

# 1. Image scaling

## 1.1. Namen en datum

Willem de Groot & Henrike Kraan-Bos, 28-02-2018

## 1.2. Doel

De face recognition applicatie maakt gebruik van images die ongeveer 40000 pixels groot zijn, bij een vierkante afbeelding komt dit neer op 200 bij 200 pixels. We moeten een afbeelding, groter dan 40000 pixels, dus van zijn oorspronkelijke grootte naar de juiste grootte brengen door middel van image scaling. Daarbij willen we dat de originele ratio behouden blijft.

Hoe kunnen we de grootte van een image, door middel van image scaling, van zijn oorspronkelijke grootte terugbrengen naar ongeveer 40000 pixels waarbij de originele ratio behouden blijft?

## 1.3. Methoden

We zijn uitgekomen op de volgende methoden om een afbeelding omlaag te schalen (decimation).

### Box sampling

Bij box sampling wordt aan de hand van de verhouding in grootte tussen de originele afbeelding en de geschaalde afbeelding bepaald hoeveel en welke pixels overeenkomen met de pixels van de geschaalde afbeelding. Vervolgens kan de waarde van die nieuwe pixel op verschillende manieren worden bepaald:

- **Middelste pixel:** Bij deze methode kiezen we simpelweg de middelste pixel van de oorspronkelijke pixels.
- **Mediaan:** De methode zet de waardes van de beschouwde pixels op volgorde van grootte en kiest hieruit de middelste waarde.
- **Gemiddelde:** Hierbij bepalen we het gemiddelde van alle beschouwde pixels en dit wordt de nieuwe waarde.
- **Gaussian:** Hier bepalen we het gewogen gemiddelde van de beschouwde pixels, aan de hand van een Gaussian kernel.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

De voordelen en nadelen van de methoden hebben we opgenomen in onderstaande tabel (Szeliski, 2010).

	Middelste pixel	Mediaan	Gemiddelde	Gaussian
Voordelen	Eenvoudig te implementeren en snel.	Werkt goed tegen ruis.	Alle pixels worden meegenomen in de berekening.	Een Gaussian filter is niet voor soort noise geschikt.
Nadelen	Heeft veel last van aliasing.	Sorteren kost veel tijd.	Hij is minder snel als middelste pixel.	Edge detection kan moeilijker zijn omdat de image high frequency-details uitvlakt.

## 1.4. Keuze

Wij willen alle vier beschreven methoden gaan maken, dit moet goed te doen zijn, aangezien ze allemaal gebaseerd zijn op box sampling, het enige verschil is de keuze die vervolgens wordt gemaakt.

## 1.5. Implementatie

We hebben zes functies geschreven:

- stepScaleImage
- resize
- middlepixel
- medianpixel
- average
- gaussian

De stepScaleImage functie die geïmplementeerd moest worden, hierin wordt de image waarin het resultaat zal worden opgeslagen aangemaakt. We kijken vervolgens of de oorspronkelijke image groot genoeg is om te schalen en rekenen dan de ratio uit. Vervolgens roepen de resize functie aan.

In de resize functie maken we een box sample aan, op basis van de ratio. Vervolgens vullen we de box sample met pixels uit de oorspronkelijke image. Zodra onze box sample is gevuld, roepen we de functie van één van de vier methoden aan. Deze geeft een pixelwaarde terug, die schrijven we vervolgens naar ons resultaat image.

In middlepixel kiezen we de middelste pixel uit de aangeleverde box sample als waarde.

In medianpixel sorteren we de pixels op hun waarde en kiezen vervolgens de middelste waarde (de mediaan).

In average tellen we de waardes van alle pixels bij elkaar op en delen door het aantal pixels (het gemiddelde) en returnen we deze waarde.

In gaussian berekenen we de Gaussian factor voor iedere pixel, vermenigvuldigen die met de pixelwaarde, tellen dit op bij het totaal en delen dat totaal vervolgens door het totale gewicht. Deze waarde geven we dan terug.

## 1.6. Evaluatie

We hebben 3 test images die door middel van alle vier de methoden geschaald worden, dit levert 12 geschaalde images op.

We gaan de vier methoden en de default methode testen op de snelheid van de methode en de kwaliteit van de resulterende image.

## **2. Bibliography**

Prasantha, H. &. (2009). Image scaling comparison using universal. *International Conference on Advances in Computing, Control, and Telecommunication Technologies* (pp. 859-863). Bangalore: PES Institute of Technology. doi:10.1109