# Task 4

## b) The histogram tail trim (contrast enhancement)

The histogram tail trim stretch can be applied on an image to avoid the problem with histogram stretch. The problem with histogram stretch is that it relies on the extrema, subsequently, if an image contains a pixel value 0 and 255, the stretch would not be successful. Histogram tail trim stretch avoids this because it uses the *5th and 95th percentile* method on the *normalised cumulative count histogram* of the image. The following code in MATLAB gives the normalised cumulative count histogram of an image *L*, after getting the *count* and *cumulative count histogram*:

% Histograms of image L

H = imhist(L); % Count Histogram

H\_cml = cumsum(H); % Cumulative Count Hist.

H\_normCmlCount = H\_cml./(M\*N); % Norm. Cumul. Count Hist.

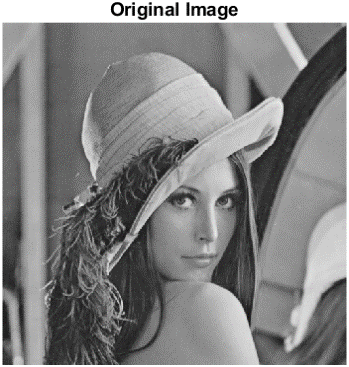
The formula in Eq. 4.1 gives the stretched version of the image *L*. In MATLAB, the formulas below are used instead of the built-in function imadjust() in combination with the stretchlim() function.

Eq. 4.1

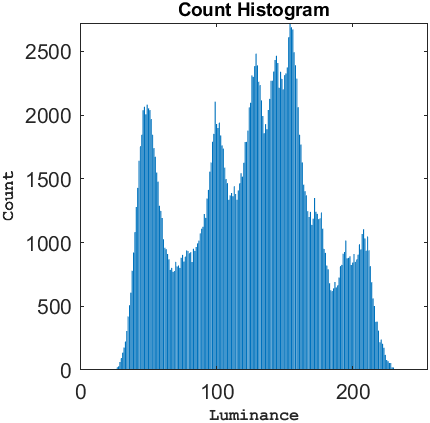
For the tail trim stretch, one should obtain the 5th and 95th percentile in the normalised cumulative count histogram, and these values replace the *minimum* and *maximum luminance* in the stretch formula (Eq. 4.1). The formula for the tail trim stretched version of the image is given in Eq. 4.2.

Eq. 4.2

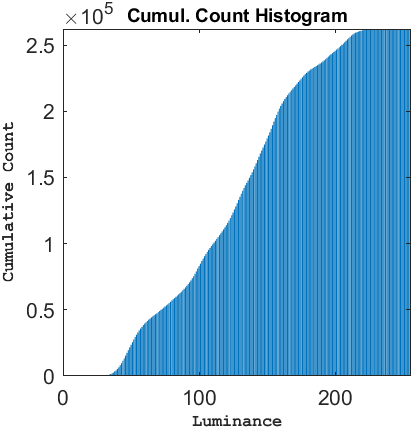
Fig. 4.1 shows the original image with the count, cumulative count and normalised cumulative count histogram of the image.



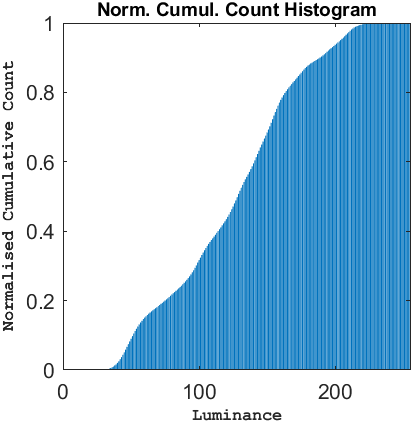
original



count histogram



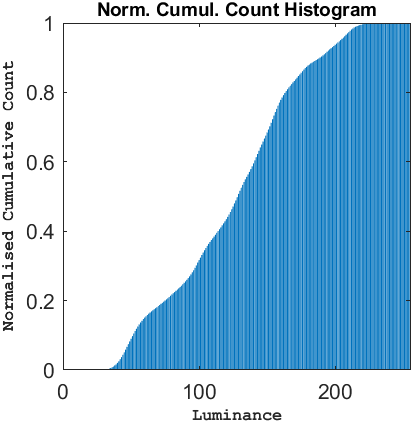
cumulative count histogram



norm. cumulative count histogram

**Fig. 4.1**: Greyscale image Lena (512X512), including the count, cumulative count, and normalised cumulative count histogram.

The new trimmed values for the maximum and minimum luminance value are found at the 5th and 95th percentile in the normalised cumulative count histogram in Fig. 4.2.



0.95

0.05

**Fig. 4.2**: Normalised cumulative count, and the 5th and 95th percentile.

In MATLAB, the following line of code gets the 5th and 95th percentile of the H\_normCmlCount histogram.

tol = 0.01; % Set tolerance (H\_normCmlCount is of type double)

minTrim = x(abs(H\_normCmlCount-0.05) < tol); % 5th percentile

L\_trimmedMin = mean(minTrim); % Get mean outputs in minTrim

maxTrim = x(abs(H\_normCmlCount-0.95) < tol); % 95th percentile

L\_trimmedMax = mean(maxTrim); % Get mean outputs in maxTrim

Fig. 4.3 shows the effect of histogram stretch and histogram tail trim stretch on the greyscale image Lena, the count histogram for each image is provided too.



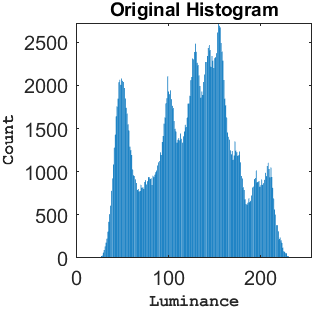
original



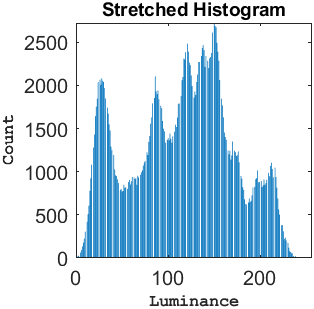
stretched



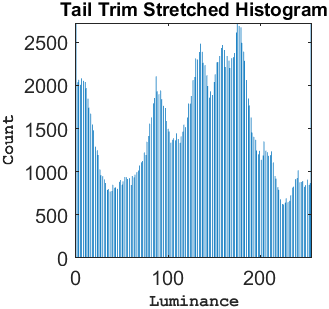
tail trim stretched



original count histogram



stretch count histogram



tail trim stretch count histogram

**Fig. 4.3**: Greyscale image Lena (512X512) before, after histogram stretch, and after tail trim stretch. Count, cumulative count, and normalised cumulative count histogram of greyscale Lena.

## Source Code

Source code of task4.m is added here. Visit MATLAB code for functions.

**task4.m**

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

% Title: Histogram tail trim stretch

% Author: Samir Habibi

% Rev. Date: 24/11/2020

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

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', 'Lena Greyscale', 'Poor Contrast CCTV1' , 'Choose own');

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

switch (fileChoice)

case 1

filename = ('lenaGreyscale.bmp');

L = imread(filename);

case 2

filename = ('poorContrastCCTV1.bmp');

L = imread(filename);

case 3

filename = uigetfile('');

L = imread(filename);

end % End the switch-statement after obtaining image in L.

% Request valid value from user for 't' in stretch formula.

t = input('Enter desired highest pixel value (255, most common): ');

% While input is not valid, keep asking user.

while (t < 0) || (t > 255) % Limits.

t = input('Enter number between 1 and 255: ');

end % End if correct value for userInput is found.

% Get row and columns of the image.

M = size(L, 1);

N = size(L, 2);

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

% Get histogram with imhist(), then the cumulative count histogram with

% cumsum(), and eventually the normalised cumulative count histogram by

% dividing the cumulative count histogram by thee number of rows (M) and

% columns(N) of image.

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

% Make count histograms.

H = imhist(L);

% Make cumulative histogram from count histogram.

H\_cml = cumsum(H);

% Make normalised cumulative count histogram from cumulative histogram.

H\_normCmlCount = H\_cml./(M\*N);

% Call method performStretch().

L\_stretched = performStretch(L, t);

% Call method performTailTrimStretch().

L\_trimStretched = performTailTrimStretch(L, t, H\_normCmlCount);

figure;

% Display original image.

subplot(2, 3, 1);

imshow(L);

title('Original Image', 'FontSize', 9, 'FontWeight', 'bold');

% Display image after stretch contrast enhancement.

subplot(2, 3, 2);

imshow(L\_stretched);

title('Stretched Image', 'FontSize', 9, 'FontWeight', 'bold');

% Display image after tail trim stretch contrast enhancement.

subplot(2, 3, 3);

imshow(L\_trimStretched);

title('Tail Trim Stretched Image', 'FontSize', 9, 'FontWeight', 'bold');

% Plot the count histogram of image L

subplot(2, 3, 4)

countHist = bar((0:255), imhist(uint8(L)));

title('Count Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

xlim([0, 255]);ylim([0, max(imhist(uint8(L)))]);

axis square;

% Plot the cumulative count histogram of image L

subplot(2, 3, 5)

cumulCountHist = bar((0:255), H\_cml);

title('Cumul. Count Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Cumulative Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

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

axis square;

% Plot the normalised cumulative count histogram.

% This histogram is used for the 5th and 95th percentile method /

% tail trim stretch enhancement technique.

subplot(2, 3, 6)

histBar = bar((0:255), H\_normCmlCount);

title('Norm. Cumul. Count Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Normalised Cumulative Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

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

axis square;

figure;

% Display original image.

subplot(2, 3, 1);

imshow(L);

title('Original Image', 'FontSize', 9, 'FontWeight', 'bold');

% Display image after stretch contrast enhancement.

subplot(2, 3, 2);

imshow(L\_stretched);

title('Stretched Image', 'FontSize', 9, 'FontWeight', 'bold');

% Display image after tail trim stretch contrast enhancement.

subplot(2, 3, 3);

imshow(L\_trimStretched);

title('Tail Trim Stretched Image', 'FontSize', 9, 'FontWeight', 'bold');

% Plot the count histogram of image L

subplot(2, 3, 4)

countHist2 = bar((0:255), imhist(uint8(L)));

title('Original Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

xlim([0, 255]);ylim([0, max(imhist(uint8(L)))]);

axis square;

% Plot the count histogram of stretched image L

subplot(2, 3, 5)

stretchHist = bar((0:255), imhist(L\_stretched));

title('Stretched Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

xlim([0, 255]);ylim([0, max(imhist(uint8(L)))]);

axis square;

% Plot the count histogram of tail trim stretched image L

subplot(2, 3, 6)

tailStretchHist = bar((0:255), imhist(L\_trimStretched));

title('Tail Trim Stretched Histogram', 'FontSize', 9, 'FontWeight', 'bold');

xlabel('Luminance', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

ylabel('Count', 'FontName', 'Courier', 'FontSize', 8, 'FontWeight', 'bold');

xlim([0, 255]);ylim([0, max(imhist(uint8(L)))]);

axis square;

## Bibliography

Dr. Rashi Agarwal, 2013. *Histogram equalization and contrast stretching in Matlab.* [video online] Available at: <https://www.youtube.com/watch?v=w9mzbiDjGYY> [Accessed 1 June 2009].

MathWorks, n.d. *stretchlim* [online] Available at: <https://nl.mathworks.com/help/images/ref/stretchlim.html> [Accessed 1 December 2020].

MathWorks, n.d. *imadjust* [online] Available at: <https://nl.mathworks.com/help/images/ref/imadjust.html> [Accessed 1 December 2020].

MathWorks, n.d. *abs* [online] Available at: <https://nl.mathworks.com/help/matlab/ref/abs.html> [Accessed 24 December 2020].

tutorialspoint, n.d. *Histogram stretching* [online] Available at: <https://www.tutorialspoint.com/dip/histogram\_stretching.htm> [Accessed 24 November 2020].