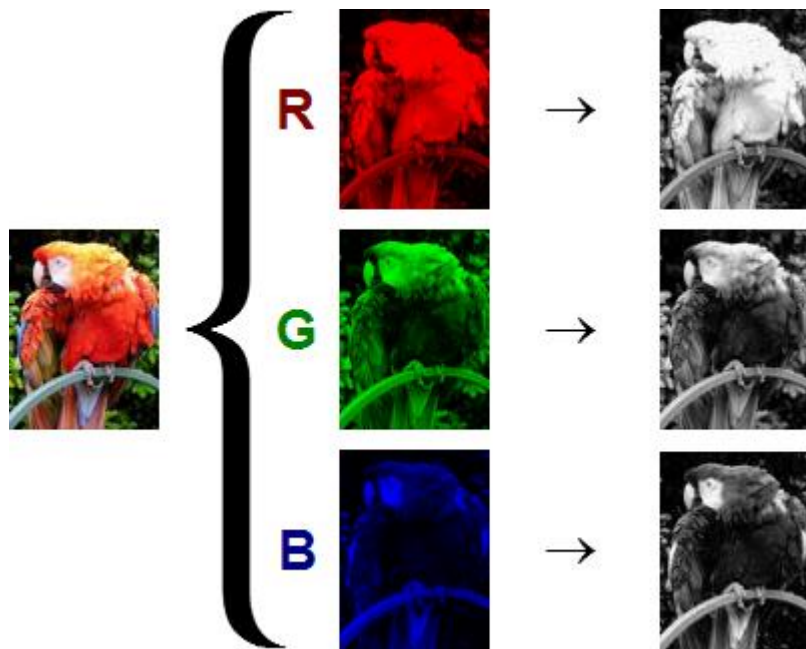


Meetrapport Conversie naar grijswaarde Algoritmes



Gemaakt door:

Patrick Dekker & Mark Gasse

Datum:

13 April 2019

Inhoudsopgave

- Inleiding	p.1
- Doel	p.1
- Hypothese	p.1
- Werkwijze	p.1
- Resultaten	p.4,5
- Verwerking	p.6,7
- Conclusie	p.7
- Evaluatie	p.7

Inleiding

Er zijn verschillende algoritmes voor het converteren van een kleuren naar een grijswaarde afbeelding. Elke algoritme heeft een andere doel einde als resultaat. Zo heb je een formule die het beste is voor het menselijk oog, een die zo snel mogelijk is wat vaak wordt gebruikt bij de camera's van telefoons.

Doel

Het doel van dit meetrapport is om er achter te komen welk algoritme het meest accurate beeld geeft van de afbeelding. Bij het gebruik van welk algoritme is het vervolg proces van gezicht kenmerken het beste en ziet er daarnaast het beste uit voor het menselijk oog

Hypothese

Wij verwachten dat Luma het beste resultaat geeft voor het menselijk oog en daarbij ook redelijk goed het vervolg proces van gezichtsherkenning zal uitvoeren.

Werkwijze

Om er achter te komen wat welk algoritme het beste resultaat oplevert. Gaan we een aantal algoritmes gebruiken en daarbij weer een aantal gezichten van personen om een nauwkeuriger resultaat te krijgen.

Voor het meten van de resultaten zijn een aantal benodigdheden nodig:

- Kleuren afbeelding (van het gezicht van een persoon)
- Een programma dat de afbeelding converteert (in ons geval ExternalDLL)












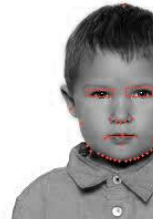







Stappenplan werkwijze:

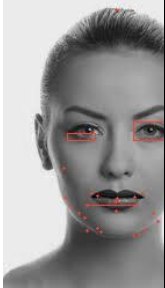
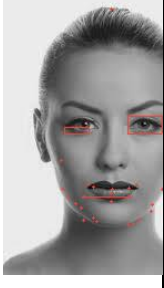
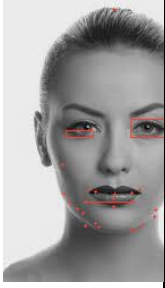

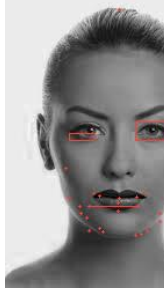











1. Kies een kleuren afbeelding
 - a. Kan je veranderen in de main.cpp, verander loadImage path
2. Kies een algoritme
 - a. Kan je veranderen in de functie StepToIntensityImage van StudentPreProcessing.cpp
3. Voer het programma uit
4. Sla de grijswaarde afbeelding & de afbeelding met gezicht herkenningpunten ergens op (bv. in een tabel van je test rapport)
 - a. Afbeelding te vinden in de Debugpicture folder
5. Herhaal dit vervolgens met een aantal formules (stap 2 t/m 4)
6. Vervolgens test je dit nog een keer met verschillende afbeeldingen(stap 1 t/m 5)
7. Als laatste nog aan een aantal personen vragen welke afbeelding de beste conversie heeft (bv. doormiddel van een enquête)

Wij gaan de resultaten vergelijken van de default implementatie, Averaging, Luma & Single color channel. Voor de default hebben we gekozen omdat we onder andere het verschil willen zien met onze eigen implementatie. Daarnaast dus een algoritme voor het menselijk oog (Luma), een simple en snelle manier (single color channel) & een die het gemiddelde van de kleuren gebruikt (Averaging).

Resultaten

In de onderstaande tabel zijn onze resultaten te zien. Aan de linker kant staat de originele afbeelding en daarnaast de resultaten van de verschillende algoritmes. Onder elk van de geconverteerde afbeeldingen staat het aantal personen die deze conversie het beste vinden en daaronder is in elke afbeelding te zien hoe goed de gezichtsdelens herkenning ging.

origineel	default	Averaging	Luma	Single color channel (green)	Single color channel (blue)	Single color channel (red)
						
Beste resultaat(a antal keer)	1	3	3	2	0	0
						X
						
Beste resultaat(a antal keer)	2	2	2	3	1	0

						X
						
Beste resultaat(a antal keer)	1	1	4	3	0	0
					X	X

Bij de afbeelding van de man met de *single color channel(blue)*, komt het programma niet door het proces van het vinden van de neus einde en het oog gebied. De resultaten van *single color red* zijn geen 1 keer door het hele proces gekomen.

In de onderstaande tabel zijn de cijfers (op de schaal van 1 t/m 10) te zien op basis van hoe moeilijk ze het vinden om de beste conversie te kiezen.

	Persoon 1	Persoon 2	Persoon 3	Persoon 4	Persoon 5	Persoon 6	Persoon 7	Persoon 8	Persoon 9
cijfer	7	7	8	2	3	6	5	6	7

Verwerking

In de onderstaande cirkel diagram is te zien welk algoritme de image conversie het beste heeft gedaan volgens de gevraagde test personen.

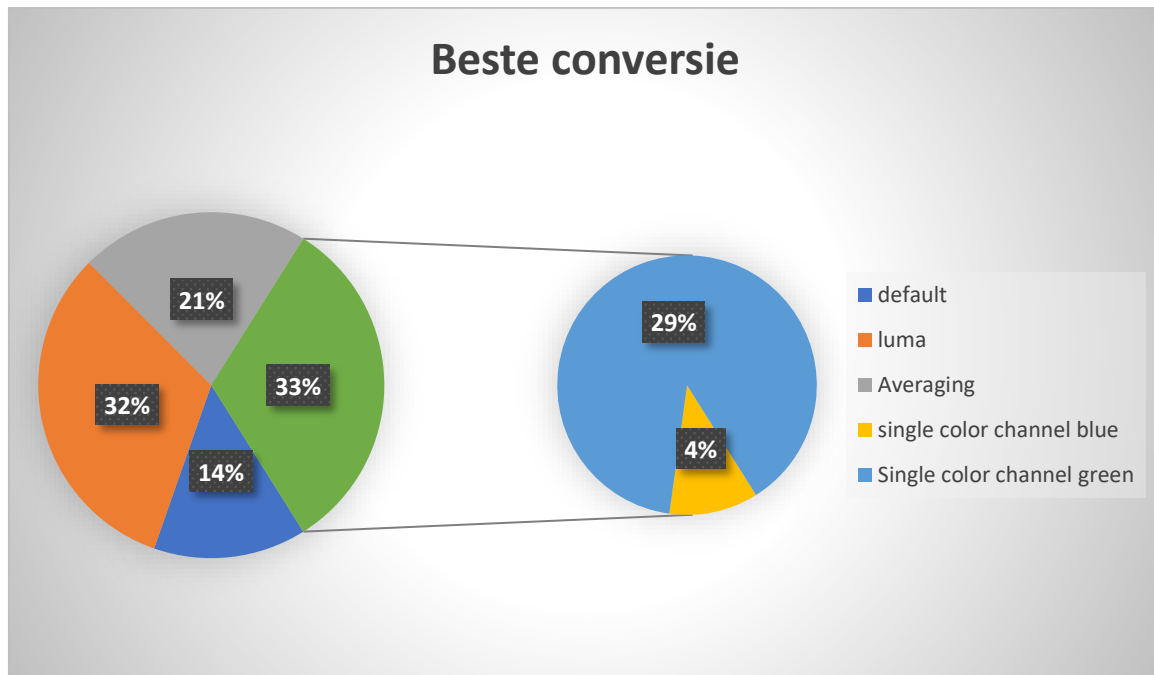


Diagram 1

In *diagram 1* is te zien dat volgens de test personen Luma de beste conversie is. Wat het meeste opvalt is dat de single *color channel green* heel hoog scoort ondanks het gebruik van maar 1 kleur.

In de onderstaande tabel is te zien of het lukt om het programma door het hele proces te laten lopen. Hierin is te zien dat met *single color channel red* het proces er nooit door heen komt & bij *single color channel blue* alleen bij het laatste afbeelding van de man.

	default	Averaging	Luma	Single color channel green	Single color channel blue	Single color channel red
Afbeelding 1 (kind)	✓	✓	✓	✓	✓	X
Afbeelding 2(vrouw)	✓	✓	✓	✓	✓	X
Afbeelding 3(man)	✓	✓	✓	✓	X	X

Tabel 1

Conclusie

Volgens *diagram 1* is Luma de beste algoritme voor conversie naar grijswaarde. Onze hypothese dat de algoritme Luma dat het beste is voor het menselijk oog ook echt het meest geschikt hiervoor is. Het valt wel op dat het verschil tussen de verschillende algoritmes niet al te groot is.

Uit *tabel 1* blijkt dat *single color channel* niet geschikt is voor dit programma. Bij het gebruik van alleen de kleur rood zijn alle drie de afbeeldingen niet door het hele proces heen gekomen. En bij blauw de laatste afbeelding. Hieruit kunnen we concluderen dat het resultaat van *single color channel* te veel afhangt van het inkomende beeld en dus niet geschikt is.

Evaluatie

In een volgend meetrappport zouden er meet test personen moeten zijn omdat je resultaat accurater is hoe meer test personen je hebt. Verder zouden er ook meer dan 3 test afbeeldingen gebruikt kunnen worden wat ook zorgt voor een accurater resultaat.