

1. Meetrapport Edge Detection

1.1. Namen en datum

Bart van Netburg en Marty Vos, 13-04-2019

1.2. Doel

Wij willen onderzoeken of onze eigen implementatie van een edge detection algoritme 'beter' werkt dan de default edge detection. Met beter bedoelen wij dat de edges duidelijk zichtbaar zijn en dat de volgende stappen van de gezichtsherkenning goed verlopen. Voor het detecteren van edges, gebruiken wij eerst een Gaussian filter, daarna een Laplacian filter en tot slot gebruiken wij thresholding om een monochrome afbeelding te genereren.

1.3. Hypothese

Wij verwachten dat de resultaten van onze implementatie duidelijkere edges opleveren en dat de resultaten van de volgende stappen in de gezichtsherkenning beter zijn.

1.4. Werkwijze

De duidelijkheid van de edges is afhankelijk van de gebruikte filters en de thresholding waarde. Daarom voeren wij meerdere experimenten uit: wij variëren eerst de thresholding waarde en veranderen de gebruikte filters niet. Na elk experiment wordt het programma opnieuw opgestart. Wij testen met elke testfoto.

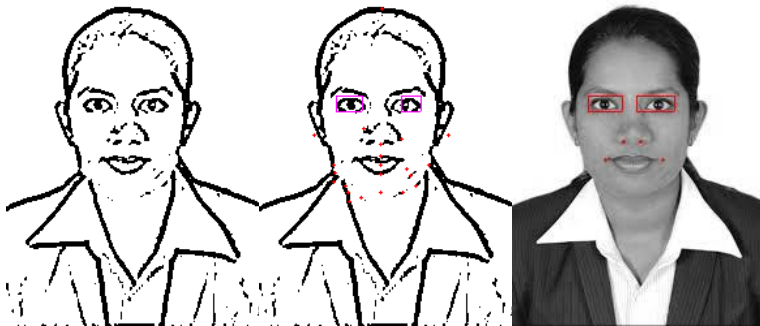
1.5. Resultaten

In de tabellen hieronder zijn de edges van de foto's te zien, samen met de features en de uiteindelijke foto. Bij bepaalde gevallen ontbreekt een uiteindelijke foto. Dit wordt later bij de conclusies behandeld. In figuur 1 t/m 7 is de uitvoer van default edge detection te zien, samen met de features en de uiteindelijke foto's.

De resultaten zijn in *Tabel 1* uitgezet. Dit tabel is vanaf pagina 4 te zien.



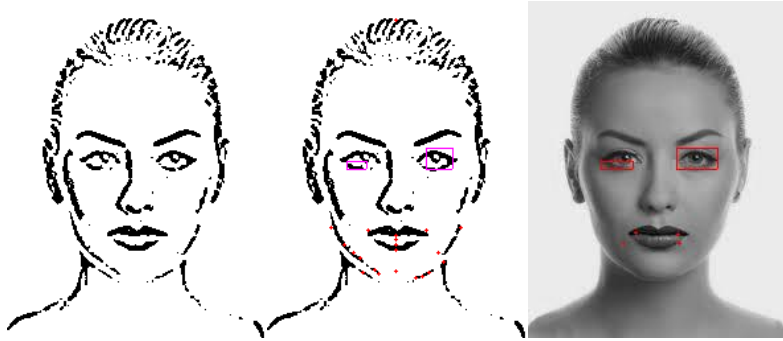
Figuur 1 Uitvoer van de default edge detection van child-1.png, met features en de uiteindelijke foto.



Figuur 2 Uitvoer van de default edge detection van female-1.png, met features en de uiteindelijke foto.



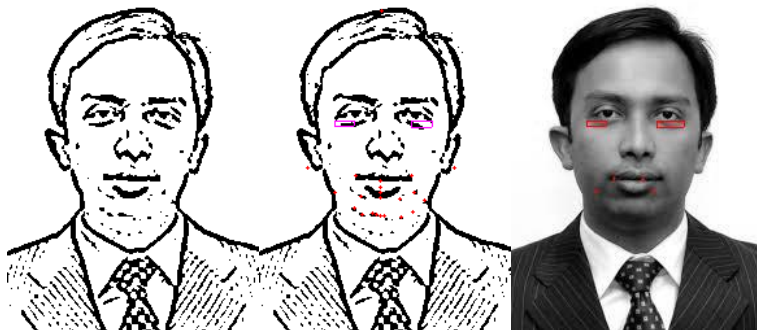
Figuur 3 Uitvoer van de default edge detection van female-2.png, met features en de uiteindelijke foto.



Figuur 4 Uitvoer van de default edge detection van female-3.png, met features en de uiteindelijke foto.



Figuur 5 Uitvoer van de default edge detection van male-1.png, met features en de uiteindelijke foto.





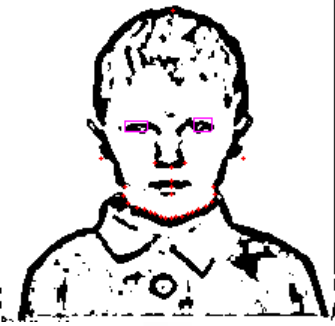


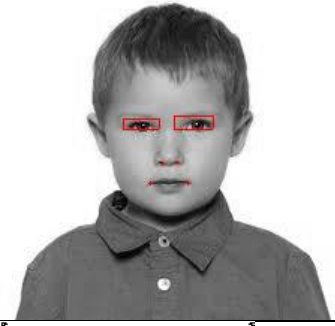
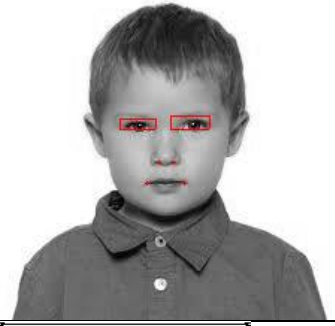





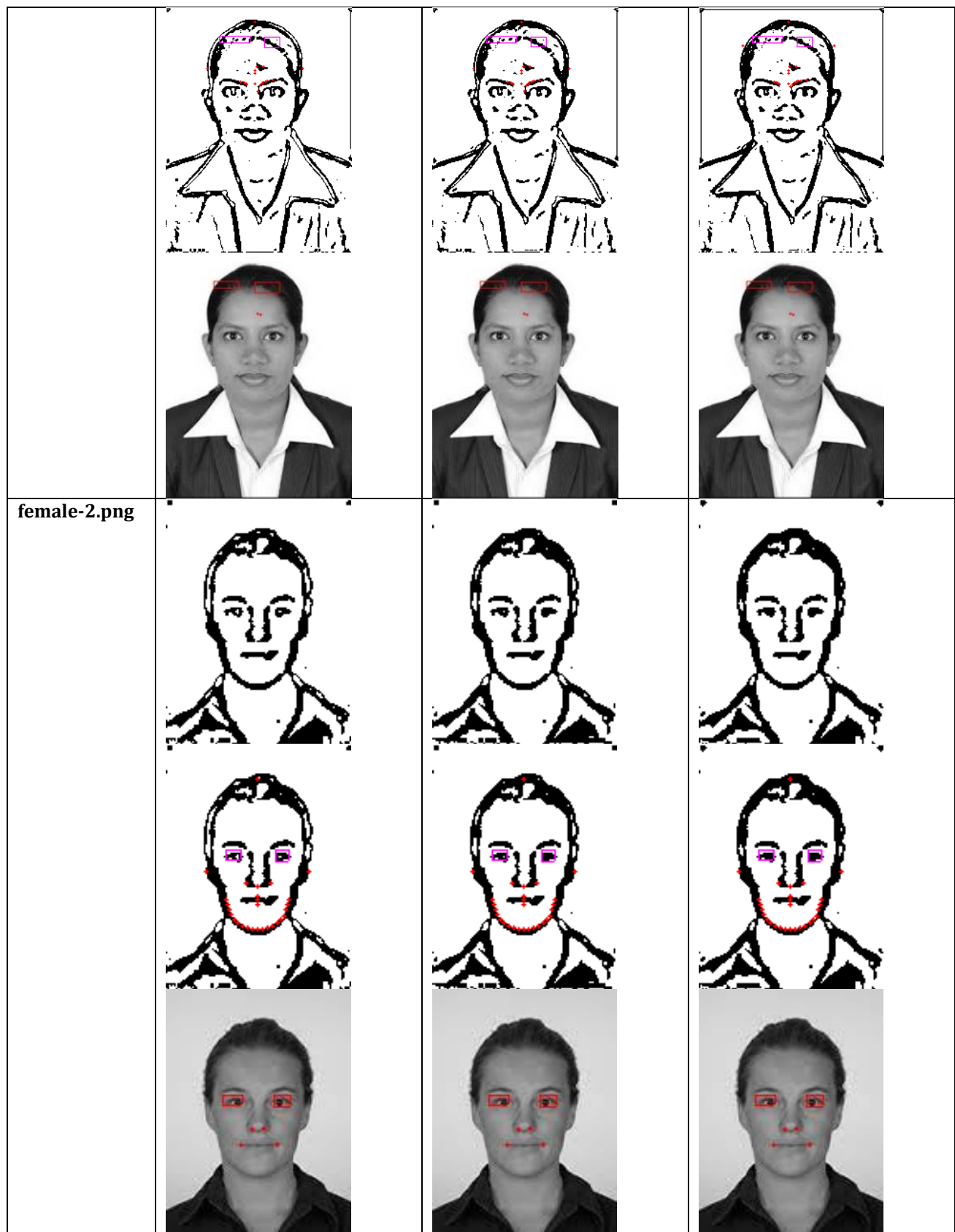
Figuur 6 Uitvoer van de default edge detection van male-2.png, met features en de uiteindelijke foto.

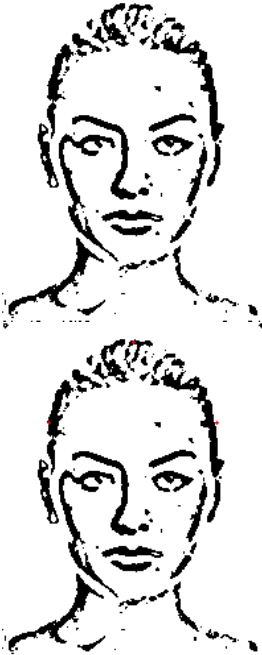
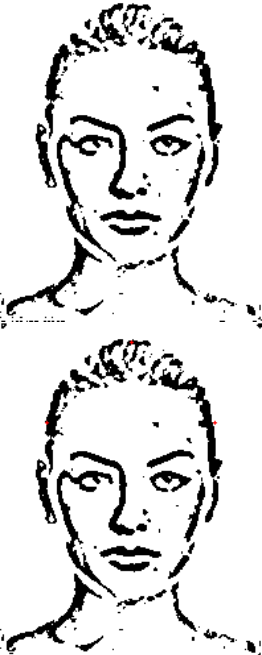
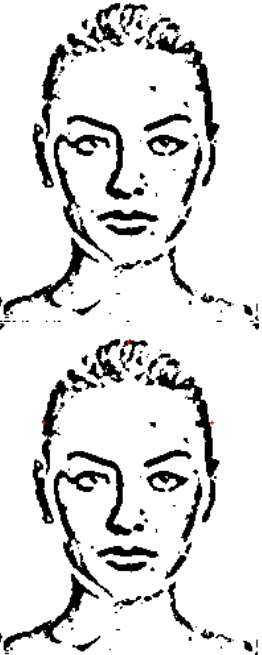
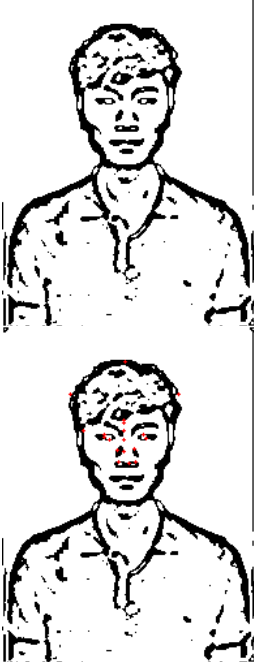
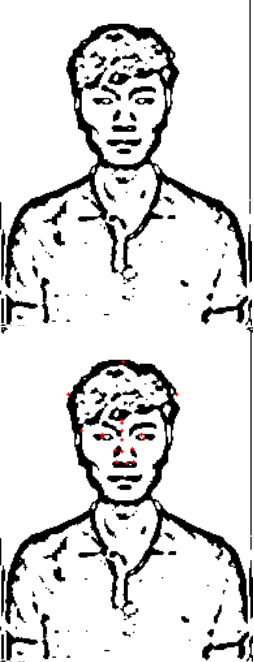
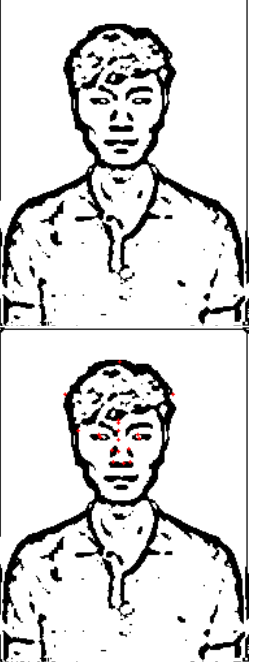









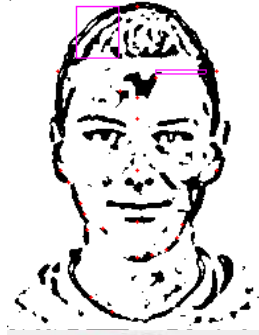


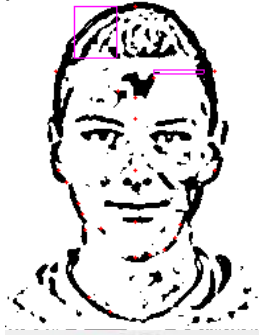



Figuur 7 Uitvoer van de default edge detection van male-3.png, met features en de uiteindelijke foto.

Tabel 1 Laplacian Filter

Tresholding waarde	80	90	100
child-1.png			
			
			
female-1.png			



female-3.png			
male-1.png			
male-2.png			

			
male-3.png	  	  	 

1.6. Verwerking

Bij een thresholding waarde van 100, worden de features minder goed herkend. Bij *male-2.png* en *male-3.png* is dat goed te zien. Echter worden de features van *male-3.png* slecht of verkeerd herkend en er is ook geen uiteindelijke foto gegenereerd. Maar bij een thresholding waarde van 80 worden de features het beste herkend en zijn de edges ook het duidelijkst, zie *male-1.png*. Ook al heeft deze geen uiteindelijke foto, er zijn wel features te zien.

1.7. Conclusie

Aan de hand van de tabellen is te zien dat bij een lagere thresholding waarde er ‘betere’ resultaten opgeleverd worden. Echter waren de eindresultaten niet goed: de features stonden bij bepaalde gevallen niet op de goede plek en de edges waren te dik, wat er waarschijnlijk voor zorgde dat de features verkeerd berekend werden. Dus deze implementatie is niet beter dan de default implementatie.

1.8. Evaluatie

Zoals het al in *Conclusie* al verteld is, is onze implementatie van edge detection niet beter dan de default implementatie. Het onderzoek zou beter kunnen gaan als wij meerdere thresholding waarden gebruikten. Nu hebben wij waarden gebruikt die dichtbij elkaar liggen. Ook konden wij een ander filter gebruiken in plaats van de Laplacian of deze aanpassen.