
Team 12	
Martijn Debeuf	44h
Toon Sauvillers	38h
Seppe Van Steenberghe	38h

Inhoudsopgave

1	Inleiding	2
2	Achtergrond	2
2.1	Gaussian Blur	2
2.2	Median Blur	2
2.3	Mean Blur	2
2.4	Fast Non-Local Mean Denoising	3
3	Methode	3
4	Bevindingen	3
4.1	Tijdsconsumptie	3
4.2	Juiste herkenning	4
4.3	Gefilterd en ongefilterd	5
5	Besluit	6
5.1	Toepassingen op het project	6



1 Inleiding

Het vorige verslag behandelde de kleurdetectie, met focus op de te herkennen kleur en de invloeden van omgevingsfactoren. Bij het herkennen van kleuren spelen oneffenheden in de foto's ook een rol, hiernaar zal nu verder onderzoek gedaan worden. Een foto is onderhevig aan verschillende factoren. Zo zal een artificiële intelligentie, die nu vaak standaard is ingebouwd in smartphones, de foto al bewerken. Lichtinval, een ander scherm of zelf een vuil scherm, maakt dat de kleuren niet egaal worden weergegeven.

In sectie 2 komen verscheidene filters aan bod die de oneffenheden en ruwheden in een foto kunnen reduceren. Dit zijn ook de filters die gebruikt worden in dit onderzoek. Op welke manier doen ze dit, hoe zijn deze geïmplementeerd... Vervolgens wordt de toegepaste methode uitgediept, waarna er meer uitleg gegeven wordt over de bevindingen van dit onderzoek. Een eerste zeer belangrijke factor is de tijdsconsumptie van de filter. Weegt de toeneming van uitvoeringstijd op tegen de verbetering van het resultaat? Daarnaast kan gekeken worden naar foto's die initieel niet correct werden gedetecteerd. Zorgt het toepassen van een filter ervoor dat deze wel correct wordt gedetecteerd? Welke filter heeft dan het beste resultaat? Zorgt het aanpassen van de kernel grootte voor betere detectie, en heeft dit effect op de uitvoeringstijd? Ten slotte wordt gekeken of er bevindingen kunnen opgenomen worden binnen het project.

2 Achtergrond

Er bestaat natuurlijk een breed gamma aan fotofilters. De benodigde code voor dit onderzoek is geschreven geweest in Python. Daarom is gekozen om vier filters uit de bibliotheek van OpenCV te gebruiken (zie sectie 3). Een eerste toegepaste filter is de *Gaussian Blur*. Gevolgd door *Median Blur* en *Mean Blur*. Bij deze methodes wordt gekeken naar de pixels rond een centrale pixel om zijn waarde te bepalen. Ten slotte is er ook een minder gekende ontruismethode gebruikt namelijk de *Fast Non-Local Mean Denoising*. Deze werkt anders dan de eerste drie.

2.1 Gaussian Blur

Deze methode filtert de hoge frequenties uit de foto en heeft dus het effect van een laag-doorlaatfilter. Er wordt een kerngrootte vooropgesteld. Deze grootte is de lengte van de matrix van pixels die bekeken wordt. Bijvoorbeeld indien de grootte 3 is zal voor elke pixel die gefilterd wordt een vierkant rond deze pixel opgesteld worden met zijde 3. Op elk van dit vierkant wordt de methode dan toegepast. NOG UITLEGGEN HOE GAUSSIAN PRECIES WERKT!!!!!!!!!!!!!!!!!!!!!!

2.2 Median Blur

Binnen de kern van pixels worden de waarden van de pixels gesorteerd van klein naar groot, daarna wordt de middelste waarde (mediaan) geselecteerd. De kleur van de centrale pixel van de kernel wordt dan op deze waarde geplaatst.

2.3 Mean Blur

Bepaald binnen de kern de gemiddelde waarde van de pixels. Daarna wordt de kleur van de centrale pixel op deze waarde geplaatst.

2.4 Fast Non-Local Mean Denoising

Aan de hand van de kern worden binnen de afbeelding gelijkende kernels gezocht. Gelijkende kernen worden bij elkaar geplaatst. Van elk van deze groepen wordt dan het gemiddelde bepaald. Daarna wordt elke pixel in elke kern binnen die groep op de gemiddelde waarde geplaatst.

3 Methode

Het onderzoek is volledig geautomatiseerd geweest aan de hand van Python. Binnen Python is gebruikt gemaakt van de bibliotheek OpenCv om de filters toe te passen op de afbeeldingen. De testen zijn uitgevoerd geweest op een gegevensbank met 670 afbeeldingen van schermen. Net zoals in het vorige onderzoek is gebruik gemaakt van verschillende omgevingsfactoren. Echter worden deze binnen dit onderzoek achterwege gelaten.

Om de kwalitatieve data te kwantificeren is gewerkt met een scoresysteem. Een foto wordt voordat deze geanalyseerd wordt, onderverdeeld in een aantal blokken. Daarna worden binnen elk blok tien procent van de pixels willekeurig gekozen. Het onderverdelen in blokken zorgt ervoor dat de pixels mooi verspreid liggen binnen de afbeelding. Daarna worden de verschillende methodes op de foto toegepast. Om uiteindelijk een score aan de afbeeldingen toe te wijzen worden de eerder willekeurig bepaalde pixels bekeken. De score geeft weer welk percentage van deze pixels juist gedetecteerd zijn geweest, zonder en na het uitvoeren van elke filter.

4 Bevindingen

Nu kan gekeken worden of het al dan niet interessant is om al dan niet de afbeeldingen te filteren. De eerste belangrijke indicator was tijdsconsumptie. Daarnaast kunnen de filters onderling vergeleken worden. Vervolgens kan gekeken worden of de kernel grootte een invloed heeft op de resultaten. Tenslotte wordt nagegaan of filteren effectief het detecteren van de kleur positief beïnvloed.

4.1 Tijdsconsumptie

Wanneer gekeken wordt naar figuur 1 valt meteen op dat *Fast Non-Local Mean Denoising* voor om het even welke kern grootte niet echt efficiënt is. Dit kon alreeds verwacht worden door de manier waarop deze methode te werk gaat. De waarde blijft nagenoeg constant geacht de kern grootte maar is trager met een orde van grootte 3 ten opzichte van de andere filters. Om deze reden zal deze filter al zeker niet opgenomen worden in het project.

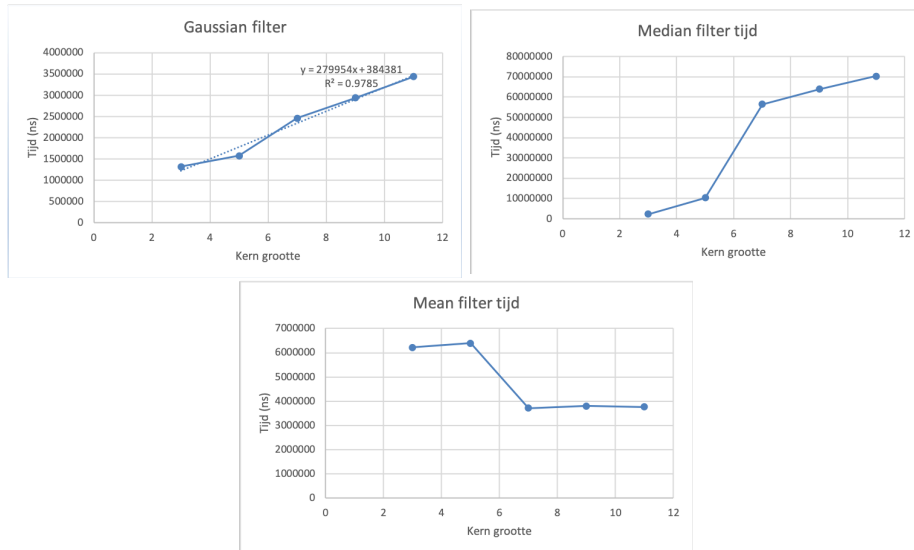
	Gemiddelde tijd (ns)			
Kernel grootte	Gaussian	Median	Mean	Fast NLM Denoising
3	1319454	2318204	6212660	2944694425
5	1571094	10311129	6388270	2768823310
7	2457046	56403157	3711259	2747259786
9	2936336	63908330	3791051	2784639628
11	3436375	70302982	3761732	2783860072

Figuur 1: Gemiddelde tijd nodig om filter op afbeelding toe te passen.

Voor de andere filters kunnen hun individuele tijdsgrafieken 2 van dichtbij bekeken worden. Bij de *Gaussian filter* is een duidelijk lineair verband tussen kerngrootte en tijdsconsumptie zichtbaar. Dit wordt bevestigd door de determinatiecoëfficiënt van 97,85 procent van de lineaire trendlijn.

Verder zijn er nog twee opvallende zaken. Enerzijds zorgt een grotere kern ervoor dat de *Median filter* veel meer tijd nodig heeft. De reden hiervoor is dat hoe groter de kern, hoe meer elementen pixelwaarden gesorteerd moeten worden van van klein naar groot.

Bij de *Mean filter* houden de waarden er nagenoeg constante waarde. Rond kern grootte 6 gebeurt een sprong in uitvoeringstijd. WHY????????????????????????????????



Figuur 2: Tijdsgrafieken voor de verschillende filters

Wanneer naar de tijdscomponent gekeken wordt geven *Gaussian* en *Median filter* de beste resultaten. Op basis van tijd wordt de laatste methode al afgeschreven.

4.2 Juiste herkenning

Vervolgens kan ook gekeken worden of het toepassen van een filter op een onjuist gedetecteerde foto een al dan niet positief effect heeft. Hiervoor kan onderscheid gemaakt worden tussen afbeeldingen die initieel juist en fout zijn gedetecteerd. In figuur 3 worden de proporties weergegeven die minder scores na het toepassen van een filter op een initieel fout gedetecteerde afbeelding. Dit geeft dus het percentage foto's weer die nog steeds niet gedetecteerd worden na het filteren ervan.

Filterscore <= ongefilterde score na foute initiële detectie				
Kernel grootte	Gaussian	Median	Mean	Fast NLM Denoising
3	0.9942529	0.994253	0.9942529	0.988571429
5	0.9942857	1	0.9942857	0.994285714
7	0.9942857	1	1	0.988636364
9	0.9942857	1	0.9942857	0.994285714
11	0.9942857	1	0.9942857	0.988636364

Figuur 3: Proportie slechter gedetecteerde foto's na foute initiële detectie.

Het is opmerkelijk hoe dicht alle waarden bij elkaar liggen. Ze leunen allen zeer dicht aan tegen de één. Dit wil dus zeggen dat de filters de resultaten niet verbeterd hebben, maar eigenlijk enkel slechter hebben gemaakt. Afbeeldingen die initieel slecht scoren zullen na de filtering geen significant beter resultaat opleveren.

Daarnaast zijn er degene die initieel juist gedetecteerd zijn. Scoren deze beter indien er een filter op wordt toegepast? Uit figuur 4 wordt al snel duidelijk dat er hier een verband is met kerngrootte. Indien een kern van grootte zeven à negen wordt gebruikt zullen alle afbeeldingen een hogere score hebben en dit bij alle geteste filters. Bij een grootte van drie zal ongeveer acht procent van de foto's een hogere score hebben. Kernen van vijf en zeven zorgen voor een middelmaat van om en beide 40 en 63 procent respectievelijk. Tussen de filters onderling zijn bij gelijke kerngroottes geen significante verschillen op te merken. De waarden zijn zo goed als gelijk.

Filterscore <= ongefilterde score na correcte initiële detectie				
Kernel grootte	Gaussian	Median	Mean	Fast NLM Denoising
3	0.9193548	0.925403	0.921371	0.925252525
5	0.5838384	0.584677	0.5838384	0.581818182
7	0	0	0	0
9	0	0	0	0
11	0.369697	0.368952	0.369697	0.36437247

Figuur 4: Proportie slechter gedetecteerde foto's na foute initiële detectie.

4.3 Gefilterd en ongefilterd

Om het uiteindelijke oordeel te vellen of filteren al dan niet een verbeterende factor kan zijn binnen het project kan nog een laatste vergelijking gedaan worden. Namelijk kijken naar het verschil in juist gedetecteerde afbeeldingen per gebruikte methode alsook zonder het gebruik van een filter.

Filterscore >= 83%					
Kern grootte	Ongefilterd	Gaussian	Median	Mean	Fast NLM Denoising
3	496	496	496	496	495
5	495	495	496	495	495
7	495	495	496	496	494
9	495	495	496	495	495
11	495	495	496	495	494

Figuur 5: Aantal juist gedetecteerde foto's van de 670, gegeven de gebruikte filter.

Alweer zijn de waarden nagenoeg gelijk. Van de 670 afbeelding worden er bij elke toegepaste filter en gebruikte kerngrootte gemiddeld 495 correct gedetecteerd. Dit wijkt slechts af met maximaal één. Hieruit volgt alweer dat een filter toepassen nagenoeg geen effect heeft op het detecteren van de kleuren.

5 Besluit

Verschiedende filters zijn getest en vergeleken geweest. Dit is gebeurd op vlak van tijdsconsumptie en aan de hand van de toegekende score. Die score werd berekend op basis van de gedetecteerde waarde van willekeurig gekozen pixels. Uit de resultaten kan geconcludeerd worden dat het toepassen van een filter geen significante verbetering teweeg brengt. De tijdsconsumptie van de filters weegt niet op tegen de prestatiewinst van het algoritme.

5.1 Toepassingen op het project

Aangezien er geen opvallende resultaten uit het onderzoek zijn voortgekomen, zullen er ook geen doorgevoerd kunnen worden naar het project. De applicatie zou een langere tijd nodig hebben om de afbeelding van de opstelling te analyseren zonder dat de kleurdetectie verbeterd is. Een sneller reagerende applicatie wordt dan voorgenomen.