



Implementation plan: RGB to Greyscale

1.1 Name & Date

Dylan van Eck - 19/06/2020

1.2 Goal

The goal is to write an RGB to Greyscale conversion method for the face recognition program.

With object recognition it is almost always unnecessary to work in the colour model RGB. It takes more time to get data from three channels (R, G, B) than from one channel (Greyscale or Intensity).

1.3 Methods

Lightness method

The lightness method averages the most prominent and least prominent colours:

$$(\max(R, G, B) + \min(R, G, B))/2$$

Average method

The average method simply averages the values:

$$(R + G + B)/3$$

Luminosity method

The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colours, so green is weighted most heavily. The formula for luminosity is:

$$(0.21 * R + 0.72 * G + 0.07 * B)$$

1.4 Choice

The lightness method tends to reduce contrast. The luminosity method works best overall. However, some images look better using one of the other algorithms. And sometimes the three methods produce very similar results.

This is what I read on the johndcook.com so I will trust that the luminosity method has in general an better score.

1.5 Implementation

I will implement the luminosity method in the "StudentPreProcessing.cpp" file. The RGB pixel will be converted with the algorithm:

$$(0.21 * R + 0.72 * G + 0.07 * B)$$

1.6 Evaluation

This "new" method will be compared with the default method. With this comparison shall I look at the differences in pixel values with the default.

Usage of image

I will use the three images of each type (so man-1, female-1, child-1). this will make a more accurate conclusion as opposed to using one image.

The implementation will be done round about 24 hours.