

Labo 1

Bert De Saffel

11 februari 2019

1 Oplosmethode

De opgave is opgelost met behulp van Python. Als eerste deel is er een minimale vorm van argument checking, om na te gaan of het opgegeven argument wel degelijk een bestand is. De controle dat het een effectieve image is wordt overgelaten aan OpenCV zelf. Het opgegeven argument wordt gesplitst zodat de extensie en het pad apart beschikbaar zijn. Het pad wordt gebruikt om een venster aan te maken met behulp van `namedWindow`, en de bestandsextensie wordt later gebruikt om de image terug weg te schrijven.

De rest van de code is het stapsgewijs uitvoeren van de labo-opdracht, met behulp van de functies in volgend onderdeel.

2 Gebruikte functies

- **`namedWindow(winname[, flags])`** → **None**

Maakt een venster aan met als naam de *winname* parameter. De optionele *flags* parameter geeft extra eigenschappen aan het venster. Deze functie heeft geen returntype, maar het aangemaakte venster kan wel aangesproken via de opgegeven *winname* bij andere functies. Een eenvoudige keuze van *winname* is de naam van het opgegeven bestand zelf.

- **`imread(filename[, flags])`** → **retval**

Probeert een foto te laden met het opgegeven *filename*. Indien de *filename* op het bestandssysteem staat wordt een referentie teruggeven naar een *numpy.ndarray* object, een multidimensionale array die de eigenlijke image voorstelt. De optionele *flags* parameter laat voornamelijk toe om kleurenconversies uit te voeren op de image. De default waarde is `IMREAD_UNCHANGED`, die aangeeft dat de image onveranderd blijft.

- **`imshow(winname, mat)`** → **None**

Zal een image, meegegeven met de *mat* parameter, tonen in het venster met als naam *winname*.

- **`imwrite(filename, img[, params])`** → **retval**

Zal een image, meegegeven met de *img* parameter, opslaan naar een specifiek bestand, meegegeven met de *filename* parameter. De optionele *params* parameter bevat opties in de vorm van paren (`paramId1, paramValue1, paramId2, paramValue2, ...`). Deze opties hebben betrekking tot onder andere compressie en kwaliteit van de image.

- **`cvtColor(src, code[, dst[, dstCn]])`** → **dst**

Converteert een image van een kleurenruimte naar een andere. De *src* parameter bevat de input image. De *code* parameter bevat de conversieregel die gebruikt moet worden. In dit

labo wordt de code `COLOR_BGR2GRAY` gebruikt, om een kleurenbeeld in grijsschaal om te zetten. De conversie van een kleurenpixel naar een grijspixel kan door de de som te nemen van de individuele kleurencomponenten, elk vermenigvuldigt met een constante:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

waarbij Y de intensiteit van een willekeurige pixel voorstelt voor de grijsschaal image, en R , G en B de kleurencomponenten die horen tot die pixel van de originele image.

Deze functie kan op twee manieren de grijsschaal image teruggeven: enerzijds via de returnwaarde *dst* en anderzijds door de referentieparameter *dst*. De eerste methode is handig voor als er nog geen image object beschikbaar is, of als er niet gewenst is om een bestaande image te overschrijven.

- **threshold(src, thresh, maxval, type[, dst]) → retval, dst**

Past een threshold toe op de *src* image. In dit labo wordt het *type* ingesteld op `THRESH_BINARY`. Dit komt neer op de stuksgewijze functie:

$$dst(x, y) = \begin{cases} maxval & , src(x, y) > thresh \\ 0 & , src(x, y) \leq thresh \end{cases}$$

Om een zwart-wit image te genereren vanuit een grijsschaal image wordt $maxval = 255$ en $thresh = 127$ genomen. Voor een pixelintensiteit groter dan 127 wordt de intensiteit 255, wat overeenkomt met een witte pixel. Anderzijds voor een pixelintensiteit kleiner of gelijk dan 127 wordt de intensiteit 0, wat overeenkomt met een zwarte pixel.

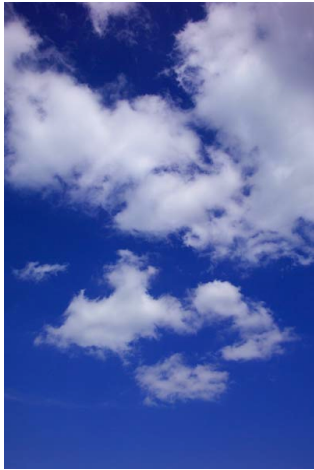
Net zoals de functie *cvtColor* heeft de *threshold* functie dezelfde twee manieren om de uitvoer op te vangen. De *retval* returnwaarde is enkel van belang indien `THRESH_OTSU` of `THRESH_TRIANGLE` gekozen wordt als waarde voor de parameter *type*.

- **waitKey([, delay]) → retval**

Deze functie wacht tot dat de gebruiker een knop heeft ingedrukt op zijn toetsenbord. Vaak wordt deze functie gebruikt in combinatie met de *imshow* functie, aangezien anders de getoonde foto niet tevoorschijn komt. Het is ook mogelijk om een *delay* op te geven, in milliseconden, die aangeeft hoe lang de functie zal wachten op antwoord van de gebruiker. Een $delay \leq 0$ zal oneindig lang wachten. De returnwaarde van deze functie is de code die overeenkomt met de ingedrukte toets.

3 Invoer en uitvoer

Als testimage wordt de foto *clouds.png* gebruikt. Figuur 1, 2 en 3 tonen respectievelijk de originele image, de grijsschaalimage en de binaire image.



Figuur 1: De originele, onaangepaste image.



Figuur 2: Figuur 1 geconverteerd naar grijsschaal.



Figuur 3: Binaire thresholding uitgevoerd op figuur 2.