

Lichtintensiteit, Resolutie en saturatie invloed op grayscale algoritmes gebruikt in facial recognition

Namen en Datum

Mick Bos
Ties Brouwer

12-03-2020

Doel

Het doel van dit onderzoek is om te onderzoeken of er voor facial recognition software gebruikt in securitycamera's uit een selectie grayscale conversie algoritme een effectiever algoritme te vinden is in vergelijking met de standaard implementatie. Dit wordt gebaseerd op hoe de verschillende algoritmes onderscheiden wanneer zij worden getest met portretten die verschillen in lichtintensiteit, saturatie en resolutie, aangezien de software voor camera situaties die hierop veel kunnen verschillen.

Hypothese

Dit onderzoek werkt met verschillende grayscale conversie algoritmes die onderling verschillende kwaliteitseigenschappen hebben (zie: Werkwijze). Voor de verschillen in Lichtintensiteit, Saturatie en Resolutie waardes maken wij de volgende verwachtingen voor de algoritmes

- **Lichtintensiteit**
Wij verwachten dat het *Luma algoritme* de beste resultaten zal leveren. Dit algoritme verkrijgt zijn resultaat, net zoals *Luminance*, door de RGB-waardes met vaste waardes te vermenigvuldigen. Het verschil ligt erin dat *Luma* haar resultaten duidelijkere verschillen tussen lichtere kleuren bevatten.
- **Saturatie**
Wij verwachten dat het *Value algoritme* de beste resultaten zal leveren. Value haalt zijn werking door de RGB-kanalen gelijk te trekken, wat overheen komt met het verhogen/verlagen van het *value* kanaal uit het HSV color space. De werking van dit algoritme zou moeten kunnen vermijden dat saturatie verschillen impact op het resultaat zullen leveren.
- **Resolutie**
Aangezien resolutie verschil geen invloed heeft op de individuele RGB-kanalen verwachten wij dat de algoritmes in hun resultaat het meest zullen verschillen in hun mogelijkheid tot het creëren van geleidelijk kleurverschil. Hierdoor verwachten wij, net als bij lichtintensiteit, dat het *Luma algoritme* de hoogste Hit/miss ratio zal leveren. Er wordt bij de *Luma* resultaten extra nadruk op kleurverschil gezet, waardoor de facial recognition software gemakkelijker features zou moeten kunnen herkennen.

Werkwijze

Voor dit onderzoek worden de volgende algoritmes¹ getest:

- **Intensity Algorithm**
 $\text{Result} = ((1/3) * (R + G + B))$
- **Value Algorithm**
 $\text{Result} = \text{MAX}(R, G, B)$
- **Luminance Algorithm**
 $\text{Result} = (R * 0.3) + (G * 0.59) + (B * 0.11)$
- **Luma Algorithm**
 $\text{Result} = (R * 0.2126) + (G * 0.7152) + (B * 0.0722)$
- **Luster Algorithm**
 $\text{Result} = (1/2) * (\text{MAX}(R, G, B) + \text{MIN}(R, G, B))$

De verschillende algoritmes worden voor Lichtintensiteit, Saturatie en Resolutie ieder getest op een bijpassende testset. Elke testset bestaat uit een verzameling van 5 verschillende portretten (figuur 1), waarvoor voor elk portret 100 versies worden getest. Deze versies lopen van een factor (0) voor foto 0 naar een factor (100) voor foto 100, waarbij de factor gelijkstaat aan de waarde levels van hun corresponderende invloed. (figuur2)

Figuur 1



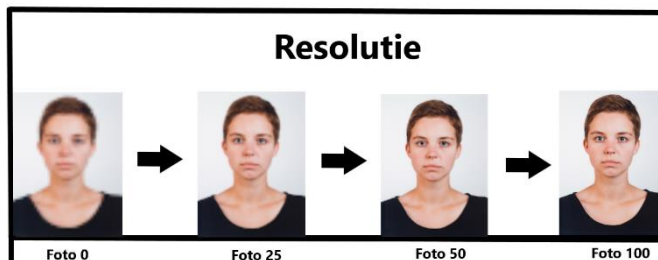
Figuur 2



Figuur 2



Figuur 2



1) "Color-to-Grayscale: Does the Method Matter in Image Recognition?", PLoS One 2012

De verschillende software steps waar het resultaat van de algoritmes op wordt getest bestaat uit:

1. PreLocalization
2. Localization (1)
3. Localization (2)
4. Localization (3)
5. Localization (4)
6. Localization (5)
7. Prepare extraction
8. Extraction (1)
9. Extraction (2)
10. Extraction (3)
11. Extraction (4)
12. PostProcessing

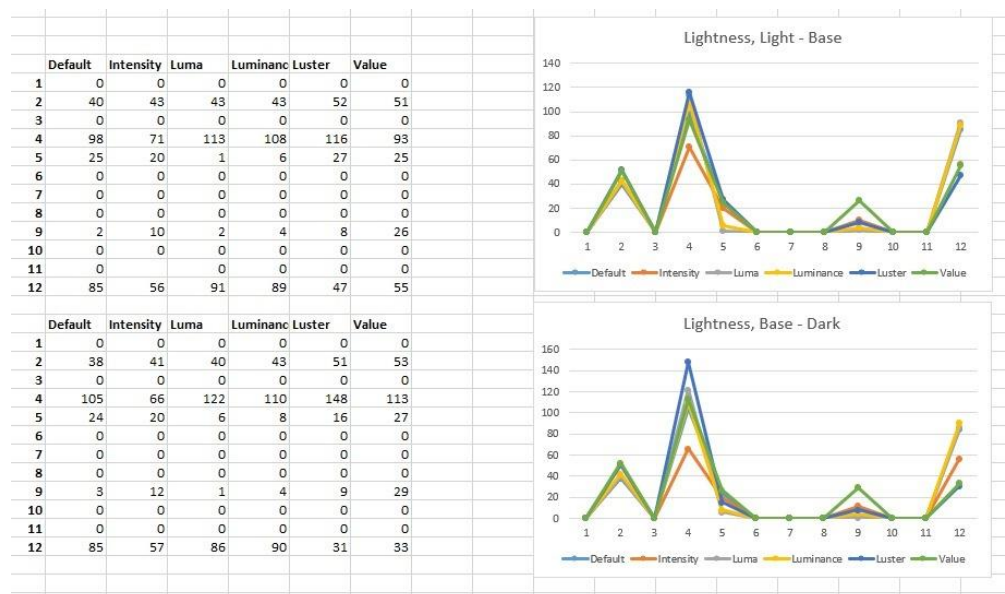
De algoritmes worden op de volgende wijze individueel getest:

- Voer voor lichtintensiteit, saturatie en resolutie de volgende stappen uit:
 1. Verwerk het eerste algoritme in de facial recognition software als grayscale conversie wijze.
 2. Laad de eerste testset in en gebruik het huidige algoritme om dit portret te verwerken tot grayscale. Het algoritme slaat nu op welke software steps er mogelijk waren voor de resulterende grayscale.
 3. Herhaal stap 2 tot alle testsets gemeten zijn.
 4. Bereken en noteer voor het algoritme hoe vaak een software step is behaald over de verschillende portret testsets.
 5. Herhaal stap 1 tot en met 3 voor de overige algoritmes.

Resultaten

De volgende 3 meetresultaten zijn afleidingen van de verkregen data. De tabellen aan de linkerkant staan voor hoeveel steps een algoritmes over al zijn testsets heeft verkregen Deze tabellen zijn opgesplitst in 2, waarvan de bovenste staat voor alle portretten oplopend van een factor (0) tot factor (50), en de onderste voor alle portretten van factor (50) tot factor (100). Rechts staan de grafieken die erop gebaseerd zijn.

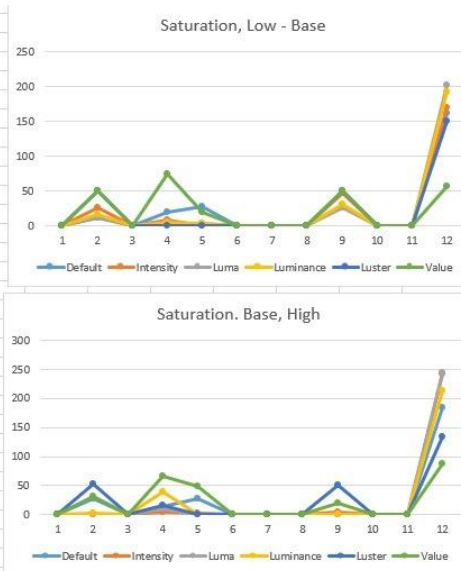
Resultaten Lichtintensiteit



Resultaten Saturatie

	Default	Intensity	Luma	Luminanc	Luster	Value
1	0	0	0	0	0	0
2	12	26	14	16	50	50
3	0	0	0	0	0	0
4	19	8	4	6	0	74
5	28	0	4	4	0	19
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	30	47	26	31	50	50
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	161	169	202	193	150	57

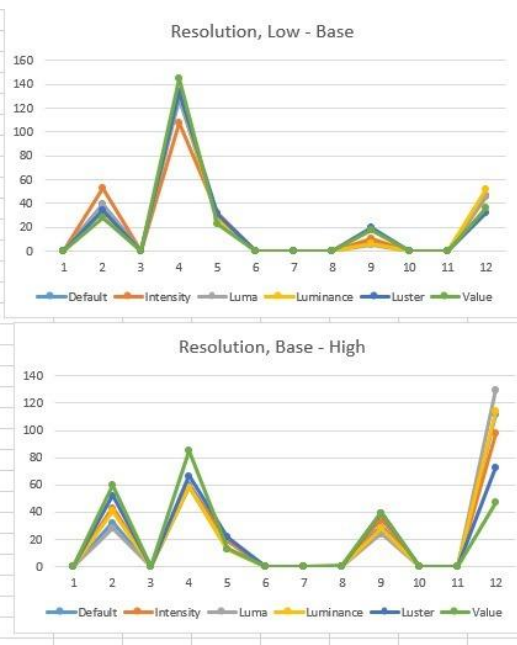
	Default	Intensity	Luma	Luminanc	Luster	Value
1	0	0	0	0	0	0
2	28	0	1	2	52	31
3	0	0	0	0	0	0
4	14	5	10	40	15	66
5	28	3	0	0	1	49
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	1	0
9	0	4	0	0	51	20
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	185	243	244	213	135	87



Resultaten Resolutie

	Default	Intensity	Luma	Luminanc	Luster	Value
1	0	0	0	0	0	0
2	39	53	38	33	34	28
3	0	0	0	0	0	0
4	128	108	132	132	133	145
5	28	32	28	27	31	23
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	8	11	5	6	20	18
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	47	46	47	52	32	36

	Default	Intensity	Luma	Luminanc	Luster	Value
1	0	0	0	0	0	0
2	33	44	28	41	53	60
3	0	0	0	0	0	0
4	61	59	59	59	67	85
5	19	19	13	13	22	14
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	1
9	30	35	25	28	40	40
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	112	98	130	114	73	47



Verwerking

Uit de meetresultaten is een conclusie te trekken op basis vanwaar de pieken in de grafiek te vinden zijn. Hierbij is het belangrijk hoe de verschillende algoritmes met elkaar vergelijken op de grafiek pieken.

Verwerking Lichtintensiteit

Het is duidelijk te zien dat het *default algoritme*, *luma* en *luminance* erg gelijklopen met hoe effectief zij werken wanneer het gaat op lichtere portretten. Dit is te vinden op basis van hoeveel portretten zij het gehele proces hebben laten doorlopen. Dit fenomeen zie je ook terugkomen in de tweede grafiek. Hier is te zien is dat de verschillen in resultaten nog steeds gelijklopen. Daarbij is aan te merken dat het *value* algoritme vaak vastliep bij de extracties, maar er wel een heel aantal doorheen heeft gekregen.

Verwerking Saturatie

Uit de grafieken is duidelijk te lezen dat ieder algoritme behalve *value* hier een stuk minder last had van de verschillen in saturatie. Daarbij behalen *Luminance* en *Luma* op een aantal haperingen na van de verschillende algoritmes de meeste portretten die door het gehele proces zijn gekregen. Verder behaalt *default implementatie* resultaten die ver achterwegen blijven hangen in vergelijking met deze twee. Bij de tweede grafiek valt dit anders, aangezien *Luminance* dit keer lagere eindresultaten behaalt dan *Luma*. Daarnaast behaalt *Intensity* dezelfde resultaten als *Luma*, terwijl het bij de lichtere portretten juist mindere resultaten leverde.

Verwerking Resolutie

Bij resolutie is het een echte nek aan nek race net zoals bij saturatie en lichtintensiteit. Zoals je kan zien zijn de resultaten een stuk hoger in het high gedeelte van de grafiek. Dat is natuurlijk omdat de algoritmes beter de kans krijgen om kleurverschil te onderscheiden wanneer er meer data is om te verwerken. Ook is er hier weer een winnaar aan te wijzen. Namelijk *Luma*. Al moeten we daar wel bij zeggen dat de *default implementatie* en *luminance* ook zeker goede opties zijn. Ook is het wel interessant dat *Luminance* een betere optie is als het gaat om plaatjes met een lagere resolutie.

Conclusie Lichtintensiteit

De verschillende algoritmes behalen niet en beter resultaat dan de huidige standaard implementatie wanneer deze vergeleken worden in hoeveel stappen zij voldoen. *Luminance* en *Luma* behalen wel vergelijkbare resultaten.

Conclusie Saturatie

Luminance en *Luma* behalen beide betere resultaten dan de standaard implementatie, maar hebben niet genoeg onderlinge verschillen in resultaat om hen van elkaar te kunnen onderscheiden.

Conclusie Resolutie

Het *Luma* algoritme heeft in de resolutietest de beste hit ratio, waardoor wij die ook zeker aanraden als het gaat om de oude implementatie vervangen. Toch moet hier wel bij vermeld worden dat *luminance* ook een goede optie kan zijn, maar dan vooral in een situatie waar resolutie een probleem kan zijn

Evaluatie

Het beste algoritme om de *default implementatie* te verbeteren is *luma*. *Luma* behaalt in de verschillende factoren een van de hoogste al niet de hoogste Hitratio's. Dit houdt ook direct in dat de *default implementatie* bij geen van de tests betere resultaten heeft geleverd. *Luminance* is ook zeker een goede optie maar dan vooral in combinatie met minder resolutie

Aan dit onderzoek zitten nog enige meetonzekerheden gebonden. De eerste is dat het testen van de algoritmes is afgelegd in een facial recognition software met specifieke eisen voor portretten die het kan herkennen of zelfs wilt accepteren. Dit heeft als gevolg gehad dat de tests zijn afgelegd met maar 5 verschillende portretten in plaats van een grotere testset. Dit wil niet zeggen dat er uit de huidige resultaten geen conclusie te trekken valt, maar het betekent wel dat de bevonden resultaten kunnen afwijken wanneer grotere testsets worden getest.