-368; 1995.
- [12] Rolf A. Heckemann, Joseph V. Hajnal, Paul Aljabar, Daniel Rueckert, and Alexander Hammersc, Automatic anatomical brain MRI segmentation combining label propagation and decision fusion, 2005
- [13] M. Garbey; B.L. Bass; C Collet et al. Computational Surgery and Dual Training, Springer-Verlag New York Inc.2010. ISBN 9781441911223
- [14] J. Sijbers et al. "Watershed-based segmentation of 3D MR data for volume quantization", 1997, Elsevier Science Inc.
- [15] M.W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, S.M. Smith. Bayesian analysis of neuroimaging data in FSL. *NeuroImage*, 45:S173-186, 2009.

- [16] S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23(S1):208-219, 2004.
- [17] S.M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143-155, November 2002.
- [18] Altman DG, Bland JM (1994). "Diagnostic tests. 1: Sensitivity and specificity", <http://www.bmj.com/content/308/6943/1552.full>
- [19] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.
- [20] Petronella Anbeek, Koen L. Vincken, et al. Probabilistic segmentation of white matter lesions in MR imaging , Utrecht 2003.
- [21] <http://www.fil.ion.ucl.ac.uk/spm/software/>

A MATLAB-kode

I denne delen av appendiket vises matlabfunksjonene som har blitt brukt. Grunnet formateringen i LyX gjengis funksjonene som bilder. Flytskjemaet fra kapittel 3 gjengis i *figur A.1* under, da dette også viser bruken av matlabfunksjonene i parentes.



Figur A.1. Figuren viser en flytskjemaoversikt av algoritmen. Matlabfunksjonene som blir brukt er gitt i parentes.

A.1 Kjøring av algoritmen

Denne matlabkoden blir brukt for å kjøre igjennom alle pasientvolumene. Den er kort fortalt en mer nøyaktig versjon av algoritmeoversikten som vises i kapittel 3.2.

```
1
2 - gruppe1 = [2,11,13,16,20,21,25,26,27,34,35,44,48,51,52,58,61,62,63,65,66,68
3 - gruppe2 = [1,3,5,7,14,15,19,22,23,24,28,29,30,32,36,47,49,50,54,55,56,59,60
4 - gruppe3 = [4,6,8,10,12,17,18,31,33,37,38,41,43,46,53,57,70,72,73,79,80,86,88
5
6 - totalspes = 0;
7 - totalsens = 0;
8 - for n = gruppe3; %for n = 1:102
9   % Generer filnavn og last inn bildet
10 -  fname = sprintf('%i_FLAIR_brain.nii', n);
11 -  x = loadniftisp(m(fname));
12
13   % Last inn fasit
14 -  fname = sprintf('%i_wmws.nii', n);
15 -  y = loadniftisp(m(fname));
16
17 -  v=size(x);
18 -  for i=1:v(3)           % for alle bilder i et volum
19
20 -    I{i} = (x(:,:,i)/(max(x(:)))));
21 -    J{i} = (y(:,:,i)/(max(y(:)))));
22
23 -    terskel{i} = finnterskel(I{i});
24 -    grense{i} = terskel{i} + 0.06;      %
25
26
27 -    [ L{i} bw{i} fgm{i} ] = watershedfunksjon( I{i}, grense{i} );
28 -    % Har bw og fgm som output her. Blir ikke brukt, men fint å sjekke..
29
30 -  end
31
32
33 -  for j=1:v(3)
34
35 -    [nyL{j}] = fnofr( L{j}, I{j}, terskel{j} );      % evt. grense{j}
36 -                                                    % dersom grense/4
37 -  end
```

```

38
39 -   tp=0;
40 -   fp=0;
41 -   tn=0;
42 -   fn=0;
43
44 -   for m=1:v(3)
45
46 -       [ny_tp ny_fp ny_tn ny_fn] = finn_tp_fp_tn_fn(nyL{m}, I{m}, J{m});
47
48 -       tp = ny_tp + tp;
49 -       fp = ny_fp + fp;
50 -       tn = ny_tn + tn;
51 -       fn = ny_fn + fn;
52
53 -   end
54
55 -   tp = tp / v(3);
56 -   fp = fp / v(3);
57 -   tn = tn / v(3);
58 -   fn = fn / v(3);
59
60 -   sens = tp / (tp + fn);
61 -   spes = tn / (fp + tn);
62 -   ppv = tp / (tp + fp);
63 -   npv = tn / (tn + fn);
64 -   fdr = fp / (fp + tp);
65
66 -   totalsens = sens+totalsens;
67 -   totalspes = spes+totalspes;
68
69 -   filename = sprintf('Resultater/Gruppe1/volum_%i_006_finnterskel.mat_', n);
70 -   save(filename);
71
72 - end

```

A.2 loadniftispm.m

loadniftispm.m er funksjonen som brukes for å laste inn MR-bildene i matlab. En er avhengig av å ha software-pakken til SPM[21] i matlab-stien for at denne skal fungere. Denne m-filen ble sendt til meg av Ketil Oppedal.

```
1 function mimage = loadniftispm(fname)
2
3     imhan = spm_vol_nifti(fname);
4     mimage = zeros(imhan.dim);
5     for i = 1 : imhan.dim(3)
6         imsli = spm_slice_vol(imhan,spm_matrix([0 0 i]),imhan.dim(1:2),0);
7         mimage(:,:,i) = imsli;
8     end
9     %[mimage, loc] = spm_read_vols(struct);
10
```

A.3 finnterskel.m

finnterskel.m finner tersklingsverdien for forgrunnsmarkørene ved hjelp av histogrammet til inngangsbildet.

```
1 function [ thresh ] = finnterskel( x )
2 %FINNTERSKEK Summary of this function goes here
3 % Detailed explanation goes here
4     [counts,q] = imhist(x(:));
5     qq = smooth(counts(2:end));
6     dqq = diff(qq);
7     [val idx] = min(dqq);
8     %display(idx)
9     %plot(dqq);
10    for i = idx:length(dqq)
11        if abs(dqq(i)) < 100
12            thresh = q(i);
13            %display(thresh);
14            break;
15        end
16    end
17
18 end
19
20
```

A.4 watershedfunksjon.m

```
1 function [ L bw7 fgm3 ] = watershedfunksjon( I, grense )
2 %WATERSHEDFUNKSJON Summary of this function goes here
3 % Detailed explanation goes here
4
5 % Morfologisk gradient.
6 gradmag = morphgrad(I);
7
8 % Forgrunnsmarkører
9 K = I>grense;
10 K2 = bwmorph(K,'bridge');
11 K3 = bwmorph(K2,'clean');
12 fgm = K3;
13
14
15 % Strukturelementer
16 se = strel('square', 7);
17 se2 = strel('square', 3);
18
19 % Glatting av bilde
20 Ier = imerode(I, se);
21 Iorec = imreconstruct(Ier, I);
22 Idil = indilate(Iorec, se);
23 Iocrec = imreconstruct(imcomplement(Idil), imcomplement(Iorec));
24 Iocrec = imcomplement(Iocrec);
25
26 bw = im2bw(Iocrec, graythresh(Iocrec));
27
28
29 % fjerning av ytterpunkter rundt hjernen
30 fill = imfill(bw, 'holes');
31 fill2 = imerode(fill, se);
32 fill3 = imcomplement(fill2);
33 fill4=fill.*fill3;
34
35 %figure, imshow(fill4)      gradient rundt hjernen, trekkes fra fgm for å
36 %                          fjerne ytterpunkter
37 fgm2 = fgm-fill4;
38 fgm3 =fgm2>0;
39
40 bw = bw-fgm2;
41 bw2 = imerode(bw, se2);
42
43 p = 50; %størrelse på elementer som fjernes fra bw
44 bw3 = bwareaopen(bw2, p);
45 bw4 = bw3 + fgm3;
46 bw5=imcomplement(bw4);
47 bw6=imerode(bw5, se2);
48 bw7 = bw4+bw6;
49
50
51 %wr1 - watershed ridge lines
52 D = bwdist(bw7);
53 DL = watershed(D);
54 wr1 = DL == 0;
55
56 wrlinv = imcomplement(wr1);
57 gradmag2 = gradmag+wrlinv;
58 gradmag3 = imimposemin(gradmag2, wrlinv | fgm);
59
60 L = watershed(gradmag3);
61
62 end
```

A.5 morphgrad.m

Denne funksjonen finner den morfologiske gradienten for hvert bilde.

```
1 function [ gradmag ] = morphgrad( I )
2 %MORPHGRAD Summary of this function goes here
3 % Finner morfologisk gradient
4 - se_morph = strel('square', 3);
5 - Jdil = imdilate(I, se_morph);
6 - Jero = imerode(I, se_morph);
7 - gradmag = Jdil-Jero;
8 - end
```

A.6 fmofr.m

fmofr.m er funksjonen som brukes for å bedre resultatene etter watershedsegmenteringen. Hvert enkelt segmenterte område sjekkes opp mot originalbildet I . Hvis det viser seg at et resultatsegment har blitt funnet der gjennomsnittsverdien av tilsvarende område i I har en gjennomsnittlig verdi lavere enn en valgt grense, forkastes dette resultatet, og L oppdateres.

```
1 function [ nyL ] = fmofr( L, I, grense )
2 %FMOFR Summary of this function goes here
3 % Fjerner Mørke Områder Fra Resultater
4
5 - nyLtmp = zeros(size(L),'double');
6 - for k=2:max(max(L))
7 -     omr = (L==k).*I;
8 -     avgverdi = (sum(omr(:)))/(sum(sum(omr>0)));
9 -     if avgverdi < (grense) %endres til (grense/4)
10 -         tmp = ((L==k)*0);
11 -     else
12 -         tmp = (L==k);
13 -     end
14 -     nyLtmp = nyLtmp + tmp;
15 - end
16
17 - nyL = nyLtmp;
18 - end
19
20
```

A.7 finn_tp_fp_tn_fn.m

finn_tp_fp_tn_fn.m finner, som funksjonsnavnet sier, tp, fp, tn og fn. Disse brukes til å regne ut de fem verdiene resultatene blir bedømt etter, spesifisitet, sensitivitet, PPV, NPV og FDR. Inni denne funksjonen finnes også en invers av fasit som kun tar hensyn til selve hjerneområdet, ikke de sorte områdene i bakgrunnen.

```
1  function [ tp fp tn fn ] = finn_tp_fp_tn_fn( resultat, I, fasit )
2  %FINN_TP_FP_TN_FN Summary of this function goes here
3  % Finner tp (true positive), fp (false positive), tn (true negative) og
4  % fn (false negative) som blant annet brukes til å finne verdier som
5  % sensitivitet og spesifisitet
6
7
8  -   se = strel('square', 3);
9  -   bw = im2bw(I, graythresh(I));
10 -   fill = imfill(bw, 'holes');
11 -   filled = imdilate(fill, se);
12 -   filled = imfill(filled, 'holes');
13 -   invfilled = inplement(filled);
14 -   invfasit = ((1-fasit)-invfilled);
15
16
17 -   tmp = fasit(:).*(resultat(:));
18 -   tp = sum(tmp(:));
19
20 -   tmp = (invfasit(:)).*(1-resultat(:));
21 -   tn = sum(tmp(:));
22
23 -   tmp = (invfasit(:)).*(resultat(:));
24 -   fp = sum(tmp(:));
25
26 -   tmp = (fasit(:)).*(1-resultat(:));
27 -   fn = sum(tmp(:));
28
29
30 -   end
31
```


A.8 finn_grupper.m

finn_grupper.m er en matlabfunksjon skrevet av Kjell Kristian Grane Torgersen, som også har arbeidet med det samme MRI-materialet. Funksjonen deler pasientvolumene inn i tre grupper. Dette ble gjort for å få et treningssett, et verifiseringssett og et backup-sett, som var like for begge to.

```
1 function [gruppe gruppe1 gruppe2 gruppe3] = finn_grupper()
2 - load('datainfo.mat');
3 - datagrupper = [newfncell{2:end,10}];
4 - bildenr = [newfncell{2:end,1}];
5 - bilde = zeros(length(datagrupper), 1);
6 - for i = 1:length(datagrupper)
7 -     bilde(bildenr(i)) = datagrupper(i);
8 - end
9
10 %for j=1:3
11 %for i=(4*j+0:3)
12 %     gruppe(j,:) = [find(bilde==i)];
13 %end
14 %end
15
16 - for i=0:11
17 -     gruppe{i+1} = find(bilde==i);
18 - end
19 - gruppe1 = [];
20 - gruppe2 = [];
21 - gruppe3 = [];
22
23 % Gruppe 2,3,4 og 5 har ett mer volum enn resten av gruppene.
24 - for i=[2 6 7 8] % 33 volum
25 -     gruppe1 = [gruppe1; gruppe{i}];
26 - end
27 - for i=[3 9 10 11] % 33 volum
28 -     gruppe2 = [gruppe2; gruppe{i}];
29 - end
30 - for i=[4 5 12] % 26 volum
31 -     gruppe3 = [gruppe3; gruppe{i}];
32 - end
33
34 % skal ikke være 0
35 - unike = unique(diff(sort([gruppe1; gruppe2; gruppe3])));
36
37 - i=1;
```