

Vision - Implementatieplan



Oscar Kromhout en Justin van Ziel
17-02-2020

Inhoudsopgave

Inhoudsopgave	2
Doel	3
Algoritmes	3
Intensity methode	3
Gleam	3
Luminance	4
Lightness	4
Value	4
Gekozen algoritmen	4
Methode van testen	5
Bronnen	6

Doel

Het doel is om te zorgen dat het standaard aangeleverde facial recognition algoritme beter werkt op onder en overbelichte foto's. Wij willen dit realiseren door verschillende methodes van color-to-grayscale te zoeken en testen, en daaruit de betere te selecteren.

Algoritmes

We verzamelen een aantal algoritmes uit papers, en kiezen hier 3 uit die ons interessant lijken om te onderzoeken.

Intensity methode

Intensity methode is een simpel 'color-to-grayscale' algoritme. Het idee van dit algoritme is van iedere pixel het gemiddelde nemen van de RGB waarde. Om zo een hoger contrast te krijgen tussen donker en licht. Dan zou je dus ook beter de edges in de foto moeten kunnen zien.

$$\mathcal{G}_{Intensity} \leftarrow \frac{1}{3}(R + G + B).$$

Gleam

Bij het gebruik van de Intensity methode wordt de berekening van de lineaire RGB channels gebruikt, terwijl in de praktijk vaak de gammacorrectie intact wordt gelaten bij het gebruik van "gamma correcte" datasets. Dit wordt ook wel **Gleam** genoemd.

$$\mathcal{G}_{Gleam} = \frac{1}{3}(R' + G' + B').$$

Deze methode leek ons interessant, omdat deze methode nooit waardes hoger heeft dan de correcte gamma intensity.

$$\mathcal{G}_{Intensity} \leq \mathcal{G}_{Gleam} \leq \Gamma(\mathcal{G}_{Intensity}).$$

Luminance

Deze methode leek ons interessant om te proberen omdat hier niet meer naar de menselijke perceptie van helderheid wordt gekeken maar dit wel probeert te bereiken. Luminance wordt volgens ons literatuuronderzoek veel gebruikt bij “standaard” image processing software en het leek ons daarom interessant om ook deze te bekijken. Het nadeel is dat deze kennelijk veel gebruikt wordt, dus misschien gaan we nu iets onderzoeken wat al meermalen onderzocht is.

$$\mathcal{G}_{Luminance} \leftarrow 0.3R + 0.59G + 0.11B.$$

Lightness

Deze methode leek ons interessant om te onderzoeken in combinatie met Luminance. Ons literatuuronderzoek wees namelijk uit dat Luminance de foto kan verbeteren voor mensen en lightness zou de foto ook beter zichtbaar moeten maken voor mensen. Door deze methode te gebruiken i.c.m. Luminance kunnen we dus mooi bekijken wat het beter zichtbaar maakt voor mensen. Dat is echter niet waar ons onderzoek nu om draait.

$$\mathcal{G}_{Lightness} \leftarrow \frac{1}{100} (116f(Y) - 16),$$

Value

Deze methode bestaat uit twee stappen en gebruikt ook gammacorrectie om te schalen. Het rekent eerst het maximum van de R-G-B waarden uit om daarna een gammacorrectie daarvan te doen (zie formules). Deze is interessant om te bekijken omdat hij uit meer stappen bestaat en dus wat ingewikkelder is. We kunnen dan gelijk kijken of het gebruiken van gamma op deze manier om te schalen ervoor zorgt dat gezichten beter herkenbaar worden voor computers.

$$\mathcal{G}_{Value} = \max(R, G, B). \quad \Gamma(\max(R, G, B)) = \max(R', G', B').$$

Gekozen algoritmen

We hebben uiteindelijk de volgende 3 methodes gekozen:

- Intensity methode

- Luminance
- Value

Intensity leek ons goed om te proberen om te onderzoeken omdat dit veruit de simpelste methode van grayscale is. Gewoon het gemiddelde. Die kunnen we dan dus gebruiken in de testresultaten om te zien of een simpel algoritme slechter is dan een meer ingewikkelde variant. Zoals het gevoel een beetje zegt.

Luminance hebben we gekozen omdat hij veel gebruikt wordt bij “standaard” software. Die is dus zinnig om te bekijken. Want dat betekent dat het makkelijk is om foto’s beter zichtbaar te maken voor recognition systemen

Value is de meer ingewikkelde die we vonden en maakt ook gebruik van gammacorrectie. Omdat hij daar ook gebruikt van maakt vonden we deze interessant. De andere twee maken daar geen gebruik van dus is het best interessant om te bekijken of het toepassen van gammacorrectie helpt bij het beter herkennen van gezichten.

Zo pakt deze methode het maximum van de gamma van de RGB waarden. Dat betekent dus dat overbelichte foto’s wellicht nog steeds overbelicht blijven, en wat voor effect heeft dat dan op de facial recognition?

Verder is de max functie makkelijk te vervangen door de min. Dus wat gebeurt er als we de min van die RGB waarden pakken? Wellicht als je de min pakt, worden overbelichte foto’s beter bruikbaar en als je de max pakt worden onderbelichte foto’s beter bruikbaar. Kortom, er is genoeg om te bekijken met deze methode.

Methode van testen

Om ons doel te onderzoeken gaan we gebruik maken van een open dataset met 10.000 foto’s van bekende mensen (zie bronnen). Omdat het doorlopen van ons testplan met 10.000 foto’s wat lang zou duren pakken we de eerste 50 foto’s uit deze map om het te proberen. Op deze map met 50 foto’s doen we het volgende:

1. Wij houden iedere foto zoals hij is, gebruiken het algoritme om hem te vergrijzen en noteren hoeveel procent van de gezichten wordt herkend in de foto’s (base case).
2. Wij verhogen de helderheid van de originele foto’s in drie opeenvolgende stappen met 5, 10 en 15 procent en doen dezelfde test als bij stap 1. Dus tot en met een helderheid van 15% hoger dan het origineel.
3. We gebruiken weer de originele foto maar dan gaan we hem in drie opeenvolgende stappen met 5, 10 en 15 procent verlagen en bekijken dan hoe ieder het algoritme het doet en noteren wederom het percentage.

Uiteindelijk houden we dan een set met data over die ons inzicht kan verschaffen in welk algoritme het beter doet bij overbelichting (en hoeveel overbelichting), welk algoritme het beter doet bij onderbelichting (en hoeveel onderbelichting) en welk algoritme in het midden uitkomt. Als we dit weten kunnen we enigszins een uitspraak doen welk algoritme mogelijk beter is om ons doel te bereiken.

Bronnen

- Kanan, C., & Cottrel, G. W. (2012, 10 januari). *Color-to-Grayscale: Does the Method Matter in Image Recognition?* Geraadpleegd op 21 februari 2020, van [Color-to-Grayscale: Does the Method Matter in Image Recognition?](#)
- [Large-scale CelebFaces Attributes \(CelebA\) Dataset](#) (open dataset met foto's)