

Implementatieplan Intensity

1 | Namen en datum

Utrecht, 5 februari 2019

Dylan Rakiman
Joost van Lingen

2 | Doel

Het doel van deze implementatie is het uitvinden of bepaalde methoden voor het omzetten van een RGB-image naar een Intensity (grayscale) image de accuraatheid van het herkennen van gezichten zou kunnen vergroten.

3 | Methoden

De methoden hebben we afgeleid uit een paper waarin verschillende formules staan om RGB naar Intensity waardes om te zetten (Kanan & Cottrell, 2012). Zie de bronnen onderaan het document.

Hier volgt de lijst van methoden zoals deze in het voorheen genoemde document te vinden zijn:

Methode 1: Intensity

$$\text{Brightness} = \frac{R+G+B}{3}$$

Methode 2: Gleam

$$\text{Brightness} = \frac{R'+G'+B'}{3}$$

Methode 3: Luminance (OpenCV standaard)

$$\text{Brightness} = 0,3 \cdot R + 0,59 \cdot G + 0,11 \cdot B$$

Methode 4: Luma

$$\text{Brightness} = 0,2126 \cdot R + 0,7152 \cdot G + 0,0722 \cdot B$$

Methode 5: Lightness

$$\text{Brightness} = \frac{116 \cdot f(Y) - 16}{100}$$

Where $Y = 0,2126 \cdot R + 0,7152 \cdot G + 0,0722 \cdot B$

$$f(t) = t^{\frac{1}{3}} \text{ for } t > \left(\frac{6}{29}\right)^3$$

$$f(t) = \frac{1}{3} \cdot \left(\frac{29}{6}\right)^2 \cdot t + \frac{4}{29} \text{ for } t \leq \left(\frac{6}{29}\right)^3$$

Methode 6: Value

Brightness = $\text{Max}(R, G, B)$

Methode 7: Luster

Brightness = $(\text{Max}(R, G, B) + \text{Min}(R, G, B)) / 2$

4 | Keuze

Methode 2 houden we buiten beschouwing omdat de implementatie hiervan hetzelfde is als die van methode 1, behalve dat hiervoor gamma-gecorrigeerde RGB waarden gebruikt worden. Aangezien het toepassen van gammacorrectie buiten de scope van dit onderzoek ligt gaan we hier verder niet op in.

Methode 3 wordt eigenlijk standaard gebruikt door OpenCV en staat bekend als 'RGB2Gray'. Deze methode wordt reeds gebruikt door het programma (Zo blijkt uit line 26 'cvtColor(image, image, CV_RGB2GRAY);' uit de file 'GrayscaleAlgorithm.cpp'), maar aangezien de implementatie hiervan triviaal gemakkelijk is nemen wij deze toch mee in onze code.

5 | Implementatie

Wij implementeren alleen code in de main en in de functie:

```
IntensityImage * StudentPreProcessing::stepToIntensityImage(const  
  RGBImage &image) const {}
```

Wij zullen alle methoden als aparte implementatie van deze functie in comments zetten met de bijbehorende naam erboven. Hierdoor kunnen wij één van deze methoden un-commenten en de rest in comment-brackets (`/*code*/`) zetten. Wij zullen door alle pixels van de RGBImage lopen en van deze pixel de waarden R, G en B volgens bovenstaande methoden omrekenen naar een Brightness (Intensity)-value. Deze value stoppen we dan weer in de Intensity-image shell klasse. Als wij klaar zijn met het lopen door alle pixels sturen we deze image shell klasse weer terug als return-value.

6 | Evaluatie

Wij zullen voor elke methode kijken of het executeren van de recognition-stappen bevorderd of verhinderd wordt. Dit doen wij door aan de stappen van recognition een nummer te hangen; volgens de executeSteps-functie in Main.cpp zijn er een totaal van 20 stappen te ondervinden voordat gezichtsherkenning correct plaats heeft gevonden. Door deze stappen te nummeren krijgen wij te zien tot welke stap in het herkennen van gezichten één bepaalde methode kan komen. Wij zullen vervolgens een testset van 10+ plaatjes nemen om te kijken hoe ver het programma komt in zijn herkenning. De hoeveelheid

stappen die de computer voltooid heeft voor alle plaatjes worden bij elkaar opgeteld en gedeeld door de hoeveelheid stappen maal de hoeveelheid plaatjes; hierdoor krijgen wij een percentage te zien van hoe ver het programma gemiddeld kan komen voor het herkennen van een plaatje. Als dit percentage hoger is, hebben wij een methode te pakken die beter werkt dan de huidige implementatie. Als dit percentage lager is, weten wij die specifieke methode te moeten vermijden.

7 | Bronnen

Kanan, C., & Cottrell, G. W. (2012). Color-to-Grayscale: PLoS ONE, *Color-to-Grayscale: Does the Method Matter in Image Recognition?*, 7(1). Doi: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029740>