

1

GROEIDOCUMENT - COMPUTERVISION

NAAM: LUC VAN DER SAR & RUBEN CLAASSEN
STUDENTEN NUMMERS: 2197154 & 2209414 OPLEIDING: TECHNISCHE INFORMATICA
PERIODE: 4
JULY 1, 2024

Contents

1 Practicum 1	2
1.1 Opdracht 1	2
1.1.1 Wat is het doel van de klasse Mat?	2
1.2 Opdracht 2	2
1.2.1 Resize / Crop	2
1.2.2 wat wordt onder de ROI van een image verstaan	3
1.3 Opdracht 3	3
1.3.1 Circle, rectangle, line, puttext	3
1.4 Opdracht 4	4
1.4.1 Warping	4
2 Practicum 2	4
2.1 Opdracht 1	5
2.1.1 Filters	5
2.1.2 c & d	5
2.1.3 e	6
2.1.4 f & g	7
2.2 Opdracht 2	8
2.2.1 Filters	8
2.2.2 b & c	9
2.3 Opdracht 3	10
2.3.1 binaire morfologie	10
2.3.2 c	11
2.3.3 d	11
2.3.4 e	11
2.4 opdracht 4	12
2.4.1 a	12
2.4.2 b	12
2.4.3 c	12
3 Practicum 3	12
3.1 Opdracht 1	12
3.1.1 A	12
3.1.2 B	14
3.2 Opdracht 2	14
3.2.1 Face detection	14

1 Practicum 1

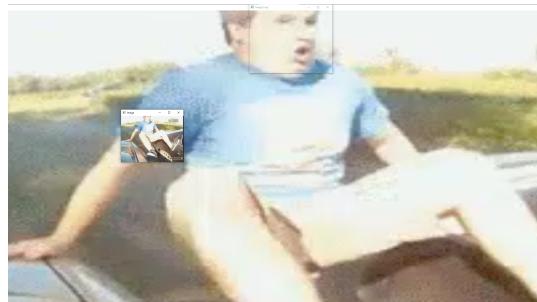
1.1 Opdracht 1

1.1.1 Wat is het doel van de klasse Mat?

Mat wordt gebruikt voor images. In een Mat object worden vectoren en matrixen opgeslagen van een color of greyscale afbeelding die vervolgens bewerkt of uitgelezen kan worden.

1.2 Opdracht 2

1.2.1 Resize / Crop



Source ivbm Copyright

```
#include <iostream>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
using namespace cv;
using namespace std;
#ifndef _DEBUG
#pragma comment( lib, "opencv_world490d")
#else
#pragma comment( lib, "opencv_world490" )
#endif

void main() {

    string path = "Resources/johannusthemannus.png";
    Mat img = imread(path);
    Mat imgResize, imgCrop;

    //resizes the image to 4k resolution
    resize(img, imgResize, Size(3840,2160));

    //sets the area of the crop and crops it to these values
    Rect roi(1500, 0, 400, 300);
    imgCrop = imgResize(roi);

    //makes 3 windows with all the images
    imshow("Image", img);
    imshow("Image Resize", imgResize);
    imshow("Image Crop", imgCrop);
    waitKey(0);
}
```

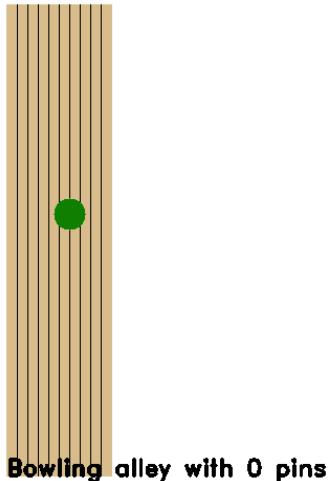
[Github](#)

1.2.2 wat wordt onder de ROI van een image verstaan

Een sectie van een image waar je meer mee wilt. Een ROI staat voor "Region of Interest" door een crop toe te passen kan je focus leggen op een specifieke sectie van het image.

1.3 Opdracht 3

1.3.1 Circle, rectangle, line, puttext



```
#include <iostream>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>

using namespace cv;
using namespace std;

void main() {

    // Blank Image
    Mat img(512, 512, CV_8UC3, Scalar(255, 255, 255));

    //Draws different shapes with set colors

    rectangle(img, Point(200, 50), Point(300, 500), Scalar(139, 188, 216), FILLED);

    for (int x = 210; x < 300; x += 10) {
        line(img, Point(x, 50), Point(x, 500), Scalar(0, 0, 0), 1);
    }
    circle(img, Point(260, 250), 15, Scalar(0, 127, 14), FILLED);

    //writes text with specific Font and Color
    putText(img, "Bowling alley with 0 pins", Point(200, 500), FONT_HERSHEY_DUPLEX, 0.75, Scalar(0, 0, 0));

    imshow("Image", img);
    waitKey(0);
}
```

[Github](#)

1.4 Opdracht 4

1.4.1 Warping



```
#include <iostream>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>

using namespace cv;
using namespace std;

////////////////// Warp Images //////////////////////

void main() {

    string path = "Resources/laptop.jpg";
    Mat img = imread(path);
    Mat matrix, imgWarp, imgResize;
    float w = 500, h = 350;

    //sets 4 points on the original image
    Point2f src[4] = { {102,700},{2489,194},{690,2445},{3800,1075} };

    //sets 4 destination points from 0,0
    Point2f dst[4] = { {0.0f,0.0f},{w,0.0f},{0.0f,h},{w,h} };

    //warps the 4 points of the original to the destination points
    matrix = getPerspectiveTransform(src, dst);
    warpPerspective(img, imgWarp, matrix, Point(w, h));

    //resizes the image to 1080p
    resize(img, imgResize, Size(1990, 1080));

    imshow("Image", imgResize);
    imshow("Image Warp", imgWarp);
    waitKey(0);
}
```

[Github](#)

2 Practicum 2

2.1 Opdracht 1

2.1.1 Filters

Operators: cvtColor() ; GaussianBlur()

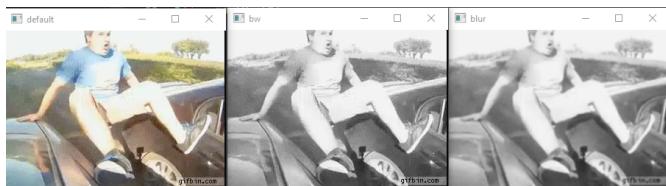
- cvtColor();

- o Wordt gebruikt om een afbeelding van de ene kleurruimte naar de andere te converteren
- o parameters
 - src : de bron afbeelding die je gebruikt als start afbeelding
 - dst : de afbeelding waar je vervolgens het veranderde bestand in op slaat.
 - code : de kleurruimte conversiecode, dit specificeert het type conversie dat je toepast
 - stCn : de hoeveelheid kanalen die gebruikt worden. als dit op 0 staat wordt dat automatisch bepaald.
- o Je zou deze operator kunnen gebruiken voor het converten naar grey scale, dit wordt vaak gedaan bij edge detection.

- GaussianBlur();

- o Wordt gebruikt om een afbeelding door middel van een Gaussian Blur te blurren. Deze blur gebruikt een Gauss filter, dit is een filter die in het midden een hoog getal is en langzaam naar beneden gaat hoe verder van het midden de pixel af is
- o wat de betekenis is van de parameters
 - src : de bron afbeelding die je gebruikt als start afbeelding
 - dst : de afbeelding waar je vervolgens de veranderde afbeelding in opslaat
 - ksize : de grootte van de gauss kernel
 - sigmaX : dit is de standaarddeviatie in de x richting voor de Gauss kernel
 - sigmaY : dit is de standaarddeviatie in de y richting van de Gauss kernel, wordt deze op 0 gezet dan is het dezelfde waarde als sigmaX
 - borderType : bepaalt het type pixel pooling voor de borderpixels.
- o Je zou deze operator kunnen gebruiken voor het vervagen van een foto om als doel ruis te verminderen.

2.1.2 c & d



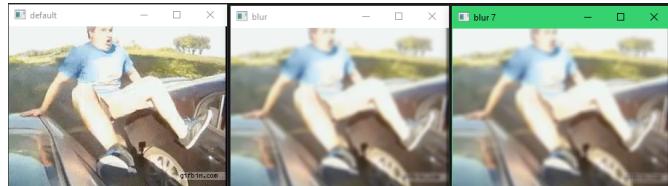
```
void main() {
    string path = "Resources/Johannusthemannus.png";
    Mat img = imread(path);
    Mat bw;
    Mat blur;

    //Changes the color space to Gray for a black and white look
    cvtColor(img, bw, COLOR_BGR2GRAY);

    //Gives a blur effect over the black and white img with a size of a 3 by 3 matrix
    GaussianBlur(bw, blur, Size(3, 3), 3, 0);
    imshow("default", img);
    imshow("bw", bw);
    imshow("blur", blur);
    waitKey(0);
}
```

[Github](#)

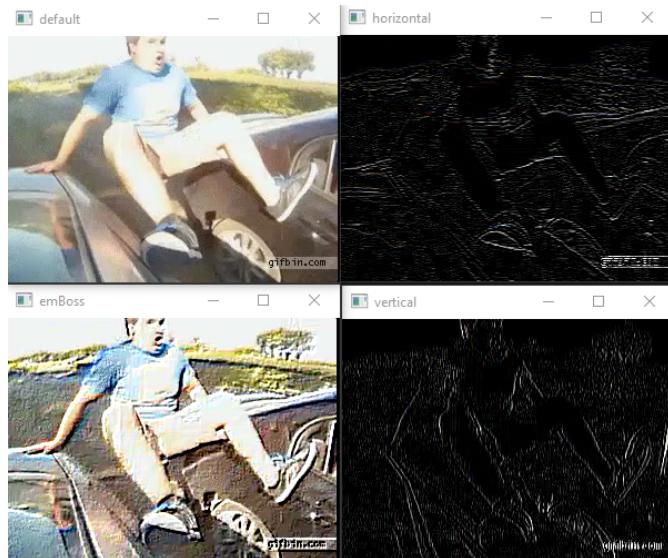
2.1.3 e



```
void main() {  
  
    string path = "Resources/Johannusthemannus.png";  
    Mat img = imread(path);  
    Mat blur;  
    Mat blur7;  
  
    //Gives a blur effect over the black and white img with a size of a 7 by 7 matrix  
    GaussianBlur(img, blur, Size(7, 7), 3, 0);  
    //Adds the same blur effect but this time with a bigger Sigma which results in a Higher Blur result  
    GaussianBlur(img, blur7, Size(7, 7), 7, 0);  
  
    imshow("default", img);  
    imshow("blur", blur);  
    imshow("blur 7", blur7);  
    waitKey(0);  
}
```

Github

2.1.4 f & g



```

void main() {

    string path = "Resources/Johannusthemannus.png";
    Mat img = imread(path);
    Mat horizontalimg;
    Mat verticalimg;
    Mat emboss;

    //sets a matrix for a horizontal line filter
    Mat horizontal = (Mat_<char>)(3, 3) << -1, -1, -1,
                                    2, 2, 2,
                                    -1, -1, -1;

    //sets a matrix for a vertical line filter
    Mat vertical = (Mat_<char>)(3, 3) << -1, 2, -1,
                                    -1, 2, -1,
                                    -1, 2, -1;

    //sets a matrix to highlight the difference in pixel intensity
    Mat embossFilter = (Mat_<char>)(3, 3) << -2, -1, -0,
                                    -1, 1, 1,
                                    0, 1, 2;

    //sets all the filters to their respectable img
    filter2D(img, emboss, 0, embossFilter);
    filter2D(img, horizontalimg, 0, horizontal);
    filter2D(img, verticalimg, 0, vertical);

    imshow("default", img);
    imshow("emBoss", emboss);
    imshow("vertical", verticalimg);
    imshow("horizontal", horizontalimg);
    waitKey(0);
}

```

[Github](#)

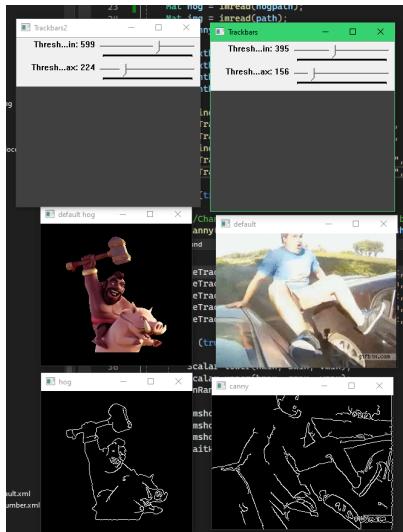
2.2 Opdracht 2

2.2.1 Filters

Operator Canny();

- Canny();
 - wordt gebruikt voor het uitlijnen van alle omtrekken in een foto van objecten
 - parameters
 - image : de bron afbeelding die je gebruikt als start afbeelding
 - edges : de afbeelding waar je vervolgens het veranderde bestand in op slaat.
 - threshold1 : weak edge detection, alles dat onder deze threshold valt wordt niet getekend.
 - threshold2 : strong edge detection. alles wat boven deze threshold valt wordt getekend.
 - apertureSize : aperture size for the Sobel operator
 - L2Gradient : een checkmark om te bepalen of een L2 norm gebruikt moet worden of de standaard L1 norm
 - Je zou deze operator kunnen gebruiken om de uitlijning van bepaalde objecten weer te geven.

2.2.2 b & c



```

void main() {

    string path = "Resources/Johannusthemannus.png";
    string hogpath = "hog rider.png";
    Mat hog = imread(hogpath);
    Mat img = imread(path);
    Mat canny, hogcanny;

    int maxthreshold = 0;
    int maxthreshold2 = 0;
    int minthreshold = 0;
    int minthreshold2 = 0;

    //makes trackbars in separate windows 2 bars for each picture
    namedWindow("Trackbars", (640, 200));
    createTrackbar("Threshold Min", "Trackbars", &minthreshold, 1000);
    createTrackbar("Threshold Max", "Trackbars", &maxthreshold, 1000);
    namedWindow("Trackbars2", (640, 200));
    createTrackbar("Threshold Min", "Trackbars2", &minthreshold2, 1000);
    createTrackbar("Threshold Max", "Trackbars2", &maxthreshold2, 1000);

    while (true)
    {
        //takes the outline of the pictures with the set thresholds
        Canny(img, canny, maxthreshold, minthreshold);
        Canny(hog, hogcanny, maxthreshold2, minthreshold2);

        imshow("default", img);
        imshow("default hog", hog);
        imshow("hog", hogcanny);
        imshow("canny", canny);
        waitKey(1);
    }
}

```

[Github](#)

2.3 Opdacht 3

2.3.1 binaire morfologie

Image operators: dilate() ; erode() ; getStructureElement()

- dilate()

- o wordt gebruikt om de witte pixels van een afbeelding te verdikken
- o parameters
 - src : de bron afbeelding die je gebruikt als start afbeelding
 - dst : de afbeelding waar je vervolgens het veranderde bestand in op slaat.
 - kernel : de matrix waarmee de lijn vergroot wordt, dit zijn altijd oneven getallen
 - anchor : Zet een anker op de kernel, negatieve waarde zet het anker in het midden
 - iterations : aantal keer dat de dilation gedaan wordt
 - borderType : het type pixel pooling dat gebruikt wordt
 - borderColor : voor het geval je een constante border hebt, de default is magisch.
- o Je zou deze operator kunnen gebruiken om gaten in edge detectie te vullen.

- erode()

- o wordt gebruikt om de witte pixels van een afbeelding te verdunnen
- o parameters
 - src : de bron afbeelding die je gebruikt als start afbeelding
 - dst : de afbeelding waar je vervolgens het veranderde bestand in op slaat.
 - kernel : de matrix waarmee de lijn vergroot wordt, dit zijn altijd oneven getallen
 - anchor : Zet een anker op de kernel, negatieve waarde zet het anker in het midden
 - iterations : aantal keer dat de dilation gedaan wordt
 - borderType : het type pixel pooling dat gebruikt wordt
 - borderColor : voor het geval je een constante border hebt, de default is magisch. te bepalen of een L2 norm gebruikt moet worden of de standaard L1 norm
- o Je zou deze operator kunnen gebruiken om dikke uitleiningen in een afbeelding te kunnen verdunnen.

- `getStructureElement()`
 - o Maakt een matrix aan de hand van de parameters die je mee geeft, in de vorm van een kernel.
 - o parameters
 - shape : een enum value uit MorphShapes
 - ksize : de groote van het structuring element
 - anchor : Zet een anker op de kernel, negatieve waarde zet het anker in het midden
 - o Je zou deze operator kunnen gebruiken om een kernel te maken die vervolgens gebruikt kan worden in het dilate en erode process

2.3.2 c

Een opening operation erode eerst de afbeelding om het daarna meteen te dilaten. Dit zorgt ervoor dat in het voorbeeld de kleinen balken eerst verdikt worden, omdat een erode de witte pixels verkleint. bij de dilate zouden de witte pixels weer vergroot worden en zal er dus niks veranderen aan de afbeelding.

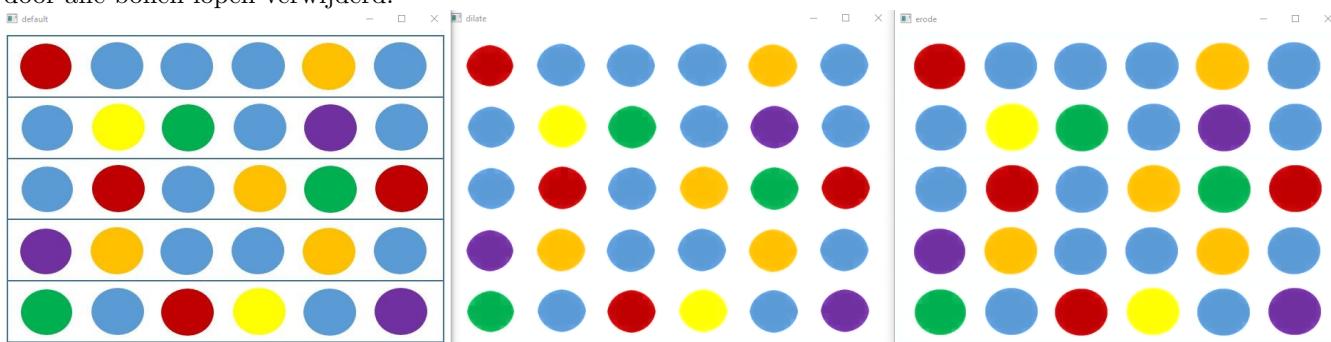
Vanwege het gebrek aan Vision Lab kunnen wij de .jl file niet inladen en hebben we dus geen afbeelding van het resultaat nadat er een opening operatie is uitgevoerd.

2.3.3 d

De afbeelding aan de rechter kant kan gecreëerd worden door eerst een dilate toe te passen die alle witte pixels verdikt, hierdoor zullen de dunne zwarte strepen overgenomen worden door de verdikte witte pixels. Vervolgens kan er een erode toegepast worden om de zwarte bollen weer te vergroten (de witte pixels eromheen worden verduld)

2.3.4 e

Wij konden geen manier vinden om ballstripes van een .jl file naar een jpg file om te zetten. wij hebben er dus voor gekozen om een soort gelijke afbeelding te gebruiken. In het voorbeeld hieronder hebben we de dunne lijnen die door alle bollen lopen verwijderd.



```
void main() {
    string path = "Resources/kleurenpatroon1.jpg";
    Mat img = imread(path), dilateimg, erodeimg;

    Mat kernel = getStructuringElement(MORPH_RECT, Size(7, 7));

    dilate(img, dilateimg, kernel);
    erode(dilateimg, erodeimg, kernel);

    imshow("default", img);
    imshow("dilate", dilateimg);
    imshow("erode", erodeimg);
    waitKey(0);
}
```

[Github](#)

2.4 opdracht 4

2.4.1 a

De saturation hoeft bij deze afbeelding alleen aangepast te worden wanneer de hue van de gele kleuren bekeken wordt. Dit is namelijk ook een beetje de kleur van de doos. Als de minimale saturation dan iets omhoog wordt gezet dan zal de doos wegvalLEN. Maar verder heeft het niet echt effect.

2.4.2 b

- gele ballen : hue level 18-28
- groene ballen : hue level 30-80
- blauwe ballen : hue level 85-112
- paarse ballen : hue level 133-147
- rood/roos : hue level 160 - 180

2.4.3 c

De hoofdkleuren die goed en duidelijk te onderscheiden zijn, zijn Rood, Groen en Blauw

3 Practicum 3

3.1 Opdracht 1

3.1.1 A

Functies: findContours() ; drawContours() ; contourArea() ; arcLength(); boundingRect(); approxPolyDP()

- findContours()
 - Deze functie geeft een lijst met contouren terug van een binaire afbeelding
 - parameters
 - image : de bron afbeelding die je gebruikt als start afbeelding
 - contours : Een lege vector van vectors om te vullen met contouren. Elke contour wordt opgeslagen als een vector van punten.
 - hierarchy : optionele output vector die informatie bevat van afbeelding topologie, dit geeft de volgorde van de contouren aan.
 - mode : Contour retrieval mode, gekozen uit standaard modus van OpenCV.
 - method : Contour approximation method, gekozen uit standaard modus van OpenCv. RETR_EXTERNAL selecteert alleen extreme outer contours in volgorden van hierarchy
 - offset : een optionele offset waarbij elke contour wordt verschoven.

- drawContours()
 - o Deze functie tekent de contouren in een afbeelding
 - o parameters
 - image : de bron afbeelding die je gebruikt als start afbeelding
 - contours : Een vector van vectors met contouren die getekend worden
 - contourIdx : Een index van welke contour getekend moet worden, negatief tekent ze allemaal
 - color : De kleur van de contouren
 - thickness : De dikte van de contouren
 - lineType : type lijn die gebruikt wordt
 - hierarchy : optionele informatie van de hierarchy, wordt alleen gebruikt als alleen sommige contouren getekend moeten worden.
 - maxLevel : optionele maximaal aantal contours die getekend kunnen worden, staat standaard op Max.
 - offset : optionele contour shift. schuift alle contours met een gegeven hoeveelheid.
- contourArea()
 - o Deze functie geeft de oppervlakte terug binnen de contouren, in pixelaantallen
 - o parameters
 - contour : input vector van contour vectoren.
 - oriented : optioneel. staat standaard op false, wordt gebruik om aan te geven dat het in een juiste orientatie staat. true wordt gebruikt voor contouren die clockwise zijn weer gegeven en false voor contouren die
- arcLength()
 - o berekend de booglengte van een contour
 - o parameters
 - curve : Een vector van 2D punten
 - closed : een flag die aan geeft of de curve gesloten is of niet.
- boundingRect()
 - o Deze functie geeft een Rectangle terug die perfect om de punten van de input array valt.
 - o parameters
 - array : een shape of een lijst met punten.
- approxPolyDP()
 - o Deze functie geeft een curve met dezelfde lengte als de originele curve maar bestaande uit minder vectoren. Dit kan gebruikt worden om het aantal hoekpunten van een contour terug te krijgen.
 - o parameters
 - curve : Een vector van 2D punten
 - approxCurve : een lege vector van 2D punten om te vullen
 - epsilon : Een parameter die de maximale afstand die mag zitten tussen de originele en de nieuwe curve
 - closed : als deze op true staat dan is de nieuwe curve gesloten(eerste en laatste vectoren zijn verbonden), zo niet dan is hij open

3.1.2 B

1. **Laad een afbeelding met objecten:** Laad de afbeelding waarin je objecten wilt identificeren.
2. **Zet de afbeelding om naar grijstinten:** Zet de afbeelding om naar grijstinten om de verwerking eenvoudiger te maken.
3. **Maak de afbeelding een beetje wazig (vervagen) om ruis te verminderen:** Vervaag de afbeelding om kleine ruis en details te verwijderen die de randen detectie kunnen beïnvloeden.
4. **Gebruik de Canny-methode om randen (contouren) in de afbeelding te vinden:** Gebruik de Canny-randdetectiemethode om de randen van de objecten in de afbeelding te detecteren.
5. **Maak de randen dikker door ze uit te dijen:** Gebruik een dilatatie-operatie om de gedetecteerde randen dikker te maken, wat helpt bij het vinden van volledige contouren.
6. **Bepaal de contouren van de objecten in de afbeelding:** Vind de contouren van de objecten in de afbeelding.
 - Zoek de contouren in de afbeelding.
 - Loop door elke gevonden contour.
7. **Bepaal de oppervlakte van elk object:** Bereken de oppervlakte van elke gevonden contour.
8. **Sla alleen contouren op van de objecten met een oppervlakte boven de 1000 pixels en ga daarmee verder:** Filter de contouren zodat alleen die met een oppervlakte groter dan 1000 pixels overblijven.
9. **Approximeer de contouren om ze simpeler te maken:** Vereenvoudig de contouren door minder punten te gebruiken, terwijl de vorm behouden blijft.
10. **Bepaal het type object (driehoek, vierkant, rechthoek, cirkel) en teken het op de afbeelding:** Identificeer het type object door het aantal hoeken te tellen en teken de contouren en type object op de afbeelding.
11. **Toon de bewerkte afbeelding:** Toon de bewerkte afbeelding met de getekende contouren en objecttypen.

3.2 Opdracht 2

3.2.1 Face detection

Klasse: CascadeClassifier; Funcites: load() ; detectMultiScale()

- load()
 - laad een classifier van een file
 - parameters
 - filename : pad van de file waarin de classifier is geladen
- detectMultiScale()
 - detecteert objecten van verschillende groottes van een input afbeelding. Deze worden terug gegeven als een lijst van rechthoeken
 - parameters
 - image : de bron afbeelding die je gebruikt als start afbeelding
 - objects : Een lege vector van Rect, deze kan gebruikt worden om te tekenen.
 - scaleFactor : specificeert met hoe veel de afbeelding wordt verkleind met elke schaal
 - minNeighbors : geeft aan hoeveel neighbors een rechthoek nodig heeft om te blijven bestaan.
 - minSize: minimale grootte van een object, kleinere objecten worden genegeerd
 - maxSize: maximale grootte van een object, grotere objecten worden genegeerd