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 testen welk grayscale conversie algoritme beter werkt dan de standaard implementatie, met blik op verschil in lichtintensiteit, saturatie en resolutie

# 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 overkleurverschil gezet, waardoor de facial recognition software gemakkelijker features zou moeten kunnen herkennen.

# Werkwijze

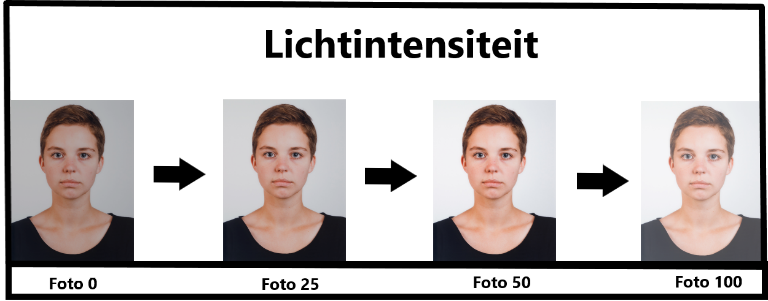
Voor dit onderzoek worden de volgende algoritmes1 getest:

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

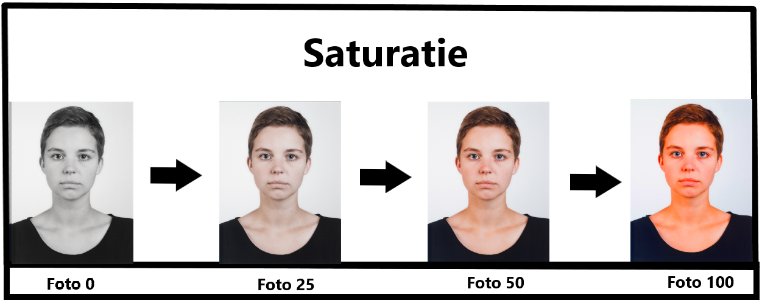
De verschillende algoritmes worden voor Lichtintensiteit, Saturatie en Resolutie ieder getest op een bijpassende testset. Elke testset bestaat uit een verzameling van 5 verschillende portretten(figuur 1), waarvoor voor elk portret 100 versies worden getest. Deze versies lopen van een factor (0) voor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



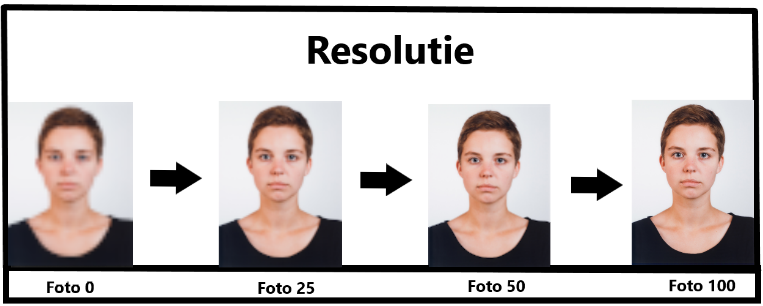
*Figuur 1*



*Figuur 2*



*Figuur 3*



*Figuur 4*

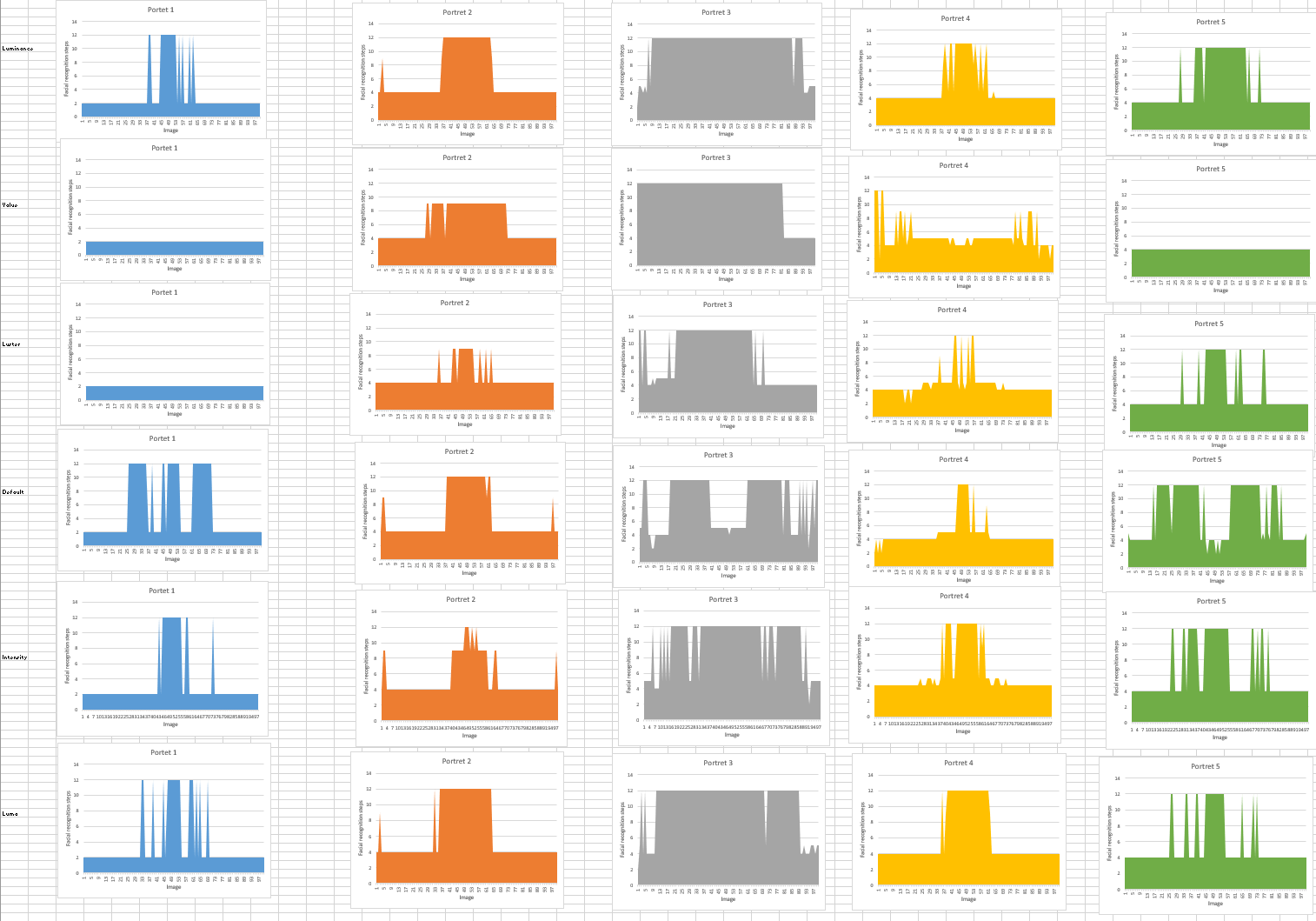
De algoritmes worden op de volgende wijze individueel getest:

1. Verwerk het algoritme in de facial recognition software als grayscale conversie wijze.
2. Laad van de eerste verzameling portretfoto’s (0) in en gebruik het huidige algoritme om dit portret te verwerken tot grayscale.
3. Verwerk de foto door de facial recognition software en noteer per facial feature of deze herkend is door de software.
4. Herhaal stap 3, 10 keer2 .
5. Herhaal nu stap 1 tot en met 4 voor elk portret in de testsets en verwerk deze in een overzichtelijke tabel en grafiek

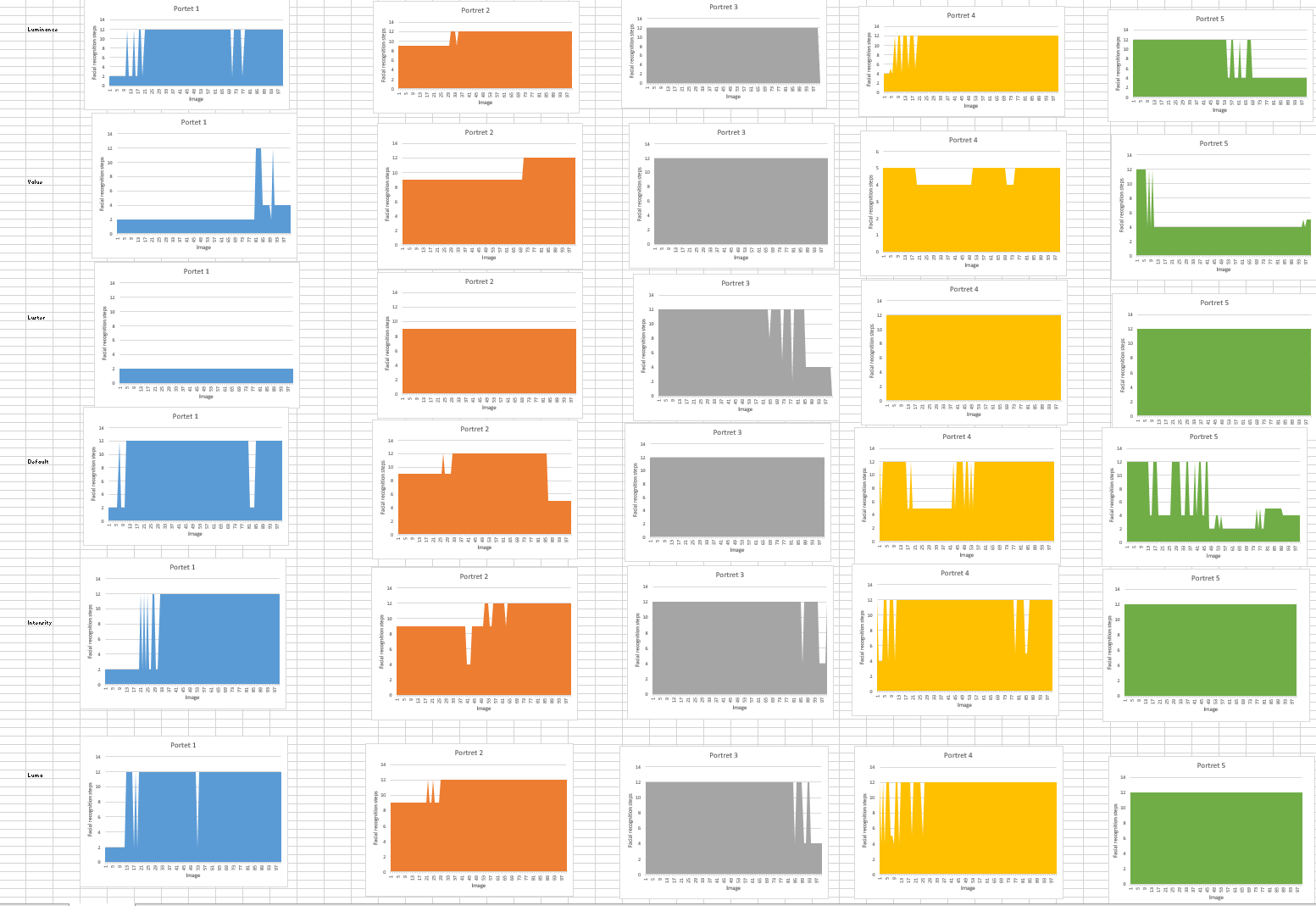
# Resultaten

De resultaten hebben wij zo overzichtelijk mogelijk gemaakt door bij elke stap van de facial recognition software een nummertje toe te kennen. Bij nummer 12 heeft de software het plaatje kunnen herkennen en alles kunnen lokaliseren. De lager dit nummer, de minder stappen zijn gelukt

# Resultaten Lichtintensiteit



# Resultaten Saturatie



# Resultaten Resolutie

# 

# Verwerking

Wij hebben onze meetresultaten niet verwerkt om onze conclusie te trekken. Dit doen wij door per algoritme en plaatje de hit ratio te pakken en daar de conclusie op te baseren

# Conclusie Lichtintensiteit

Lichtintensiteit is een lastige om een duidelijk antwoord te formuleren. Als je de gemiddelde hit ratio pakt van lichtintensiteit, kom je erachter dat Luma en Luminance ongeveer dezelfde hit ratio hebben. Daardoor zijn beide zeker geschikt als algoritme voor grayscaling. Ook is de hit ratio van deze algoritmes beter dan de rest, zonder dat deze hele rekenkracht-vragende berekeningen hebben.

# Conclusie Saturatie

Zoals je duidelijk kan zien, is er in de strijd der algoritmes als het gaat om Saturatie een duidelijke winnaar. Het onderste algoritme genaamd Luma is duidelijk de beste van de gekozen algoritmes. Luma heeft de beste hit ratio van alle algoritmes.

# Conclusie Resolutie

Als we kijken naar de diagrammen van Resolutie, is goed te zien dat Luma de beste optie is van alle vijf gekozen algoritmes. De hit ratio van het Luma algoritme is ook gemiddeld het hoogste van alle vijf gekozen algoritmes.

# Evaluatie

Om de hypothese van dit meetrapport te beantwoorden: Het beste algoritme om de default implementatie te verbeteren is Luma. Luma heeft de hoogste hit ratio van alle geteste algoritmes.

Als je kijkt naar de hypothese die wij gedaan hebben, valt te zien dat we een goede hypothese hebben gedaan. Van de drie hypotheses, zijn er twee waar. Dit komt omdat het Luma algoritme goed zijn in het creëren van geleidelijk kleurverschil. Wij hadden niet verwacht dat Luma ook het beste zal zijn in het saturatiegedeelte van het onderzoek.

Om het verschil in hit ratio te verduidelijken, moet er natuurlijk wel gezegd worden dan de rest van de facial recognition software niet is afgestemd op het grayscale algoritme. Dit kan komen omdat de software ontwikkeld is met een bepaalde implementatie in gedachten, waardoor deze beter werkt of minder goed werkt met bepaalde algoritmen. De resultaten die wij hebben gekregen zijn daardoor ook heel erg implementatie gevoelig.