# RGB to grayscale

## Naam en datum David Driessen 20/02/2020

## Doel

Ik wil rgb images om gaan zetten in intenstiets images. Dit wil ik gaan doen, omdat 3 kanaals(RGB) images moeilijker zijn te verwerken dan 1 kanaals(Gray scale) images. Dit wil ik gaan impelemteren in gezichts herkenning.

## Methoden

* (Intensity) Een simpelen reken som gebruiken. De gemiddelde waarde gebruiken van de 3 beschikbare kanalen. ((R+G+B)/3)  
  Deze methode is zeer makkelijk te implemeteren, is zeer snel , per pixel en is accuraat.
* (Intensity) De wortel nemen uit de kwadraaten van de rgb waarde die delen door 3 en daar de wortel van nemen. (Sqrt((R\*R+G\*G+B\*B)/3))  
  Deze methode is makkelijkte implemeteren, is snel , per pixel en is accuraat. (Min punt heeft sqrt nodig)
* (Glean) De afgeleide waardes van RGB bij elkaar optellen en dat delen door 3. ((R’ + G’+ B’)/3)  
  Deze methode is redelijke makkelijk te implemeteren, is redelijke snel , per pixel en is zeer accuraat. (Min punt neemt de afgeleide van een getal, dat kost extra tijd)
* (GLuminance) Vermedevuldigd de waarde tot 1 kanaal die geoptimaiseerd is voor menselijk waarneeming (0.3\*R + 0.59\*G + 0.11\*B)  
  Deze methode is makkelijk te implemeteren, isredlijke snel , per pixel en is redelijk accuraat.
* (GLuma) Bij deze methode wordt de afgeliede van RGB uitrekend en dan worden de waardes vermedevuldigd tot 1 kanaal die geoptimaiseerd is voor HDTVs (0,2126\*R’ + 0 ,7152\*G’ + 0,0722\*B’)  
  Deze methode is redelijke makkelijk te implemeteren, is redelijke snel , per pixel en is zeer accuraat. (Min punt neemt de afgeleide van een getal, dat kost extra tijd)
* (GLuster) Neemt de maximale waarde van RGB en telt daar de minimale waarde bij op en deelt het dan door 2. (max(rgb)+ min(rgb)/2)  
  Deze methode is redelijke makkelijk te implemeteren, is redelijke snel , over de hele image en is zeer accuraat. (Min punt eerst moet de maximalen en de minimalen waarde bepaald worden van een image, dat kost extra tijd)

## Keuze

Mijn keuze is om Intensity te gebruiken, want deze is zeer accuaart. Verder is deze methode zeer makkleijk te implemetren, is snel en rekent per pixel. De grafiek hier naast is een test met 4.000 gezichten in verschillende belichting en uitdrukking(AR). Hier uit blijkt dat Intensity een van de beste is voor het doeleinde.



## Implementatie

Je geeft aan hoe deze keuze is geimplementeerd in de code. Ik heb de Intensity methode((R+G+B)/3) geimplementeerd met volgende code:

IntensityImage \* img = ImageFactory::newIntensityImage();

img->set(image.getWidth(), image.getHeight());

int max = image.getWidth() \* image.getHeight();

for (int i = 0; i < max; i++) {

img->setPixel(i, (image.getPixel(i).R + image.getPixel(i).G + image.getPixel(i).B)/3);

}

return img;

## Evaluatie

Je geeft aan welke experimenten er gedaan zullen worden om de implementatie te testen en te ‘bewijzen’ dat de implementatie daadwerkelijk correct werkt. Dit geeft direct informatie over de meetrapporten die er zullen worden gemaakt.

De imageshell en de greyscale zullen allebei getest worden op snelheid, er zullen dus 2 meetrapotren overkomen