

Meetrapport

Betrouwbaarheid

Jeroen

Stefan

22-03-2017

Inhoudsopgave

Doel	3
Hypothese	3
Werkwijze	3
Resultaten	4
Originele Kernel	4
BGD	5
Sobel	5
Laplacian	6
Verwerking	6
Conclusie	6
Evaluatie	7
Appendix	8
BGD	8
Sobel	9
Laplacian	10

Doel

Het doel van dit meetrapport is om te kijken welke van de gekozen methodes beschreven in het implementatieplan de meest betrouwbare is. Oftewel: hoe goed kan het programma de features nog herkennen met elke kernel en welke is het beste? Gemiddeld moet 90% van de tijd alle features correct worden herkent. Des te hoger het percentage, des te beter.

Hypothese

Wij vermoeden dat Sobel het meeste betrouwbaar is omdat deze de dikste lijnen oplevert en de feature detectie gemaakt is op de kernel van de originele code, die dikkere lijnen produceert. Sobel is alleen wel een ruis gevoeliger dan de originele kernel omdat sobel het gemiddelde van minder pixel neemt.

Wij vermoeden ook dat laplacian het slechtst zal werken omdat het onder andere geen gebruik maakt van ruisfiltering en daardoor veel meer false-positives genereert. Ook geeft laplacian er dunne edges weer waar het huidige systeem niet op gebouwd is.

Wij verwachten dat BGD net zo goed zal werken als laplacian maar minder goed dan sobel omdat BGD geen ruisfiltering heeft dus erg ruisgevoelig zal zijn.

Werkwijze

Wij zullen voor elke methode de features generen met de originele code als nulmeting. Daarna zullen wij kijken hoe goed elke methode de features heeft gedetecteerd vergeleken met de nulmeting. Wij zullen elke methode maar één keer op de foto toe passen omdat de uitkomst altijd hetzelfde zal zijn bij elke methode. Ook zullen de metingen op dezelfde computer gedaan worden.

stappenplan van de metingen:

1. Neem de plaatjes die als resultaat komen van het herkennen van de test-foto als nulmeting.
2. Doe hetzelfde voor elke te testen methoden (BGD, Sobel en Laplacian).
3. Vergelijk de resultaten of features op ongeveer dezelfde plek te vinden zijn en of de features wel herkent worden.

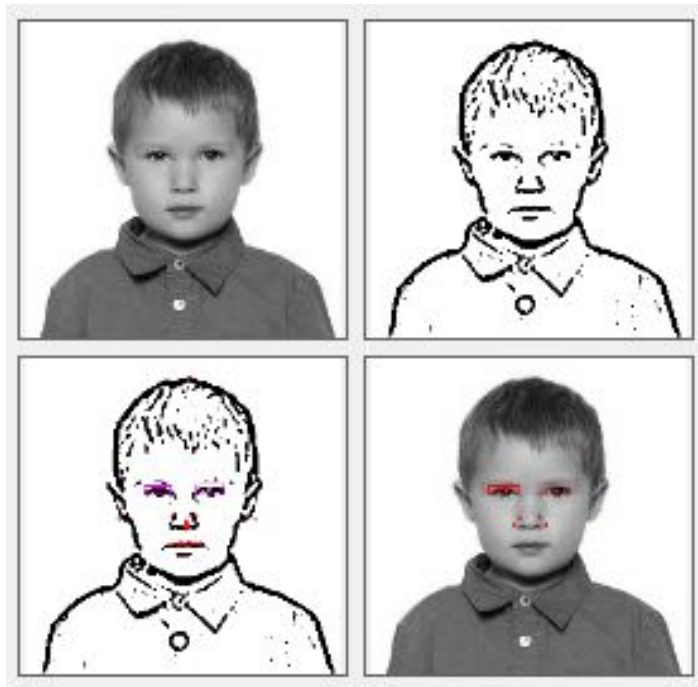
De foto waar de edge-detection overheen zal gaan:



Resultaten

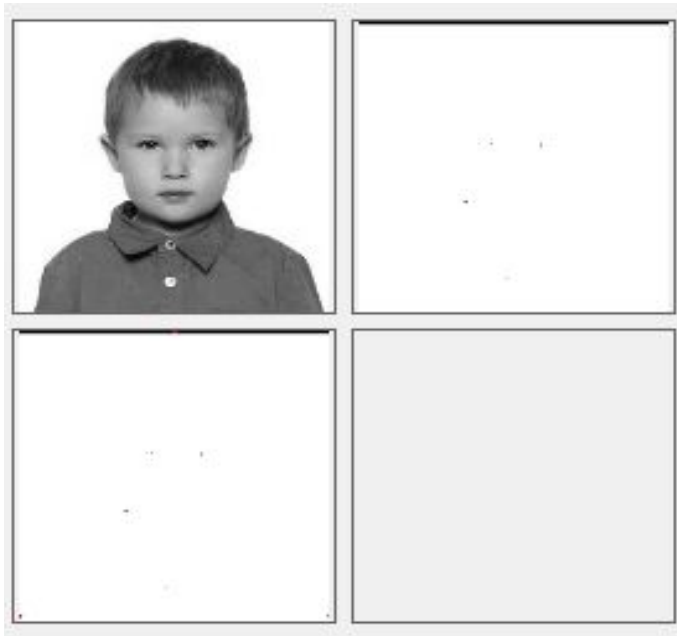
De resultaten waren onverwacht en we zijn erachter gekomen dat al onze kernels niet zullen werken in het huidige systeem.

Originele Kernel



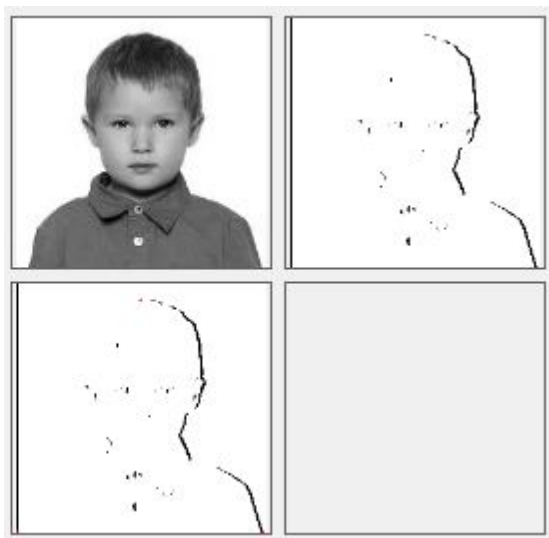
Dit zijn de uitkomsten van de originele kernel. Hierin is goed te zien dat deze kernel duidelijke lijnen weergeeft en hierdoor ook features gedetecteerd worden.

BGD



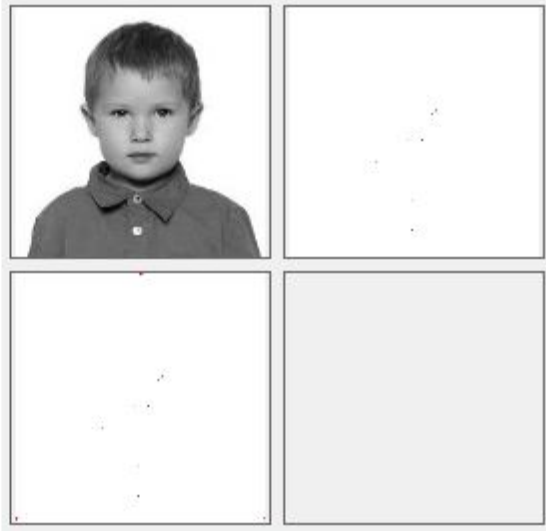
Dit zijn de uitkomsten van de BGD kernel. Hierin zijn maar een paar pixel uit de thresholding gekomen en aan de bovenkant een zwarte lijn.

Sobel



Dit is het resultaat van het programma met Sobel. Alleen een soort schaduw van de persoon is nog te zien.

Laplacian



Dit zijn de uitkomsten van de laplacian kernel. Hierin zijn maar een paar pixel uit de thresholding gekomen.

Verwerking

Onze hypothese lijkt totaal niet te kloppen. Dat is te zien aan het gebrek aan features in het vierde vakje van elke uitkomst. Maar ook dat de edge-detection, of specifieker; de thresholding niet goed lijkt te gaan. Ook tussen elke kernel is er eigenlijk niks te vergelijken omdat alle gegevens/pixels verdwijnen.

Als er gekeken wordt naar wat de edge-detection produceert in een stap buiten de werkwijze lijkt dat onze code echter wel goed functioneert (te zien in de appendix). Hierdoor vermoeden wij dat het gebrek aan een goede thresholding die compatibel is met onze code.

Conclusie

De conclusie uit deze metingen is dat BGD, sobel en de laplacian niet werken in het huidige systeem doordat er geen juiste thresholding is.

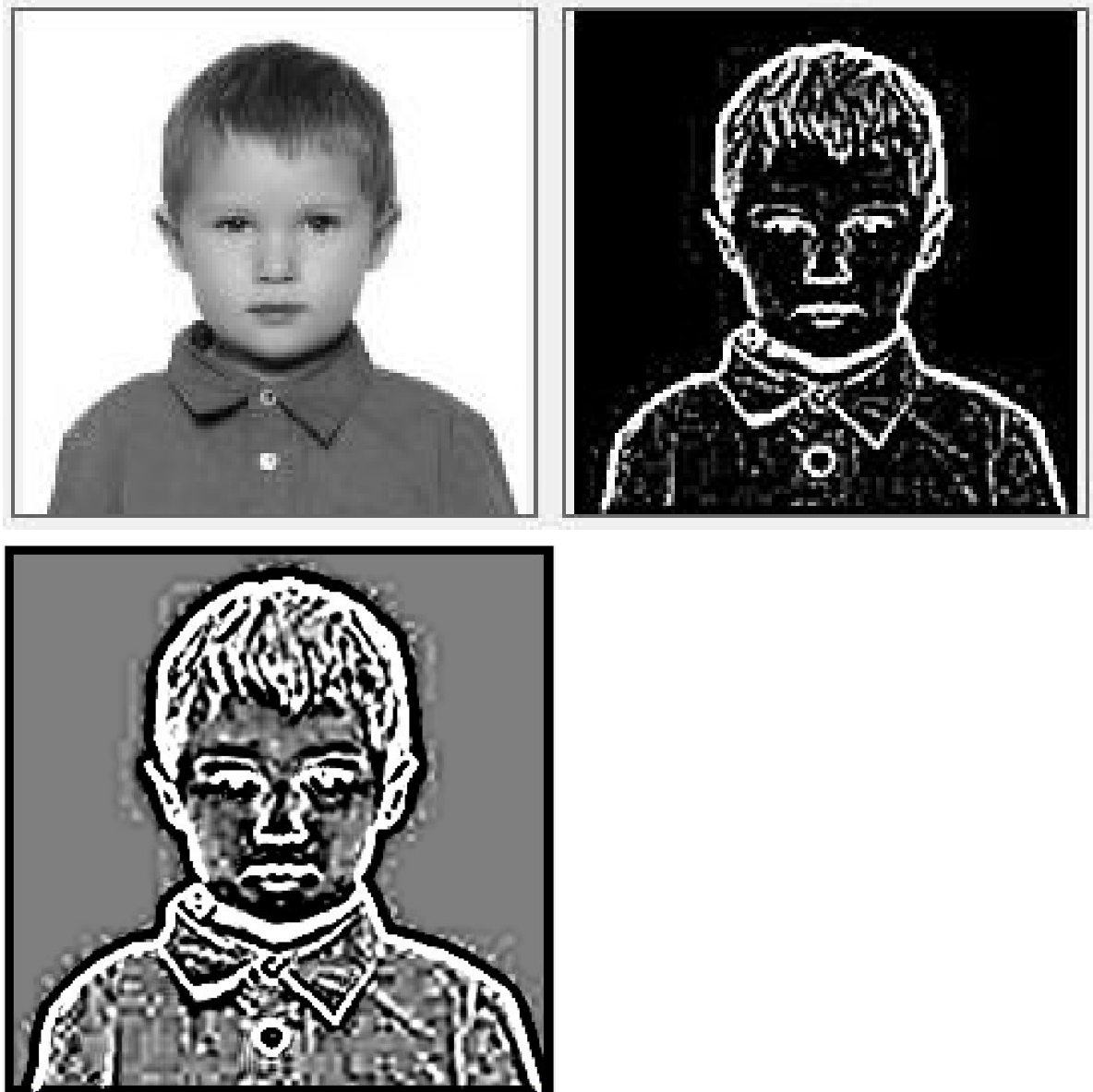
Evaluatie

Een combinatie van de documentatie niet goed doorlezen en geen rekening houden met de rest van het programma leidde ertoe dat de hypothese in dit meetrapport vanaf het begin al dreigde te falen.

Verdere evaluatie leidde er echter wel toe dat duidelijk werd dat dit vooral kwam door het gebrek aan compatible thresholding. De edge detection op zich werkt gewoon prima. Als we alleen naar de plaatjes uit de tussenstap kijken (zie appendix), dan lijkt dat kernel sobel de beste zou zijn geweest als we wel goede thresholding hadden en er een andere feature detectie is en er ruisfiltering aanwezig was.

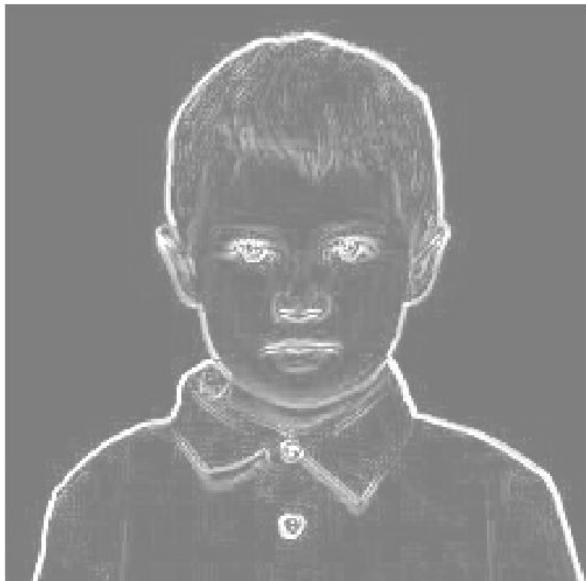
Appendix

De originele kernel



*Bij het onderste plaatje is 127 bij de intensiteit opgeteld. En is er geen thresholding uitgevoerd

BGD



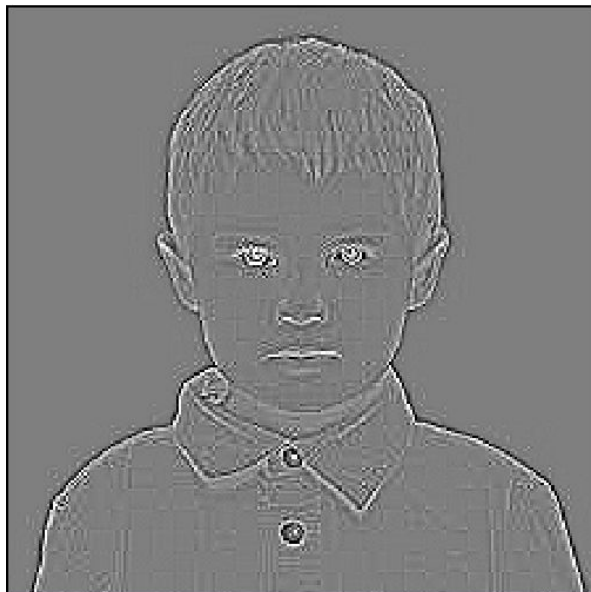
*Bij het onderste plaatje is 127 bij de intensiteit opgeteld. En is er geen thresholding uitgevoerd

Sobel



*Bij het onderste plaatje is 127 bij de intensiteit opgeteld. En is er geen thresholding uitgevoerd

Laplacian



*Bij het onderste plaatje is 127 bij de intensiteit opgeteld. En is er geen thresholding uitgevoerd