Image Processing assignment

By: Ian Zunderdorp (3643034) and Kian Khoshnoodpour (3709248)

# Problem:

Our chosen problem was to recognize roadside traffic signage. To be more precise, we wanted to recognize the presence of the well-known traffic signs, since recognizing the exact sign is extremely difficult with the tools available with Image Processing.

The well-known traffic signs almost all have one of the following shapes:

* Rectangular (sometimes rotated)
* Circular
* Triangular

# Approach:

The shapes of the signs are extremely basic and thus can be recognized easily using only a few shape factors. For example a circular sign would theoretically have a circularity near 1 and rectangularity near 0.8, while a triangle would have a rectangularity of near 0.5 and a circularity of near 0.6. However, this assumes perfect shapes and in reality all signs have smoothed corners meaning that the shape is slightly off from the theoretical geometrical shape and thus the values need to be tweaked.

Traffic signs are usually painted in the following colors: Red, blue, white and black. However, there are also instances of grey and yellow being used in some signs. This provides us with an interesting option: If we acquire the area in which the sign is, using for example filtering on the primary color at the edge of the sign and then generating a bounding box, we can look at the combination of colors used. For the majority of the signs, we expect mostly a blue&white or red&white sign. Using a threshold on the red value we can extract the red areas that we expect on the edges of red&white signs. This generates a binary image onto which we can then perform closings and openings to remove noise and imperfections. Now if we have a mostly continuous area for the sign, we can calculate the bounding box for the sign and see if the dimensions and position of the bounding box roughly matches with the bounding box of the shapes that we calculated earlier and if so, see if the colored histogram is roughly what we expected.