

Subfaculteit wetenschappen



Probleemoplossen en ontwerpen 3

Biologische Data Analyse App

**Marthe Böting
Robin Bruneel
Toon Ingelaere**

Titularis : Koen Van Den Abeele

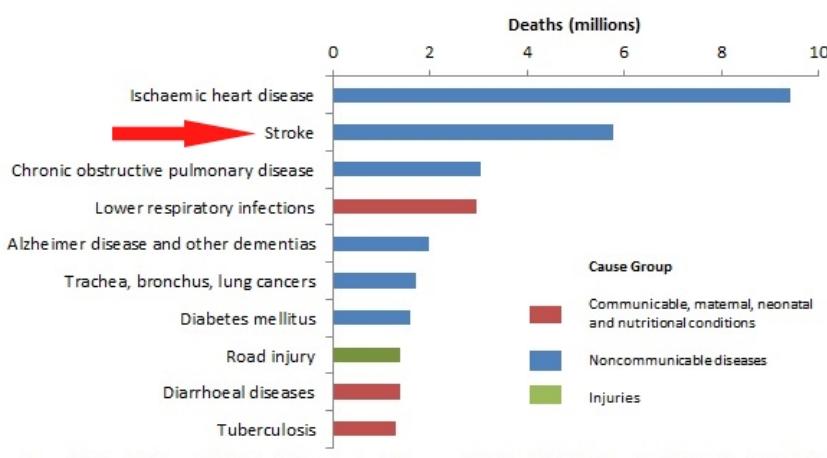
Begeleider : Senna Staessens

Academiejaar 2018 - 2019

Inleiding

Beroertes zijn in onze westerse samenleving de derde grootste doodsoorzaak na hartinfarcten en kanker. Zoals we zien op Figuur ?? kapen ze op wereldvlak zelfs de tweede plaats weg [?]. In België komen er gemiddeld 25 000 beroertes per jaar voor. In 15% van deze gevallen overlijdt de patiënt. De overige 85% heeft na een beroerte vaak last van blijvende functiebeperkingen zoals cognitieve-, emotionele of gedragsproblemen. Een ischemische beroerte ontstaat doordat een bloedklonter emboliseert en in één van de hersenbloedvaten vast komt te zitten. Deze bloedklonter moet verwijderd worden of de patiënt loopt een hersenschade op. Momenteel focust de huidige therapie zich op snel en efficiënt verwijderen van de bloedklonter. In eerste instantie kan men via een geneesmiddel, weefsel plasminogen activator, proberen om de klonter op te lossen. Dit geneesmiddel moet gegeven worden binnen de eerste 4,5 uur na het optreden van de symptomen. Wanneer dit geneesmiddel wordt toegediend na dit tijdstip kan dit leiden tot bloedingen of toxiciteit in de hersenen. Van alle patiënten die het geneesmiddel toegediend krijgen, lost de klonter slechts in 1/3 van de patiënten op. Op het moment dat de klonter niet oplost, maakt men gebruik van trombectomie om de klonter er manueel uit te halen. Om huidige therapeutische opties te verbeteren en om het aantal slachtoffers aan beroertes te doen slinken, gebruikt men deze bloedklonters om verder onderzoek op uit te voeren. Hiervoor gaan ze de samenstelling van de bloedklonter proberen te analyseren. Dit gebeurt op basis van afbeeldingen die gemaakt zijn van de bloedklonter waarbij welbepaalde componenten gekleurd zijn (bijvoorbeeld: rode bloedcellen, witte bloedcellen en bloedplaatjes) en vervolgens geanalyseerd worden via kleur-gebaseerde segmentatie analyse. Deze analyses zijn echter erg tijdverzijdend. Er is ons dan ook gevraagd om een gebruiksvriendelijke app te ontwikkelen die de analyse van de afbeeldingen kan automatiseren. In dit verslag gaan we eerst in op wat de klant specifiek van ons verwacht en aan welke specificaties ons ontwerp moet voldoen. Hierna bespreken we ons design en lichten we het toe. Verder bespreken we ook enkele van onze voorlopige resultaten. Ten slotte wordt er nog een blik geworpen naar de vakken uit eerste drie semesters die ons hierbij geholpen hebben.

Top 10 global causes of deaths, 2016



Source: Global Health Estimates 2016: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2016. Geneva, World Health Organization; 2018.

Figuur 1: Statistieken van de *World Health Organization* van 2016 waarin te zien is dat wereldwijd beroertes (*strokes*) de tweede meest frequente doodsoorzaak is.



Inhoudsopgave

1 Klantenvereisten

De klant verwacht een gebruiksvriendelijke app die de afbeeldingen automatisch verwerkt. Dit houdt in dat een afbeelding ingeladen kan worden, de foto bijgesneden en de achtergrond verwijderd wordt. Daarnaast is het de bedoeling om de samenstelling van de bloedklonter te analyseren aan de hand van de aangebrachte indicator.

2 Ontwerpspecificatie

De klant wil dat een afbeelding van een bloedklonter automatisch bewerkt en daarna geanalyseerd wordt. Het bewerken van de afbeelding houdt twee dingen in. Eerst en vooral moet de afbeelding zodanig bijgesneden worden dat de volledige bloedklonter erop staat. Hierbij mogen we de randen echter niet te breed nemen, aangezien er dan nuttige geheugenruimte¹ verspild wordt. Naast het bijsnijden, moet ook de achtergrond verwijderd worden. Dit betekent dat alle pixels die niet tot de bloedklonter behoren wit gekleurd worden. Indien dit niet goed gebeurt, kunnen de resultaten van de kleurenanalyse namelijk vertekend zijn.

In de kleurenanalyse moet het percentage van met indicator gekleurde pixels geteld worden. Voor dit project moeten we slechts twee soorten kleuringen analyseren. Een voorbeeld van deze is te zien in Figuur ???. In beide gevallen moet de app op een accurate manier onderscheid kunnen maken tussen de eiwitten die gedetecteerd moeten worden en de rest van de bloedklonter.

Dit alles moet samengegoten worden in een visuele en gebruiksvriendelijke app. Dit wil zeggen dat de app makkelijk te installeren en te gebruiken is. De gebruiker moet ook een overzicht van de verschillende afbeeldingen van de bloedklonter kunnen terugroepen. Dit overzicht bestaat uit de originele afbeelding, de afbeelding zonder achtergrond en de afbeelding waarbij de indicatorpixels zijn aangeduid. Zo kunnen mogelijke fouten snel gedetecteerd worden. Daarnaast moet deze app nog wat extra functionaliteiten uitvoeren zoals het opslaan van deze afbeeldingen, het manueel bijwerken van het bekomen resultaat, het wegschrijven van alle data in een csv² bestand, etc.

3 Onze oplossing

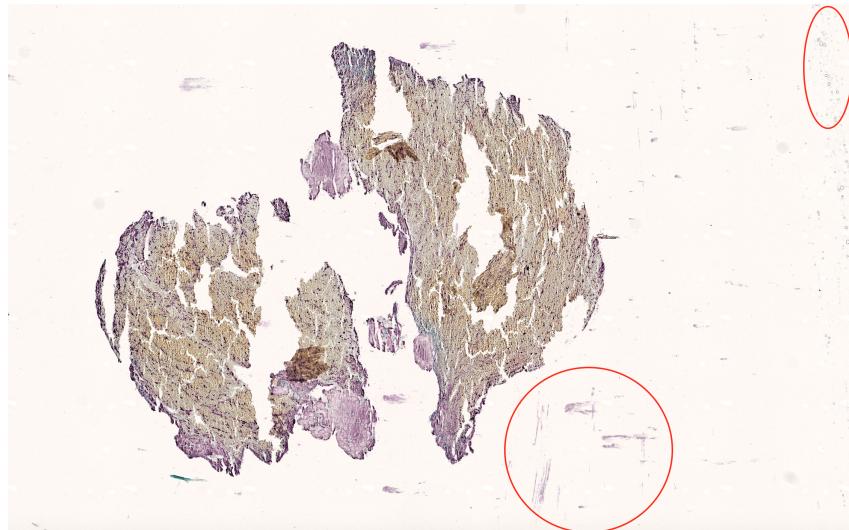
Dit hoofdstuk hebben we opgedeeld in verschillende deelproblemen, namelijk het verwijderen van de achtergrond, het lokaliseren van de indicator en de gebruiksvriendelijke app. We zullen deze deelproblemen nader bespreken en onze bekomen bevindingen rapporteren. Om onze app te realiseren, maken we gebruik van de programmeertaal MATLAB.

3.1 Achtergrondverwijdering

Het eerste deelprobleem van ons project is het verwijderen van de achtergrond. Op de afbeeldingen is er heel wat vuilheid te vinden. Voorbeelden hiervan zijn luchtbellen of kleine verkleuringen naast de bloedklonter zoals men ziet op de Figuur ???. Hieronder zullen we verschillende operaties beschrijven om de grootste kloners te lokaliseren en alles wat geen klonter is te verwijderen. Dit leidt tot een ruisvrije afbeelding.

¹We werken namelijk met foto's van de orde van 200MB, het sparen van pixels op ons resultaat is uiterst voordelig

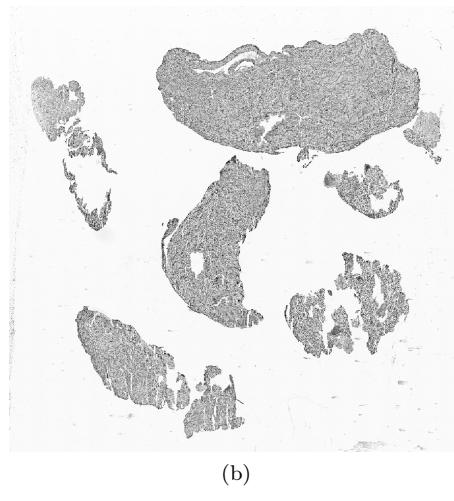
²Een csv (Comma Separated Values) bestand is een tekstbestand die als tabel ingelezen kan worden. Het kan eenvoudig in excel geopend worden voor verdere analyse.



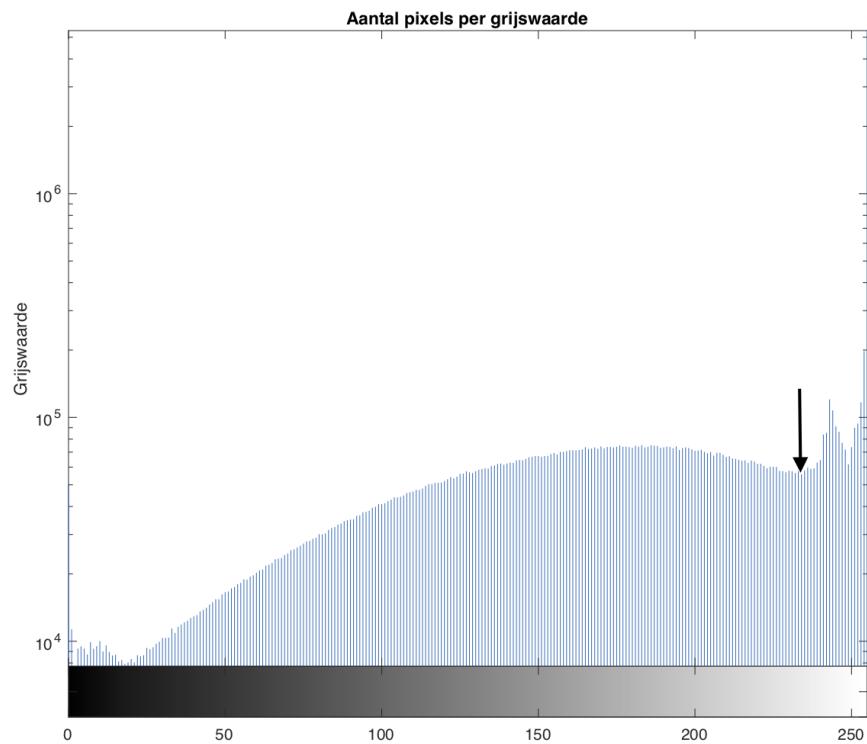
Figuur 2: In de rechterbovenhoek zien we luchtbellen en in de onderste cirkel zien we een verkleuring die geen deel uitmaakt van een bloedklonter.

3.1.1 Bepalen van de beste threshold

Wanneer we de grijswaarden van de afbeelding berekenen, zien we een opmerkelijk verschil tussen de achtergrond (eerder wit) en de bloedklonter (eerder grijs), zie Figuur ???. Wanneer we een histogram van deze grijswaarden opstellen, vinden we ook een lokaal minimum tussen het grijs en het wit. Dit zien we duidelijk op het histogram in Figuur ???. Wanneer we dit minimum berekenen, vinden we de theoretisch optimale threshold, de grijswaarde om een bloedklonterpixel van een achtergrondpixel te onderscheiden. Wanneer we die threshold toepassen en een binaire representatie zoals in Figuur ?? vormen, zien we namelijk dat dit inderdaad een goede threshold is.



Figuur 3: Illustratie van de originele foto (a) en deze omgezet in grijswaarden (b).



Figuur 4: Histogram van het aantal pixels gegroepeerd per grijswaarde. We zien duidelijk een lokaal minimum rond de waarde 230. Opmerking: we maken hier gebruik van een logaritmische y-as.



Figuur 5: Op deze binaire representatie zijn de bloedklonters duidelijk te zien.

3.2 De gebruiksvriendelijke applicatie

De app werkt op basis van de klant die een foto kan ingegeven. Het resultaat is een bewerkte foto waarbij er een percentage van de hoeveelheid aanwezig kleur weergegeven wordt.

Op het moment dat de klant een afbeelding gekozen heeft, zal in de eerste fase de achtergrond van de foto wit worden gemaakt en wordt de foto bijgesneden zodanig dat enkel de bloedklonter zichtbaar is. Als dit gebeurd is, zal er al een eerste resultaat getoond worden. Hierdoor heeft de klant een beeld waarop de verdere analyse zal gebeuren en kan die eventueel ingrijpen bij fouten.

Vervolgens zal de kleurenanalyse gebeuren. Ook hiervoor zal de foto getoond worden waarop de klant kan zien welke delen de indicator bevatten en zal het percentage gegeven worden.

Soms is de kleuring intenser dan anders dit is het gevolg van verschillende factoren (bijvoorbeeld: temperatuur, druk, vochtigheid,...). Hierdoor kan het resultaat die wordt bekomen via de app eens afwijken van de werkelijke waarde. Daarom hebben wij sliders toegevoegd. Zoals eerder vermeld hebben wij experimentele waarden bepaald waartussen gewerkt wordt. Door het toevoegen van deze sliders kan de klant zelf deze waarden aanpassen wanneer de kleuring afwijkt. Deze kunnen ook gebruikt worden op het moment dat de bewerkte foto . Daarnaast is er de mogelijkheid aan om niet foto per foto in te geven maar een volledige map met foto's die dan elk afzonderlijk bewerkt en opgeslagen worden.

Besluit

Afsluitende tekst

	Week 1							Week 2							Week 3							Week 4							Week 5							Week 6						
	Sep							Oct							Nov																											
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	S	S	M	T	W	F	S	S	M	T	W	T	F	S	S	S									
24	25	26	27	28	29	30	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	01	02	03	04	

T.T. verslag

The figure is a Gantt chart illustrating a project timeline from November 5 to December 23. The x-axis represents time, with major ticks every two days. The y-axis lists tasks, each represented by a blue bar indicating its duration and timing.

Tasks and Durations:

- Task 1:** Starts on Nov 5, ends on Nov 11, duration 7 days.
- Task 2:** Starts on Nov 11, ends on Nov 18, duration 8 days.
- Task 3:** Starts on Nov 18, ends on Nov 25, duration 8 days.
- Task 4:** Starts on Nov 25, ends on Dec 01, duration 7 days.
- Task 5:** Starts on Dec 01, ends on Dec 08, duration 8 days.
- Task 6:** Starts on Dec 08, ends on Dec 15, duration 8 days.
- Task 7:** Starts on Dec 15, ends on Dec 22, duration 8 days.
- Task 8:** Starts on Dec 22, ends on Dec 23, duration 1 day.

Key Dates and Annotations:

- A thick black arrow points from the start of Task 1 (Nov 5) to the end of Task 2 (Nov 18).
- Two black diamonds are placed on the timeline: one at the end of Task 1 (Nov 11) and another at the end of Task 2 (Nov 18).

3

4

Eindverslag

Eindpres.

Taaknummer	Taakomschrijving
1	Achtergrond verwijderen en afbeelding bijsnijden
2	Kleuren detecteren en kwantificeren
3	Ontwerpen van een gebruiksvriendelijke app
4	Optimalisatie van het algoritme

Table 1: Omschrijvingen van de taken in de gantt-chart