Table of Contents

Universal Image Quality Index

```
% Author: NALLAPALEM NEERAJSRINIVAS & ROHITH KAMBAMPATI
% NET ID: ns620 & ka517
% Roll Number: 1710110224 & 1710110168
% Instructor: Prof. Vijay Kumar Chakka and Dr. Upendra Pandey
% DSP PROJECT : UNIVERSAL IMAGE QUALITY INDEX

clc;            % Clear Command window
clearvars;      % Clear all variables
close all;      % close all Figures and Plots

x = double(imread('OrigLena.gif'));%converts the iimage read to double
h1 = double(imread('Meanshiftlena.gif'));%converts the iimage read to
 double
Q1 = UIQI(x, h1);%calling the function universal image quality index
[MSE1,PSNR1] = msepsnr(x,h1);%calling the function that generates MSE
 and PSNR
figure;
display(['MEAN SHIFT LENA IMAGE - ','Q = ',num2str(Q1),'  MSE =
 ',num2str(MSE1)]);%displays the value of Q1
subplot(2,2,1);
imshow(uint8(x));%displays the mean shift image
title('Original Image');
subplot(2,2,2);
imshow(uint8(h1));%displays the original image
title('Mean shift lena image');%title of the image

h4 = double(imread('Contraststretchinglena.gif'));%reads the image and
 converts into double
Q4 = UIQI(x, h4);%calling universal image quality index function
[MSE4,PSNR4] = msepsnr(x,h4);%calling the function that generates MSE
 and PSNR
display(['CONTRAST STRETCHING LENA IMAGE - ','Q = ',num2str(Q4),'  MSE
 = ',num2str(MSE4)]);
subplot(2,2,3);
imshow(uint8(x));%displays the image
title('Original Image');%the title of the image
subplot(2,2,4);
imshow(uint8(h4));
title('Contrast stretching lena image');
```

```
h7 = double(imread('saltpepperlena.gif'));
Q7 = UIQI(x, h7);
[MSE7,PSNR7] = msepsnr(x,h7);
figure;                                    %same comments as above
display(['SALT PEPPER LENA IMAGE - ','Q = ',num2str(Q7),'  MSE =
 ',num2str(MSE7)]);
subplot(2,2,1);
imshow(uint8(x));
title('Original Image');
subplot(2,2,2);
imshow(uint8(h7));
title('Salt pepper lena image');

h6 = double(imread('Multipliedspecklelena.gif'));
Q6 = UIQI(x, h6);
[MSE6,PSNR6] = msepsnr(x,h6);
display(['MULTIPLIED SPECKLE LENA IMAGE - ','Q = ',num2str(Q6),'  MSE
 = ',num2str(MSE6)]);
subplot(2,2,3);    %same comments as above         %
imshow(uint8(x));
title('Original Image');
subplot(2,2,4);
imshow(uint8(h6));
title('Multiplied speckle lena image');

h3 = double(imread('Additivegaussianlena.gif'));
Q3 = UIQI(x, h3);
[MSE3,PSNR3] = msepsnr(x,h3);
figure;             %comments same as above
display(['ADDITIVE GAUSSIAN LENA IMAGE - ','Q = ',num2str(Q3),'  MSE =
 ',num2str(MSE3)]);
subplot(2,2,1);
imshow(uint8(x));
title('Original Image');
subplot(2,2,2);
imshow(uint8(h3));
title('Additive gaussian lena image');

h2 = double(imread('Blurringlena.gif'));
Q2 = UIQI(x, h2);
[MSE2,PSNR2] = msepsnr(x,h2);
display(['BLURRING LENA IMAGE - ','Q = ',num2str(Q2),'  MSE =
 ',num2str(MSE2)]);
subplot(2,2,3);
imshow(uint8(x));       %comments same as above
title('Original Image');
subplot(2,2,4);
imshow(uint8(h2));
title('blurring lena image');

h5 = double(imread('JPEGcompressedlena.gif'));
Q5 = UIQI(x, h5);
[MSE5,PSNR5] = msepsnr(x,h5);
figure;             %comments same as above
```

```
display(['JPEG COMPRESSED LENA IMAGE - ','Q = ',num2str(Q5),'  MSE =
 ',num2str(MSE5)]);
subplot(2,2,1);
imshow(uint8(x));
title('Original Image');
subplot(2,2,2);
imshow(uint8(h5));
title('JPEG compressed lena image');

% from statistical Image quality
Qa = quality(x, h1);%getting the quality index from our code
Qb = quality(x, h4);
Qc = quality(x, h7);
Qd = quality(x, h6);
Qe = quality(x, h3);
Qf = quality(x, h2);
Qg = quality(x, h5);
display(['MEAN SHIFT LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qa)]);%displays the image
display(['CONTRAST STRETCHING LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qb)]);
display(['SALT PEPPER LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qc)]);
display(['MULTIPLIED SPECKLE LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qd)]);
display(['ADDITIVE LENA LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qe)]);
display(['BLURRING LENA IMAGE - Q(from IEEE paper) - ',num2str(Qf)]);
display(['JPEG COMPRESSED LENA IMAGE - Q(from IEEE paper) -
 ',num2str(Qg)]);
```

```
MEAN SHIFT LENA IMAGE - Q = 0.98942  MSE = 224.9993
CONTRAST STRETCHING LENA IMAGE - Q = 0.93719  MSE = 225.0932
SALT PEPPER LENA IMAGE - Q = 0.64938  MSE = 225.3684
MULTIPLIED SPECKLE LENA IMAGE - Q = 0.44076  MSE = 224.7482
ADDITIVE GAUSSIAN LENA IMAGE - Q = 0.38911  MSE = 225.1804
BLURRING LENA IMAGE - Q = 0.34612  MSE = 224.1397
JPEG COMPRESSED LENA IMAGE - Q = 0.28755  MSE = 215.1139
MEAN SHIFT LENA IMAGE - Q(from IEEE paper) - 0.99337
CONTRAST STRETCHING LENA IMAGE - Q(from IEEE paper) - 0.96393
SALT PEPPER LENA IMAGE - Q(from IEEE paper) - 0.95254
MULTIPLIED SPECKLE LENA IMAGE - Q(from IEEE paper) - 0.95322
ADDITIVE LENA LENA IMAGE - Q(from IEEE paper) - 0.95315
BLURRING LENA IMAGE - Q(from IEEE paper) - 0.94608
JPEG COMPRESSED LENA IMAGE - Q(from IEEE paper) - 0.95359
```

Original Image

Mean shift lena image

Original Image

Contrast stretching lena image

Original Image

Salt pepper lena image

Original Image

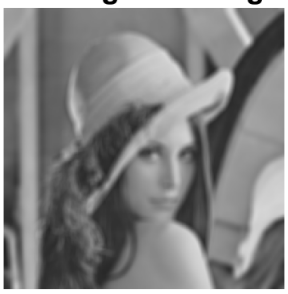
Multiplied speckle lena image

Original Image

Additive gaussian lena image

Original Image

blurring lena image



Original Image

JPEG compressed lena image

MSE & PSNR

```
function [MSE,PSNR] = msepsnr(x,h)
[m,n] = size(x);
MSE = sum(sum(((x-h).^2)))/(m*n);
PSNR = 10*log10((255^2)/MSE);
end
```

Universal Quality Index

```
function quality = UIQI(img1, img2)
block_size = 8;     % block size is considered as 8
N = block_size.^2;
sum2_filter = ones(block_size);     % creating a window size of ones
 of 8x8

img1_sq   = img1.*img1;     % multiplying the matrices element wise
img2_sq   = img2.*img2;     % multiplying the matrices element wise
img12 = img1.*img2;

img1_sum   = filter2(sum2_filter, img1, 'valid');
% filter2 does a convolution computed without including zeropadding
 edges
% This is basically considered as mean to the image 1
img2_sum   = filter2(sum2_filter, img2, 'valid');
% This is basically considered as mean to the image 2
img1_sq_sum = filter2(sum2_filter, img1_sq, 'valid');
% This is considered as variance of img1 + img1_sq
img2_sq_sum = filter2(sum2_filter, img2_sq, 'valid');
% This is considered as variance of img2 + img2_sq
img12_sum = filter2(sum2_filter, img12, 'valid');
% This is considered as covariance of img1&2 + img12
img12_sum_mul = img1_sum.*img2_sum;
% xbar*ybar
img12_sq_sum_mul = img1_sum.*img1_sum + img2_sum.*img2_sum;
% xbar^2+ybar^2
numerator = 4*(N*img12_sum - img12_sum_mul).*img12_sum_mul;
% It is a representation of 4*covariance*xbar*ybar
denominator1 = N*(img1_sq_sum + img2_sq_sum) - img12_sq_sum_mul;
% It is a representation of variance1 + variance2
denominator = denominator1.*img12_sq_sum_mul;

quality_map = numerator./denominator;
% total UIQI function
quality = mean2(quality_map);
% mean of each element in quality map matrix
end
```

Universal quality index(our code)

```
function Q=quality(x,y)
       m1=mean(mean(x));%calculating mean
       m2=mean(mean(y));
       i1 = x-m1;%calculating the diffeence between each element and
 mean
       i2 = y-m2;
       v1 = sum(sum(i1.*i1))/(numel(x)-1);%calculating variance
       v2 = sum(sum(i2.*i2))/(numel(x)-1);
       covari = sum(sum((i1.*i2)))/(numel(x)-1);%calculating
 covariance
       Q=((4*covari*m1*m2)/((v1+v2)*((m1^2)+(m2^2))));%calculation of
 image quality index
end
```

Published with MATLAB® R2015b