

Assignment 4 - INFOIBV: Object Detection

Sylvie J Steward & Daan van den Berg
8568197 & 6836321

For this assignment, we will be detecting bike/bicycle wheels. We will be attempting this with clear images, but also images which have more noise or where bikes could be slightly tilted, but also in different lighting.

Criterium	Possible values
Minimum/maximum size	All/Minimum size/Maximum size
Lighting variations	Indoors/Outside/Night/Daylight/Stock image
Rotation variations	Straight/In motion/Slightly tilted/Turned
Occlusion	None/Partial/More than half
Other	No noise/Specific color/Object viewed frontally

Image 1

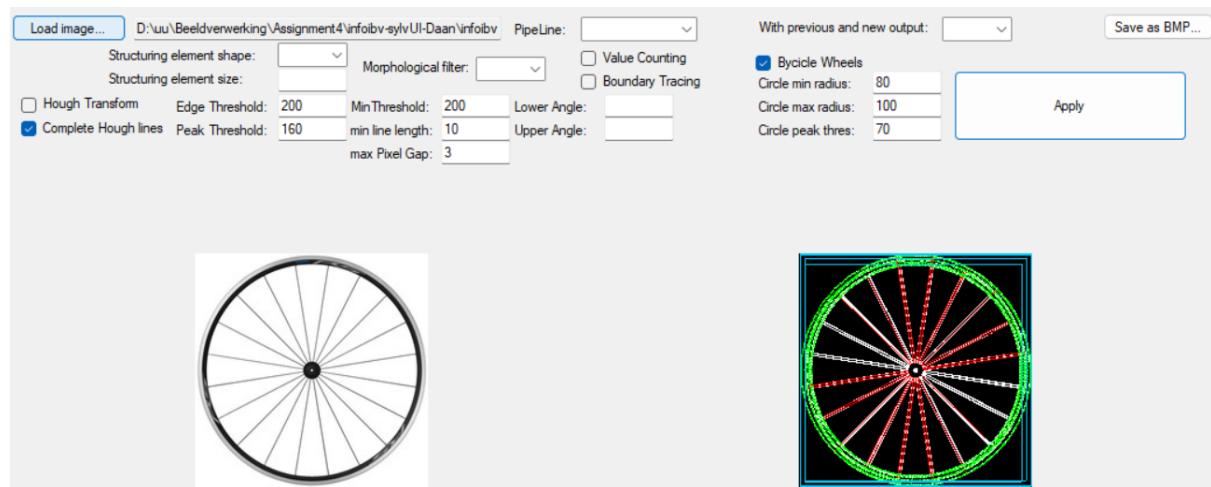


Image 2

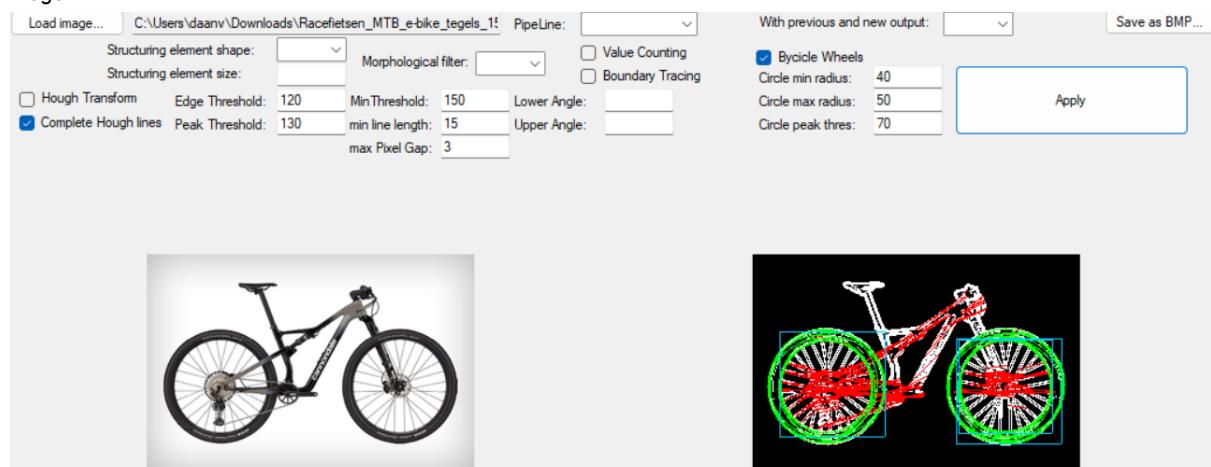


Image 3

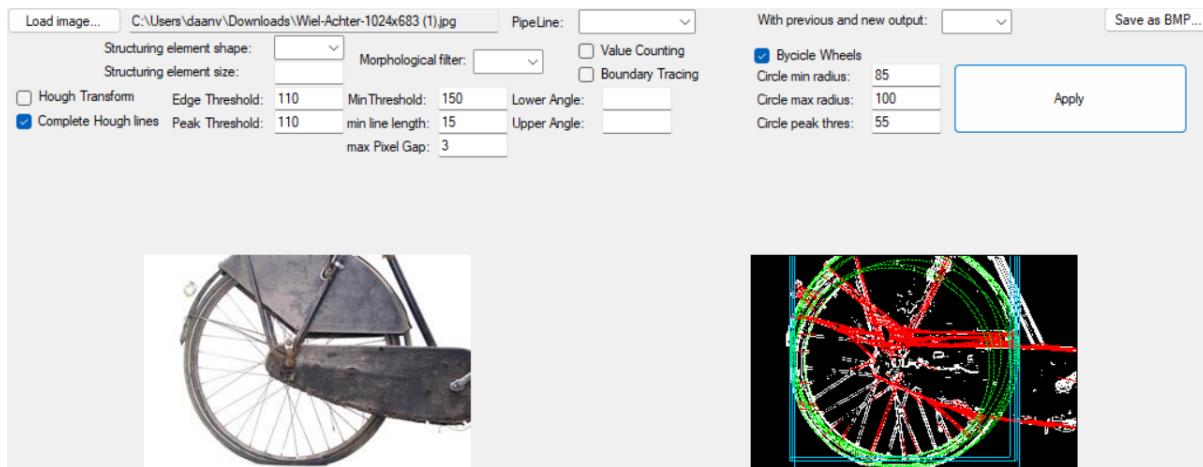


Image 4

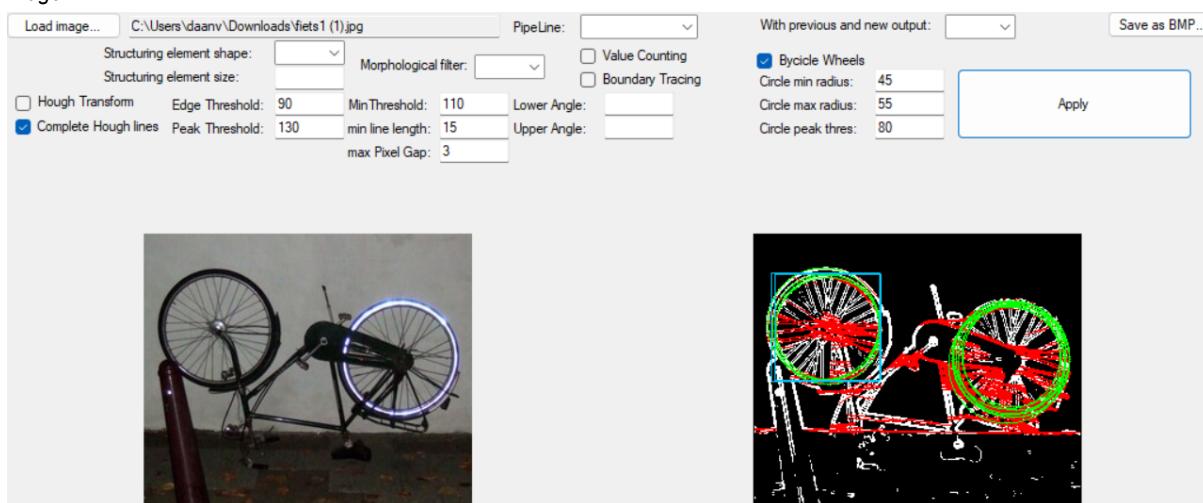


Image 5

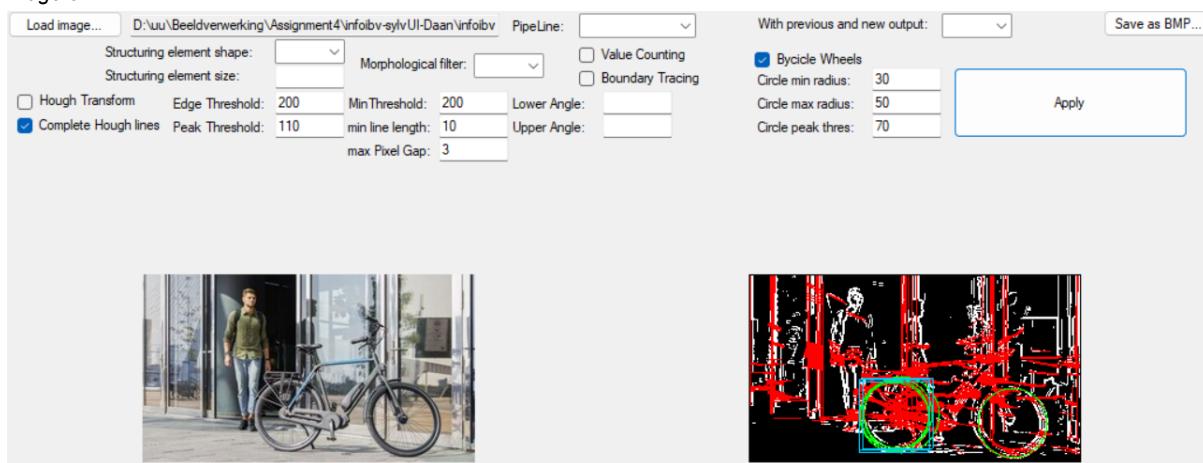


Image 6

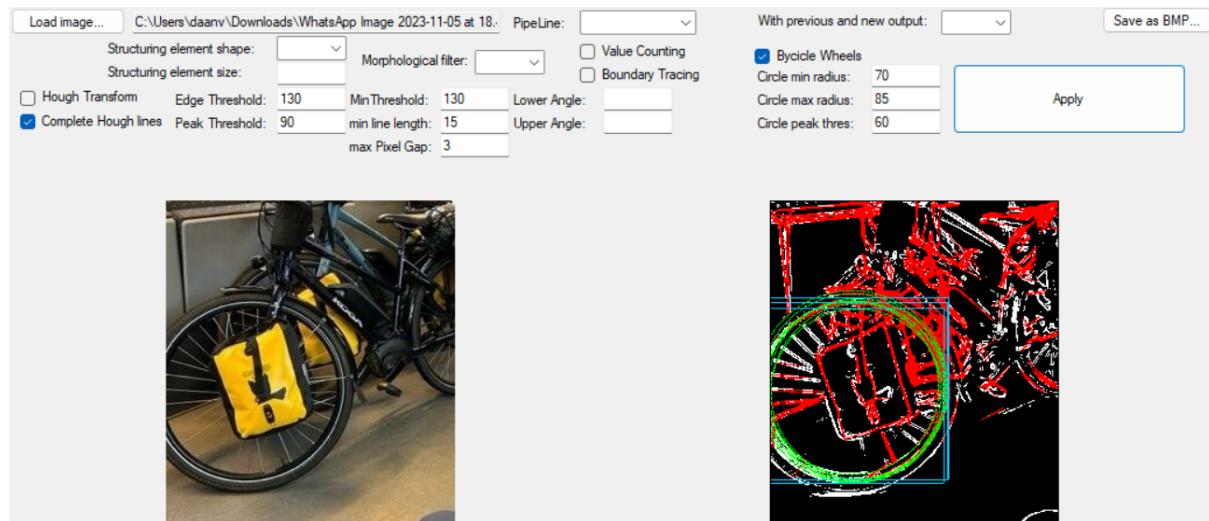


Image 7

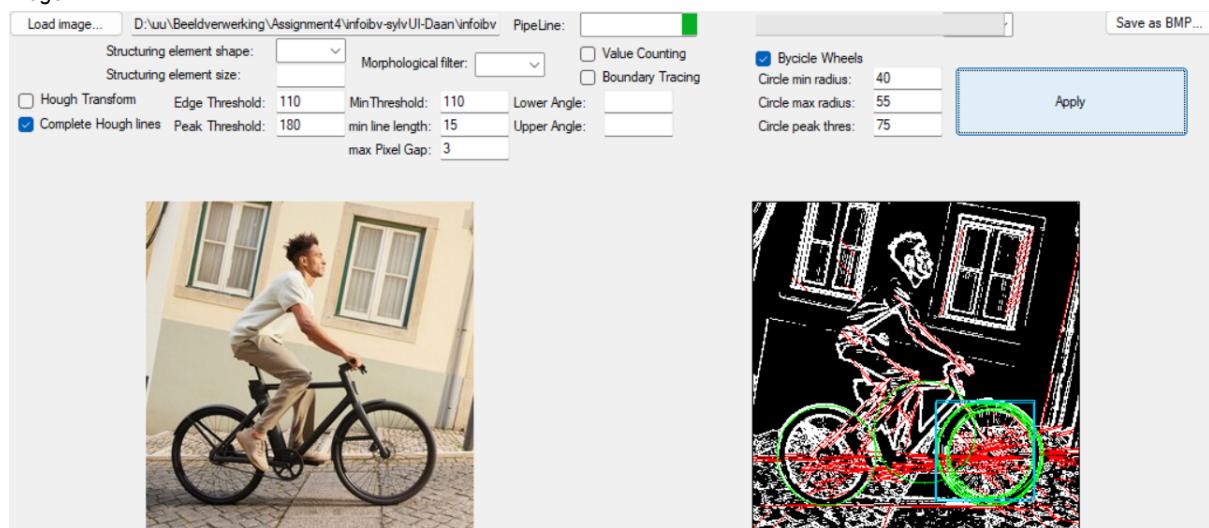


Image 8

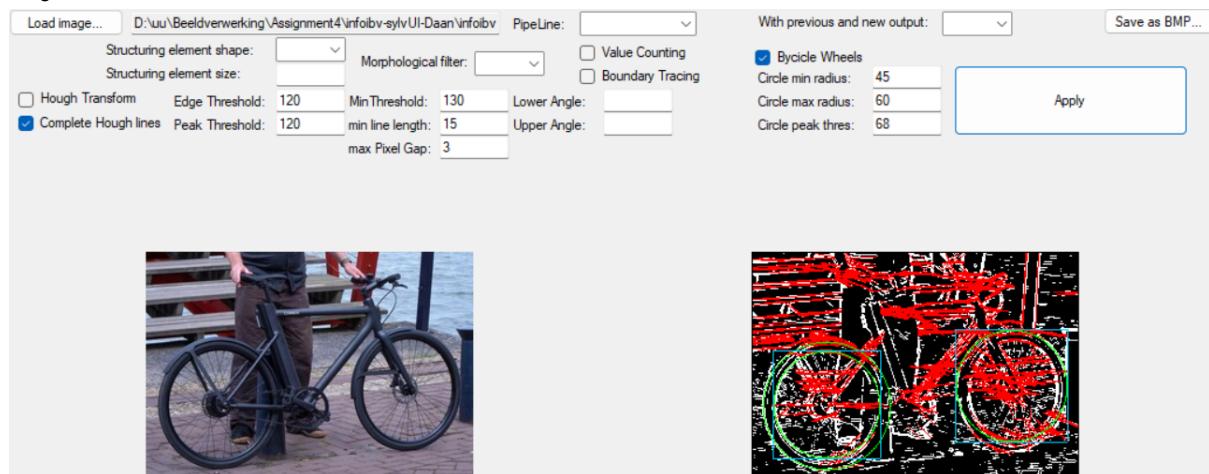


Image 9

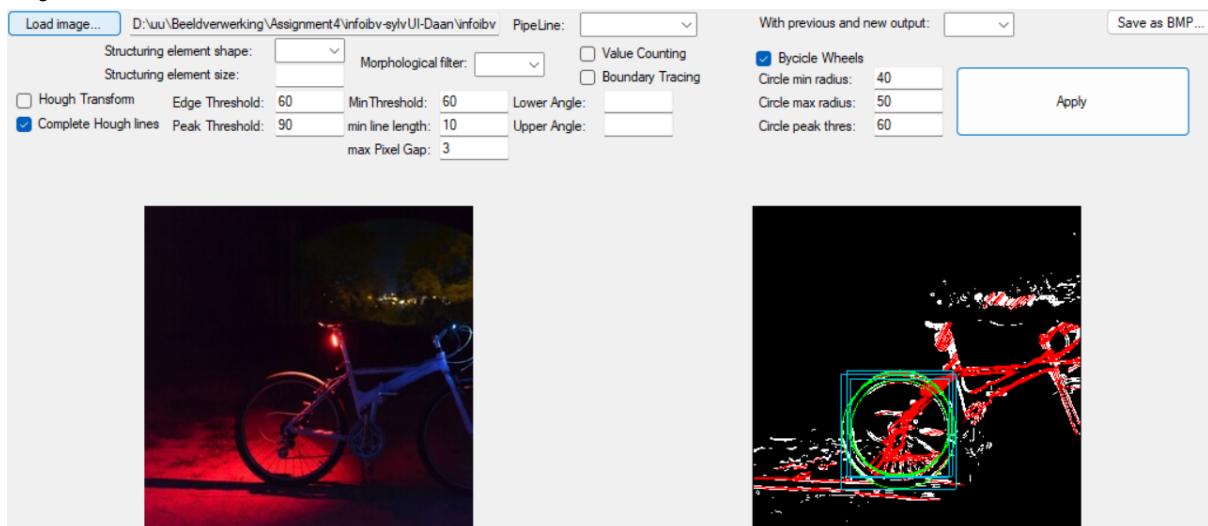
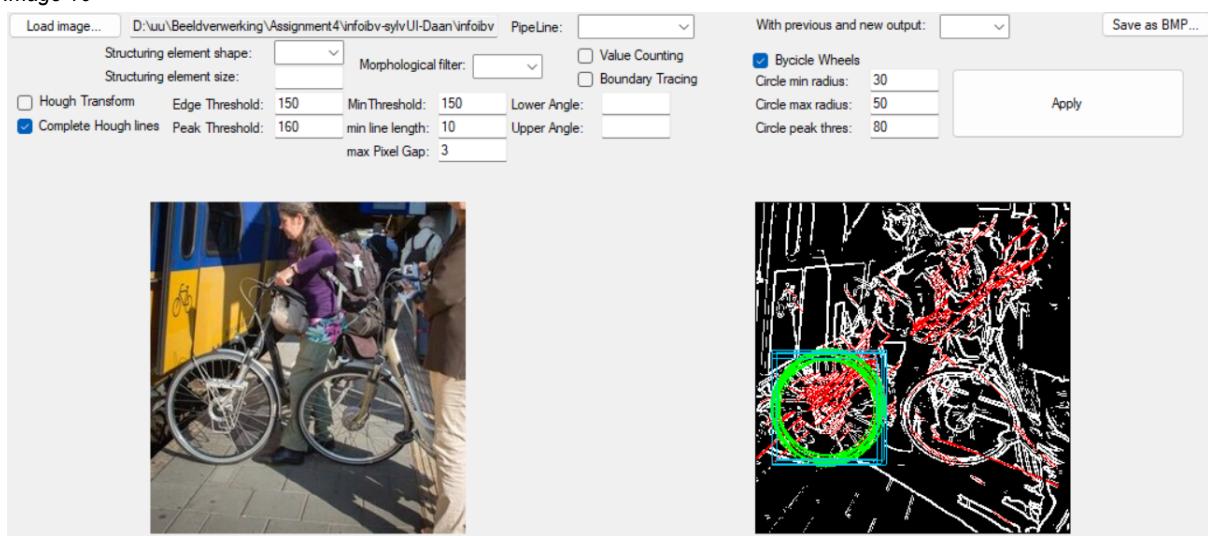


Image 10



2) A brief explanation of the pipeline and the processes in it

- Preprocessing:

For this phase, we decided to mainly use the method of edge detection (`edgeMagnitude`). We would do this to highlight the edges / boundaries between objects or regions of different intensities, so we can detect significant circles within an image. We considered using a gaussian filter to reduce noise, but noticed this caused the spokes of the bicycle wheels to be rather harder to detect. This wasn't optimal for our next phase.

- Object recognition

As we detect all these circles, we check each one whether there are line segments going through the center of them, and whether these stay within the circle. So to say, if we detect these circles, and we find spokes within these circles, we know we detected a bike wheel (the function `bicycleWheel`).

We use a method called *HoughTransformCircle*, which loops through each pixel in the input image, checking if the pixel value is greater than 0. If it is, it performs Hough Transform accumulation for circles of various radius. Thereby the method *peakFindingCircle* helps us find detected circles which hold to a certain threshold.

We also visualize the result, using our method *visualizeHoughCircles* and *visualizeHoughLineSegments*, afterwards by marking the wheel with a green color, spokes with red, within a blue box.

```

if (BicycleWheels.Checked)
{
    int minRadius = Convert.ToInt32(CircleRadius.Text);
    int maxRadius = Convert.ToInt32(CircleRadius.Text);
    int circPeakThres = Convert.ToInt32(CirclePeakThres.Text);

    byte[,] houghTransformImageCircle = HoughTransformCircle(workingImage, maxRadius, minRadius);
    List<Circle>, int, int> circlePeaks = peakFindingCircle(houghTransformImageCircle, circPeakThres, minRadius);
    List<Circle>, int, int> bicycleWheels = BicycleWheel(circlePeaks, LineSegments);
    workingImage = visualizeHoughLineSegments(workingImage, LineSegments);
    workingImage = visualizeHoughCircles(workingImage, circlePeaks);
    foreach (Circle item, int, int) in bicycleWheels
    {
        int centerX = BicycleWheel.Item;
        int centerY = BicycleWheel.Item2;
        int radius = BicycleWheel.Item3;
        MessageBox.Show("x:" + centerX + " y:" + centerY + " radius:" + radius);
        int image = workingImage.GetLength(0);
        int imageY = workingImage.GetLength(1);
        for (int x = centerX - radius; x < centerX + radius; x++)
        {
            if (x > 0 && x < image)
            {
                if (centerY - radius > 0)
                {
                    workingImage[x, centerY - radius] = 197;
                }
                if (centerY + radius < imageY)
                {
                    workingImage[x, centerY + radius] = 197;
                }
            }
        }
        for (int y = centerY - radius; y < centerY + radius; y++)
        {
            if (y > 0 && y < imageY)
            {
                if (centerX - radius > 0)
                {
                    workingImage[centerX - radius, y] = 197;
                }
                if (centerX + radius < imageX)
                {
                    workingImage[centerX + radius, y] = 197;
                }
            }
        }
    }
    else
    {
        workingImage = visualizeHoughLineSegments(workingImage, LineSegments);
    }
}

```

3) A reflection on your choice of parameters, and how it affects the results (also for the distractor images).

In the process of detecting these wheels, we are working with a few different parameters: Edge threshold, Peak threshold, min threshold, min line length, max pixel gap, circle min/max radius and circle peak threshold.

Edge Threshold: This determines the minimum edge magnitude required for an edge to be considered during the Hough transform. We set a higher edge threshold, when there's less noise to be taken into account, so only the most prominent edges will be considered (take for example the stock images). But rather lower when we wanted to consider more edges, as there might be less in the picture.

Peak Threshold: This sets a threshold for the magnitude of peaks in the accumulator space. A higher peak threshold indicates only the most prominent wheel candidates being detected, while a lower threshold may identify weaker candidates. We would put this lower in noisier or darker lighting condition pictures, e.g. image 9.

Minimum Threshold: This sets the minimum edge strength required for detection. A higher value filters out weaker edges, while a lower value detects more edges, useful in noisier images.

Minimum Line Length: It defines the shortest valid line length. We kept this around 10-15.

Maximum Pixel Gap: This sets the maximum gap between connected edge points. Increasing it makes us able to detect larger gaps, but could also increase the chance for false positives. We kept this on 3 for all tests.

Circle Min/Max Radius: The decision on this parameter had to be guessed with each picture, which was sometimes hard to do. As we make a 3d array for the circle hough (which consists of an x, y and a radius), it would create all possible circles for each 'on' pixel with different radius. This results in a huge list over which we need to iterate a few times, which by guessing we can speed up the whole process.

Circle Peak Threshold: This parameter sets a threshold for considering a peak in the Hough accumulator as a circle candidate. A higher peak threshold will result in more stringent filtering, leading to fewer detected circles.

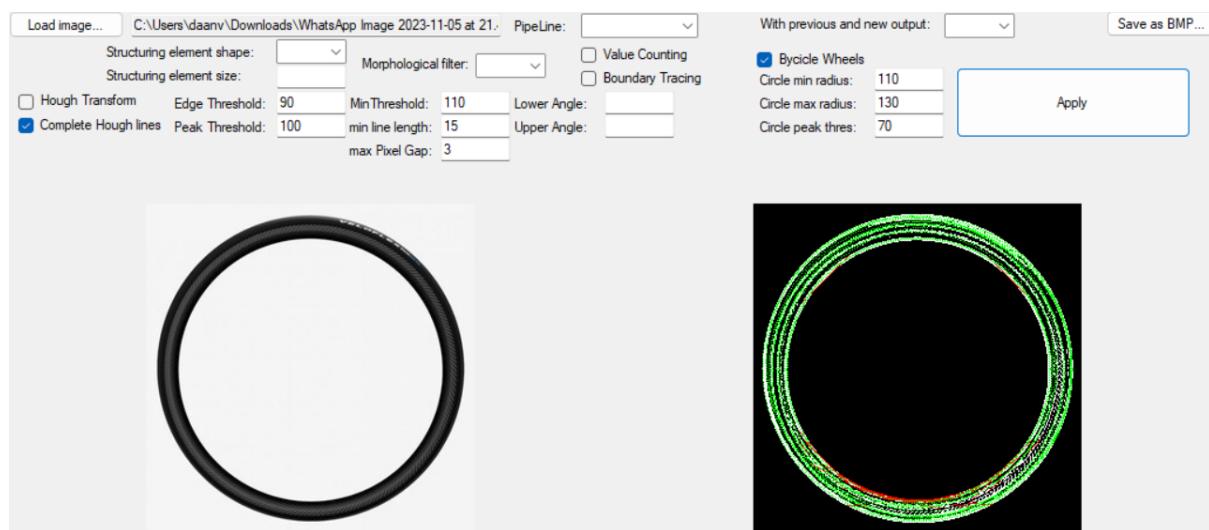
It's important we set the parameters for each image accordingly to avoid false positives or whatsoever.

4) Example output, with some good and some bad detections. Discuss these results.

Our method of detecting bicycles could work, but we face some problems when a bike is tilted, causing it not to be a circle anymore. We won't be able to detect half a wheel, for example. We can see an example of this in image 10.

It could also create a chance of false positives: if we put the line threshold significantly low, we could for example detect the wheels of a car as well. For it to detect bicycle wheels specifically, it would require detecting a bicycle completely, to verify we have a bike wheel on our hands.

Since we use circle detection for our bicycle wheel detector, we might also see moments (which it technically should do) we mark prominent circles in images with a green color. We can see this happen quite a lot in our distractor images.



Load image... D:\uu\Beeldverwerking\Assignment4\infoibv-sylv\UI-Daan\infoibv PipeLine: With previous and new output: Save as BMP...

Structuring element shape: Structuring element size: Morphological filter: Value Counting
 Hough Transform Edge Threshold: 250 MinThreshold: 250 Lower Angle:
 Complete Hough lines Peak Threshold: 80 min line length: 10 Upper Angle:
max Pixel Gap: 3

Bicycle Wheels Circle min radius: 15
Circle max radius: 30 Circle peak thres: 70

Apply



Load image... C:\Users\daanv\Downloads\WhatsApp Image 2023-11-05 at 21. PipeLine: With previous and new output: Save as BMP...

Structuring element shape: Structuring element size: Morphological filter: Value Counting
 Hough Transform Edge Threshold: 200 MinThreshold: 200 Lower Angle:
 Complete Hough lines Peak Threshold: 120 min line length: 10 Upper Angle:
max Pixel Gap: 3

Bicycle Wheels Circle min radius: 50
Circle max radius: 60 Circle peak thres: 58

Apply

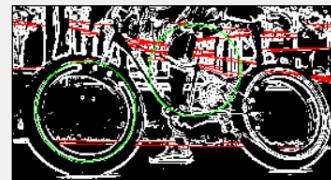


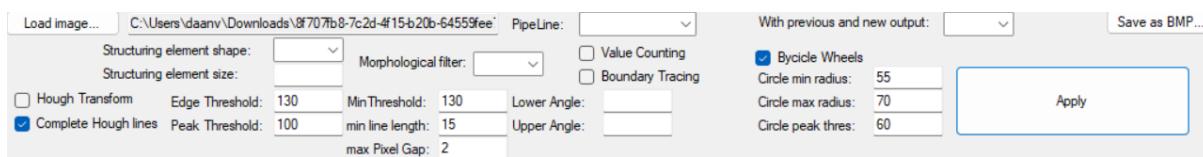
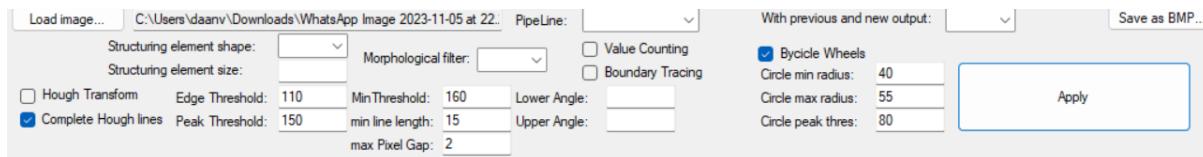
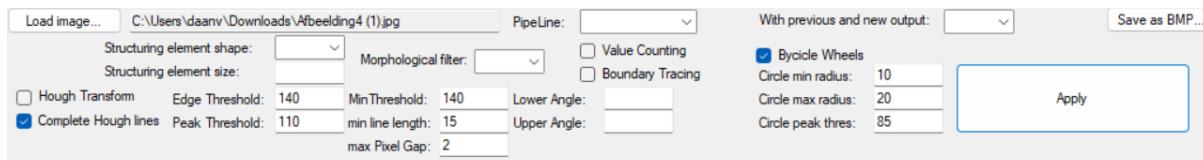
Load image... C:\Users\daanv\Downloads\WhatsApp Image 2023-11-05 at 21. PipeLine: With previous and new output: Save as BMP...

Structuring element shape: Structuring element size: Morphological filter: Value Counting
 Hough Transform Edge Threshold: 160 MinThreshold: 160 Lower Angle:
 Complete Hough lines Peak Threshold: 160 min line length: 15 Upper Angle:
max Pixel Gap: 2

Bicycle Wheels Circle min radius: 40
Circle max radius: 55 Circle peak thres: 68

Apply



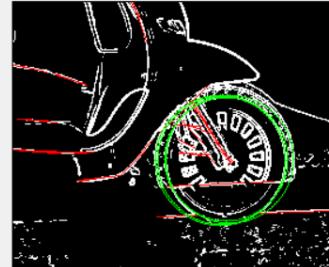


Load image... C:\Users\daanv\Downloads\Afbeelding5.jpg Pipeline: With previous and new output: Save as BMP...

Structuring element shape:	Morphological filter:	<input type="checkbox"/> Value Counting
Structuring element size:		<input type="checkbox"/> Boundary Tracing
<input type="checkbox"/> Hough Transform Edge Threshold: 150	MinThreshold: 150	Lower Angle:
<input checked="" type="checkbox"/> Complete Hough lines Peak Threshold: 95	min line length: 15	Upper Angle:
	max Pixel Gap: 2	

Bycicle Wheels
Circle min radius: 50
Circle max radius: 60
Circle peak thres: 65

Apply



Load image... C:\Users\daanv\Downloads\c1byrob2hto_360.jpg Pipeline: With previous and new output: Save as BMP...

Structuring element shape:	Morphological filter:	<input type="checkbox"/> Value Counting
Structuring element size:		<input type="checkbox"/> Boundary Tracing
<input type="checkbox"/> Hough Transform Edge Threshold: 120	MinThreshold: 120	Lower Angle:
<input checked="" type="checkbox"/> Complete Hough lines Peak Threshold: 150	min line length: 10	Upper Angle:
	max Pixel Gap: 3	

Bycicle Wheels
Circle min radius: 15
Circle max radius: 30
Circle peak thres: 80

Apply



Load image... C:\Users\daanv\Downloads\WhatsApp Image 2023-11-05 at 23. Morphological filter: With previous and new output: Save as BMP...

Structuring element shape:	Morphological filter:	<input type="checkbox"/> Value Counting
Structuring element size:		<input type="checkbox"/> Boundary Tracing
<input type="checkbox"/> Hough Transform Edge Threshold: 120	MinThreshold: 120	Lower Angle:
<input checked="" type="checkbox"/> Complete Hough lines Peak Threshold: 150	min line length: 10	Upper Angle:
	max Pixel Gap: 3	

Bycicle Wheels
Circle min radius: 15
Circle max radius: 30
Circle peak thres: 78

Apply

