

1. RGB to grayscale

1.1. Naam en datum

David Driessen en Mike Hilhorst 5-4-17

1.2. Doel

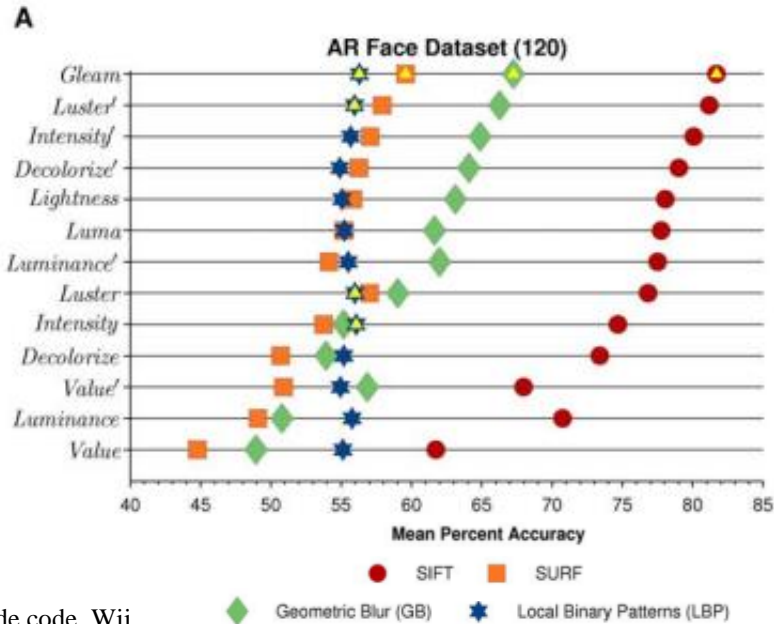
Wij willen rgb images om gaan zetten in intensiteits images, dit willen wij gaan doen omdat 3 kanaals(Gray) images moeilijker zijn te verwerken dan 1 kanaals(Gray) images. Dit willen wij gaan implementeren in gezichts herkenning.

1.3. Methoden

- (Intensity) Een simpel reken som gebruiken. De gemiddelde waarde gebruiken van de 3 beschikbare kanalen. $((R+G+B)/3)$
Deze methode is zeer makkelijk te implementeren, is zeer snel , per pixel en is accuraat.
- (Intensity) De wortel nemen uit de kwadraten van de rgb waarde die delen door 3 en daar de wortel van nemen. $(\sqrt{(R^2+G^2+B^2)/3})$
Deze methode is makkelijk te implementeren, is snel , per pixel en is accuraat. (Min punt heeft sqrt nodig)
- (Glean) De afgeleide waarden van RGB bij elkaar optellen en dat delen door 3. $((R' + G' + B')/3)$
Deze methode is redelijke makkelijk te implementeren, 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 geoptimaliseerd is voor menselijk waarneeming $(0.3*R + 0.59*G + 0.11*B)$
Deze methode is makkelijk te implementeren, isredlijke snel , per pixel en is redelijk accuraat.
- (GLuma) Bij deze methode wordt de afgeliende van RGB uitrekend en dan worden de waarden vermedevuldigd tot 1 kanaal die geoptimaliseerd is voor HDTVs $(0,2126*R' + 0,7152*G' + 0,0722*B')$
Deze methode is redelijke makkelijk te implementeren, 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. $(\max(\text{rgb}) + \min(\text{rgb})/2)$
Deze methode is redelijke makkelijk te implementeren, 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)

1.4.Keuze

Onze keuze is om Intensity te gebruiken, want deze is zeer accuaart. Verder is deze methode zeer makkelijk 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 onze doel einde.



1.5.Implementatie

Je geeft aan hoe deze keuze is geïmplementeerd in de code. Wij hebben de Intensity methode $((R+G+B)/3)$ geïmplementeerd 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;
```

1.6.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 meetrapporten overkomen