# Task 3

## a) Threshold Image for Road Sign (Colour) Isolation

A combination of *threshold* values has to be used to isolate road signs with a particular colour in an RGB image. First, in MATLAB read the RGB image with the imread command:

filename = (‘roadsigns.jpg’);

L = imread(filename);

The image (roadsigns.jpg) contains road signs with the colours yellow, red and blue. The formula in Eq. 3.1 can be used to isolate the yellow road signs (colours) of the image, where the values of T1 and T2 should be high, and T3 should be low.

Eq. 3.1

The values below for T1,T2 and T3 isolate the yellow colours in MATLAB:

y = L(:,:,1)>150 & L(:,:,2)>150 & L(:,:,3)<50;

After isolating the yellow colours in the binary image y, the output should be stacked to get the same shape as image L. The following command is used:

yStacked = cat(3, y);

Then, the stacked version is converted to uint8 data type for multiplication by the original image to get the new image with only yellow road signs; the following command is used:

L\_yellowFeatures = L .\* uint8(yStacked);

The formulas that can be used to isolate the red colour sign is given in Eq. 3.2. The values of T1 and T3 should be relatively low, and the value for T2 should be low.

Eq. 3.2

The threshold values for the equations above is given in the lines of MATLAB code below:

% Red colour (road signs)

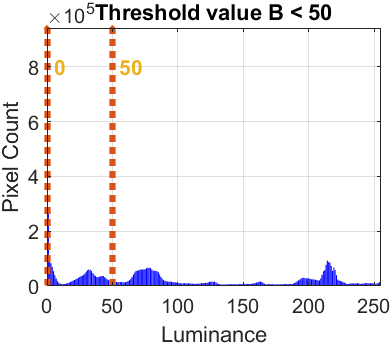
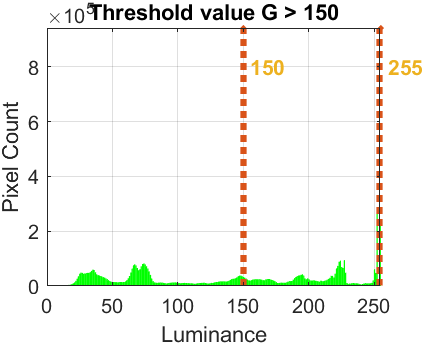
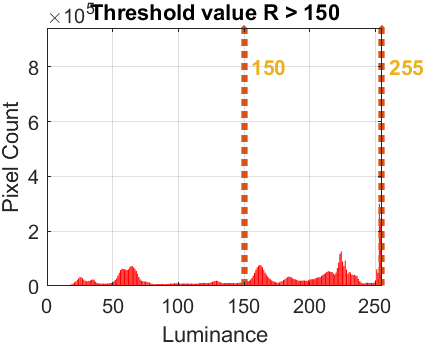
r = L(:,:,1)>50 & L(:,:,2)<50 & L(:,:,3)<50;

rStacked = cat(3, r);

L\_redFeatures = L .\* uint8(rStacked);

Fig 3.1 shows the *histogram* of the red, green and blue colour of the daytime road image, alongside the threshold values for each colour for separating the yellow road signs.

threshold range for G



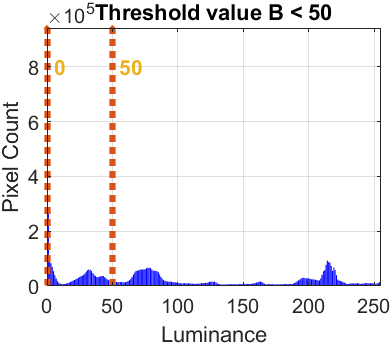
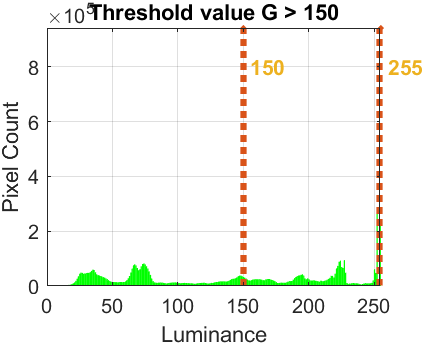
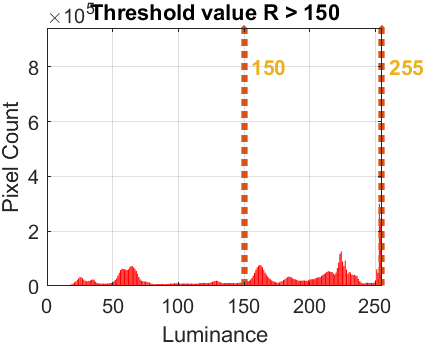
threshold range for R

threshold range for B

**Fig. 3.1**: Histogram of each colour (R, G and B) in the input image and the threshold values for separating the yellow road signs in the image.

Fig. 3.2 shows the same histograms, but with threshold values for red isolation in the input image.

threshold range for G



threshold range for R

threshold range for B

**Fig. 3.2**: Histogram of each colour (R, G and B) in the input image and the threshold values for separating the red road signs in the image.

The results after thresholding are shown in image Fig. 3.3.

The road signs are not very clear in Fig. 3.3, to get a better result, one should consider using *morphological operations*. The operations filling (Eq. 3.3) (Soille, 2014), closing (Eq. 3.4) and opening (Eq. 3.5) are applied to get a better result, note that this has to be done with the binary image so before stacking the image, cat().



original image

yellow road sign, R > 150 ∧ G > 150 ∧ B < 50

yellow road sign, R > 150 ∧ G > 150 ∧ B < 50

**Fig. 3.3**: Original image and the result after yellow and red colour isolation for road sign recognition.

Eq. 3.3

where,

Eq. 3.4

Eq. 3.5

The new results are shown in Fig 3.4 below.



original image

yellow road sign, R > 150 ∧ G > 150 ∧ B < 50

yellow road sign, R > 150 ∧ G > 150 ∧ B < 50

**Fig. 3.4**: Original image and the results after thresholding, filling, closing and opening for road sign recognition.

## Source Code

Source code of task3.m is added here. Visit MATLAB code for functions.

**task3.m**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Title: Threshold Image for Road Sign Isolation

% Author: Samir Habibi

% Rev. Date: 23/11/2020

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear; % Delete all variables.

close all; % Close all windows.

clc; % Clear command window.

% Ask user for file by presenting options with menu() command.

fileChoice = menu('File', 'Road Signs', 'Choose own');

% Use switch() to read file based on user's choice (fileChoice).

switch (fileChoice)

case 1

filename = ('roadsigns.jpg');

L = imread(filename);

case 2

filename = uigetfile('');

L = imread(filename);

end % End the switch-statement after obtaining image.

% Create array for each color band in the original image, so 3 separate 2D

% arrays.

R = L(:,:,1);

G = L(:,:,2);

B = L(:,:,3);

% Check if the image's datatype is uint8, this is required for scaling

% later. Obtained from https://nl.mathworks.com/matlabcentral/fileexchange/26420-simplecolordetection

if strcmpi(class(L), 'uint8')

% 256 levels

eightBit = true;

else

eightBit = false;

end

% Call methods for recognition and isolation of yellow features.

L\_yellowFeatures = getYellowFeatures(L, R, G, B);

% Call methods for recognition and isolation of red features.

L\_redFeatures = getRedFeatures(L, R, G, B);

figure;

% Maximize user screen for image plots.

set(gcf, 'Position', get(0, 'ScreenSize'));

% Display original image.

subplot(3, 4, 1);

imshow(L);

title('Input Image');

% Display histogram of image L.

subplot(3, 4, 2);

H\_count = imhist(L);

bar(0:255, H\_count, 'k');

title('Input Image Histogram')

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

xlim([0 255]);ylim([0 max(H\_count)]);

% Display image with only yellow features.

subplot(3, 4, 5);

imshow(L\_yellowFeatures);

title('Yellow Road Sign');

% Get and plot the red colour histogram of the input image. The histograms

% for each triplet are based on SimpleColorDetectio?n() by Image Analyst,

% available at: <https://nl.mathworks.com/matlabcentral/fileexchange/26420-simplecolordetection>

histRed = subplot(3, 4, 6);

[countsRed, lumRed] = imhist(R);

maxLumRed = find(countsRed > 0, 1, 'last'); % x-axis

maxCountRed = max(countsRed); % y-axis

bar(countsRed, 'r');

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value R > 150');

% Get and plot the green colour histogram of the input image.

histGreen = subplot(3, 4, 7);

[countsGreen, lumGreen] = imhist(G);

maxLumGreen = find(countsGreen > 0, 1, 'last');

maxCountGreen = max(countsGreen);

bar(countsGreen, 'g', 'BarWidth', 0.95);

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value G > 150');

% Get and plot the blue colour histogram of the input image.

histBlue = subplot(3, 4, 8);

[countsBlue, lumBlue] = imhist(B);

maxLumBlue = find(countsBlue > 0, 1, 'last');

maxCountBlue = max(countsBlue);

bar(countsBlue, 'b');

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value B < 50');

% Plot all 3 color bands' histogram.

subplot(3, 4, 3);

plot(lumRed, countsRed, 'r', 'LineWidth', 2);

xlabel('Luminance');

ylabel('Pixel Count');

grid on;

hold on;

plot(lumGreen, countsGreen, 'g', 'LineWidth', 2);

plot(lumBlue, countsBlue, 'b', 'LineWidth', 2);

title('Histogram of All Colour Bands');

% Get the maximum luminance of each colour in the RGB triplet.

maxLuminance = max([maxLumRed, maxLumGreen, maxLumBlue]);

% Set x-axis to appropriate value for data type uint8, maximum luminance is

% 255. If not datatype uint8, then take the maximum luminance.

if eightBit

xlim([0 255]);

else

xlim([0 maxLuminance]);

end

% Set both axes to same size, easier to compare.

% x-axis

maxLum = max([maxLumRed, maxLumGreen, maxLumBlue]);

if eightBit

maxLum = 255;

end

% y-axis

maxCount = max([maxCountRed, maxCountGreen, maxCountBlue]);

axis([histRed histGreen histBlue], [0 maxLum 0 maxCount]);

% Display image with only red features.

subplot(3, 4, 9);

imshow(L\_redFeatures);

title('Red Road Sign');

% Plot red colour histogram again, adjust title appropriately.

subplot(3, 4, 10);

bar(countsRed, 'r');

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value R > 50');

% Plot green colour histogram again, adjust title appropriately.

subplot(3, 4, 11);

bar(countsGreen, 'g');

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value < 50');

% Plot blue colour histogram again, adjust title appropriately.

subplot(3, 4, 12);

bar(countsBlue, 'b');

grid on;

xlabel('Luminance');

ylabel('Pixel Count');

title('Threshold value B < 50');

% Show the thresholds as vertical red bars on the histograms. For this task

% the function PlaceThresholdBars() is used (small adjustments, such as

% removing unused variables). Aformentioned function was found in

% SimpleColorDetectio?n() by Image Analyst, available at:

% <https://nl.mathworks.com/matlabcentral/fileexchange/26420-simplecolordetection>

PlaceThresholdBars(6, 150, 255);

PlaceThresholdBars(7, 150, 255);

PlaceThresholdBars(8, 0, 50);

PlaceThresholdBars(10, 50, 255);

PlaceThresholdBars(11, 0, 50);

PlaceThresholdBars(12, 0, 50);

## Bibliography

Dos Anjos A. and Shahbazkia, H. R., 2008. BI-LEVEL IMAGE THRESHOLDING - A Fast Method. *In Proceedings of the First International Conference on Bio-inspired Systems and Signal Processing*, [e-journal] 2, pp. 70-76. http://doi.org/10.5220/0001064300700076.

MathWorks, 2015. *How can i extract color from RGB image section ?*. [online] Available at: <https://nl.mathworks.com/matlabcentral/answers/258906-how-can-i-extract-color-from-rgb-image-section> [Accessed 23 November 2020].

MathWorks, 2018. *imfill* [online] Available at: <https://nl.mathworks.com/help/images/ref/imfill.html?searchHighlight=imfill&s\_tid=srchtitle#bup36av-2> [Accessed 30 November 2020].

MathWorks, 2018. *SimpleColorDetectio​n()* [online] Available at: <https://nl.mathworks.com/matlabcentral/fileexchange/26420-simplecolordetection> [Accessed 23 November 2020].

Soille, P. 2014. *Morphological Image Analysis: Principles and Applications*. [e-book] New York: Springer-Verlag Berlin Heidelberg. Available at: Google Books <https://books.google.nl/books?id=ZFzxCAAAQBAJ&printsec=frontcover#v=onepage&q&f=false> [Accessed 30 November 2020].

blue road signs, R < 50 ∧ G < 50 ∧ B > 50

**Fig. 3.4**: Original image and the result after yellow, red and blue colour isolation, by using thresholding, for road sign recognition.