Implementatieplan Grayscaling

V2B-vision

Scott Timmermans & Sander Boot
17-2-2020

Doel

Het doel van deze implementatie is om een RGB foto om te zetten in een duidelijke foto in grayscale zonder te veel detail en contrast te verliezen. Hierdoor moet de face recognition een gezicht kunnen herkennen. Het doel is dat de accuratie van de totale pipeline hoger wordt dan die van de huidige implementatie. De accuratie van de pipeline is het percentage waarbij de pipeline een gezicht uit een foto haalt bij het testen met een dataset.

Methoden

Lightness

De lightness methode berekent het gemiddelde van de kleur met de hoogste intensiteit en de laagste intensiteit.: (max(R, G, B) + min(R, G, B)) / 2.

Deze methode zorgt voor een verlies in contrast en wordt dus niet vaak gebruikt.

Colorimetric (perceptual luminance-preserving) conversion to grayscale

Y'= 0.2126**R**' + 0.7152**G**' + 0.0722**B**'.

Je berekent de Y' voor R,G en B waardoor ze dezelfde waarde krijgen. Hierdoor wordt de image grayscale. Dit is de meest gebruikte methode en wordt ook gebruikt in programma's zoals Gimp als je de afbeelding omzet naar grayscale.

"*These three particular coefficients represent the intensity (luminance) perception of typical trichromat humans to light of the precise Rec. 709 additive primary colors (chromaticities) that are used in the definition of sRGB.*"

Luma coding

Verschillende factoren gebruikt in HDTV RGB

Y'= 0.299**R**' + 0.587**G**' + 0.114**B**'.

Y'= 0.2127**R**' + 0.6780**G**' + 0.0593**B**'.

Met deze methode wordt er per kleur een percentage van de intensiteit gebruikt voor het totaal.

Single channel greyscaling (average)

Je pakt per kleur de intensiteit en berekent dan het gemiddelde van de 3 waardes en dit wordt de nieuwe intensiteit voor de RGB waardes per pixel. : (R + G + B) / 3

Keuze

Het plan is om de "*Colorimetric (perceptual luminance-preserving) conversion to grayscale method*" te implementeren. Dit levert de beste resultaten op in een online image to grayscale converter. Het zou ook de meest gebruikte methode zijn in foto bewerking programma's. Met deze methode worden de details en het contrast goed behouden.

Om te checken of deze methode het beste werkt worden de andere methodes ook geïmplementeerd en vergeleken met de face recognition software.

Implementatie

```
IntensityImage *StudentPreProcessing::stepToIntensityImage(const RGBImage &image) const
{
    auto intesityImage = ImageFactory::newIntensityImage(image.getWidth(), image.getHeight());
    Intensity pixelIntensity = 0;

    char method = 2;  // Specify the method you want to use

    for (int x = 0; x < image.getWidth(); x++){
        for (int y = 0; y < image.getHeight(); y++){
            RGB CPixel = image.getPixel(x, y);

            if (method == 1){   // average method
                pixelIntensity = (CPixel.r + CPixel.g + CPixel.b) / 3;
            }
            else if (method == 2){  // colorimetric (perceptual luminance) method
                pixelIntensity = (Intensity)((double)(CPixel.r * 0.21) +
                                             (double)(CPixel.g * 0.72) +
                                             (double)(CPixel.b * 0.07));
            }
            else if (method == 3){  // lightness method
                pixelIntensity = (std::max(std::max(CPixel.r, CPixel.g), CPixel.b) +
                                  std::min(std::min(CPixel.r, CPixel.g), CPixel.b)) / 2;
            }

            intesityImage->setPixel(x, y, pixelIntensity);
        }
    }
    return intesityImage;
}
```

Door de method variabele aan te passen wordt een van de 3 geïmplementeerde methodes gekozen. Hiervoor is gekozen omdat de originele code dan zo min mogelijk wordt aangepast.

Om de Luma coding methode toe te passen kunnen de waarden van methode 2 aangepast worden.

Evaluatie

Om te evalueren welke methode werkt wordt er een test uitgevoerd met meerdere data sets die na het grayscalen door de face recognition software worden behandeld. De betrouwbaarheid van de resultaten zal getest worden door eerst alle foto's mét gezicht te testen, en daarna degene zonder gezicht. Daardoor is altijd duidelijk of het resultaat betrouwbaar is.

Door middel van een standaard afwijking te berekenen wordt er gekeken of er een significant verschil is met de Default methode van de docent ten opzichte van onze methodes.

De data set die gebruikt gaat worden bestaat uit foto's van gezichten. Op de site kaggle.com zijn heel veel verschillende soorten data sets te vinden waaronder degene die gebruikt wordt door ons. De dataset van kaggle.com die gebruikt gaat worden bestaat uit een lijst met echte gezichten en nep (fake) gezichten. In onze evaluatie gaan alleen de echte gezichten gebruikt worden.

De dataset die van kaggle.com komt is 1081 foto's groot. Deze wordt opgedeeld in datasets van 100 om zo een standaardafwijking te kunnen berekenen tussen de 10 resultaten.

Bibliografie

Computational Intelligence and Photography Lab, Y. U. (2019). *Real and Fake Face Detection*. Opgehaald van Kaggle: https://www.kaggle.com/ciplab/real-and-fake-face-detection/data

Cook, J. (2009, Augustis 24). *Three algorithms for converting color to grayscale*. Opgehaald van johndcook.com: https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

Poynton, C. (1997, Juli 15). *The magnitude of nonconstant luminance errors*. Opgehaald van poynton.ca: https://poynton.ca/PDFs/Mag_of_nonconst_luminance.pdf

Stokes, M., Anderson, M., Chandrasekar, S., & Motta, R. (1996, November 5). *A Standard Default Color Space for the Internet - sRGB*. Opgehaald van w3.org: https://www.w3.org/Graphics/Color/sRGB

Timmermans, S., & Boot, S. (2020, Maart 6). *HU-Vision-1920-Sander-Scott*. Opgehaald van Github: https://github.com/KingPungy/HU-Vision-1920-Sander-Scott