CS663: Digital Image Processing

Assignment 1, Question 2

(a) Linear Contrast Stretching}

function [outputImage] = myLinearContrastStretching(inputImage)

[sizeX, sizeY] = size(inputImage);

outputImage = zeros(sizeX, sizeY, 'uint8');

minVal = min(min(inputImage));

maxVal = max(max(inputImage));

f = @(x) uint8(255.0 .\* double(x - minVal) ./ double(maxVal - minVal));

outputImage = f(inputImage);

end

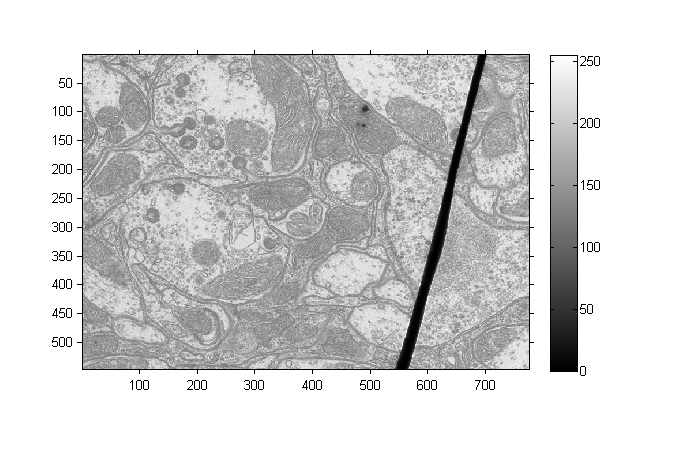
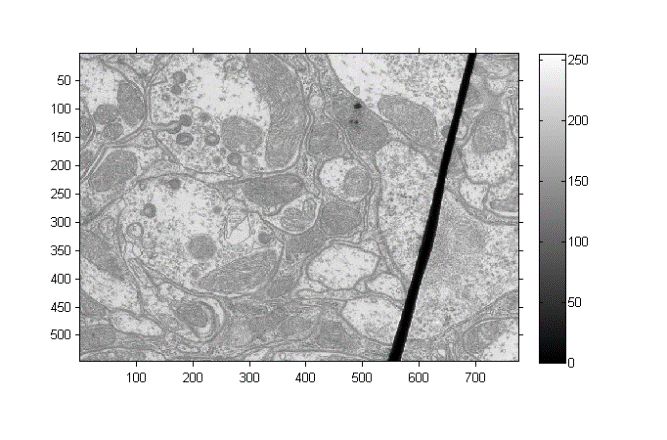
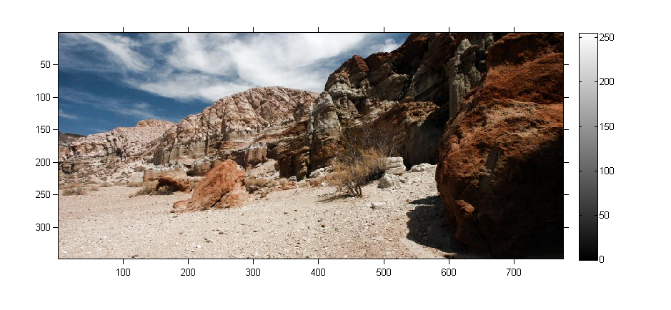
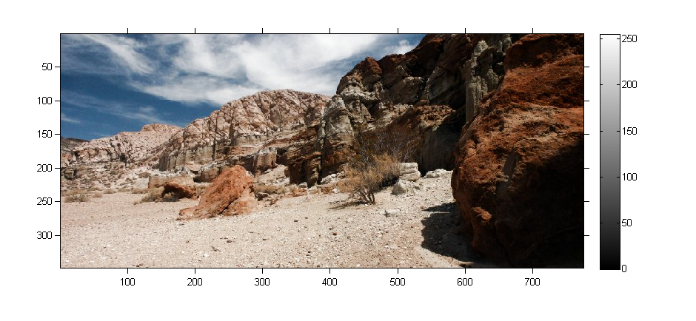
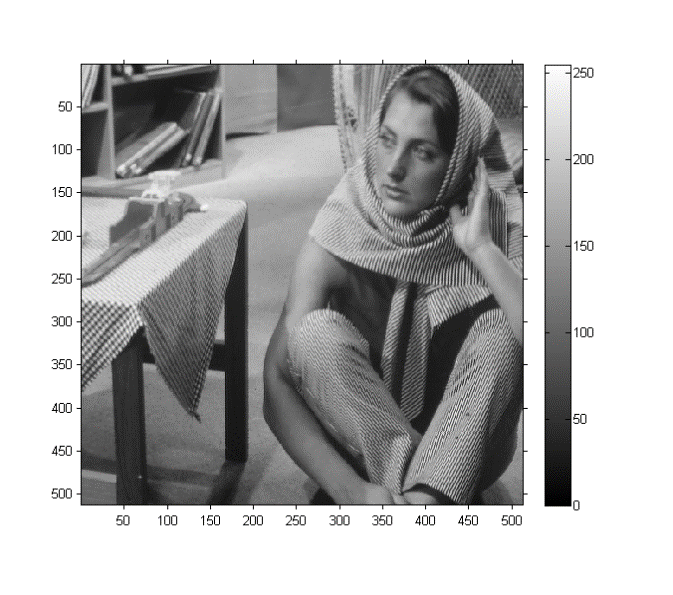


Figure. Images before and after Linear Contrast Stretching. Note that there was no noticeable difference between two images as the minimum and maximum values of the intensities in image were close 0 and 255 respectively.

(b) Global Histogram Equalization

function [outputImage] = myHE(inputImage)

% myAHE takes the inputImage, 8 bit single channel image as the input and

% enhances it using Histogram equalisation

[sizeX, sizeY] = size(inputImage);

outputImage = zeros(sizeX, sizeY, 'uint8');

ColorNumber = 256;

bins = zeros(ColorNumber, 1);

cdf = zeros(ColorNumber, 1);

for i = 1:sizeX

for j = 1:sizeY

bins(int32(inputImage(i,j))+1, 1) = bins(int32(inputImage(i,j))+1, 1) + 1;

end

end

bins = bins ./ (sizeX\*sizeY);

sum = 0;

for i = 1:ColorNumber

sum = sum + bins(i);

cdf(i,1) = sum;

end

for i = 1:sizeX

for j = 1:sizeY

outputImage(i,j) = uint8(255.0\*cdf(inputImage(i,j)+1));

end

end

end

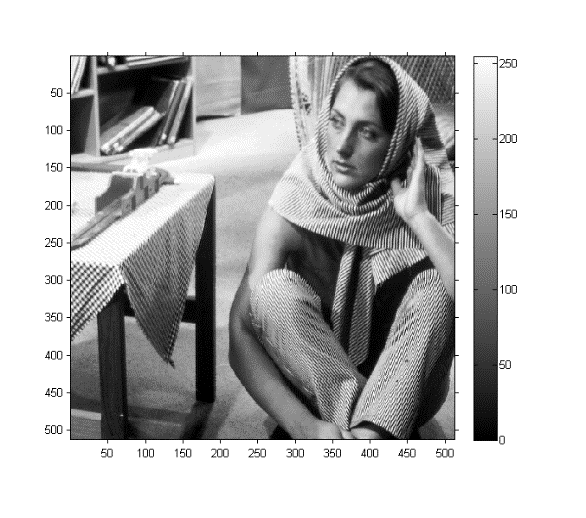
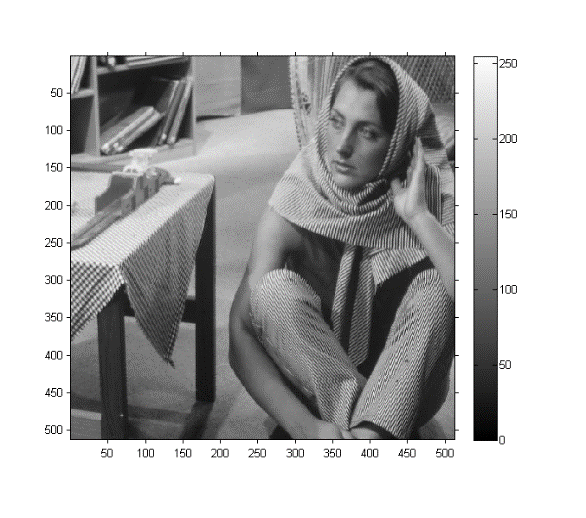


Figure. (left) Barbara, original image, (right) After global histogram equalization

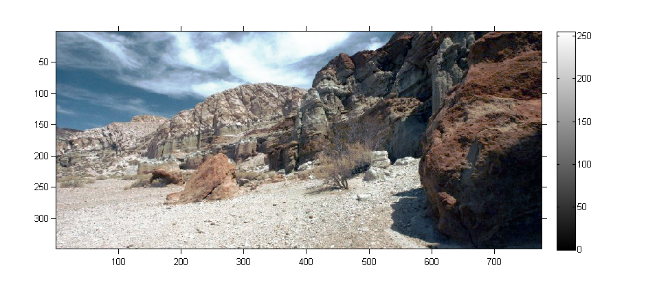
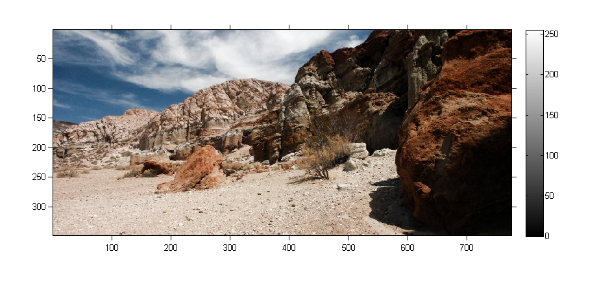


Figure. (left) canyon, original image, (right) After global histogram equalization

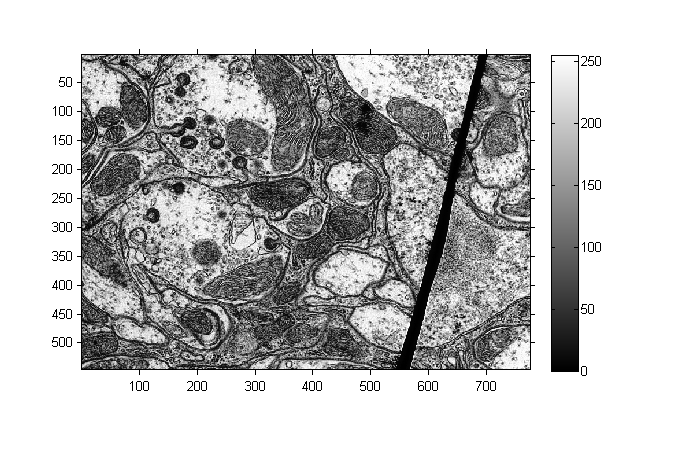
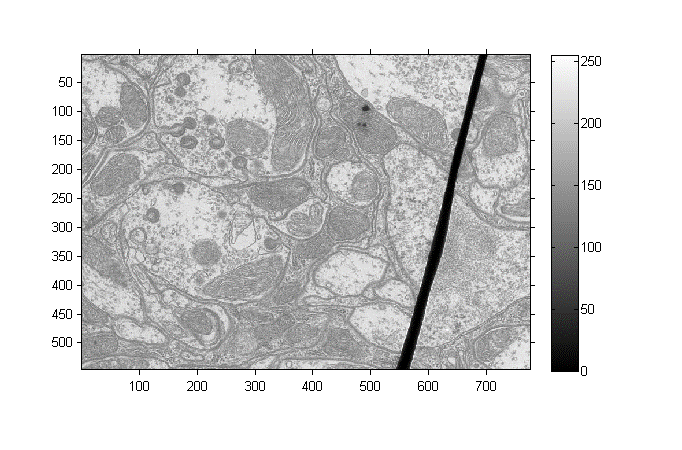


Figure. (left) TEM, original image, (right) After global histogram equalization

(c) Adaptive Histogram Equalization

function [outputImage] = myAHE(inputImage, N)

% Applies adaptive histogram equalization on inputImage with window size N

% and returns output Image

funAHE = @(x) myAHEHelper(x);

outputImage = nlfilter(inputImage, [N, N], funAHE);

end

function [outputValue] = myAHEHelper(inputImage)

% myAHE takes the inputImage, 8 bit single channel image as the input and

% enhances it using Adaptive Histogram equalisation

[sizeX, sizeY] = size(inputImage);

ColorNumber = 256;

bins = zeros(ColorNumber, 1);

cdf = zeros(ColorNumber, 1);

for i = 1:sizeX

for j = 1:sizeY

bins(int32(inputImage(i,j))+1, 1) = bins(int32(inputImage(i,j))+1, 1) + 1;

end

end

bins = bins ./ (sizeX\*sizeY);

sum = 0;

for i = 1:ColorNumber

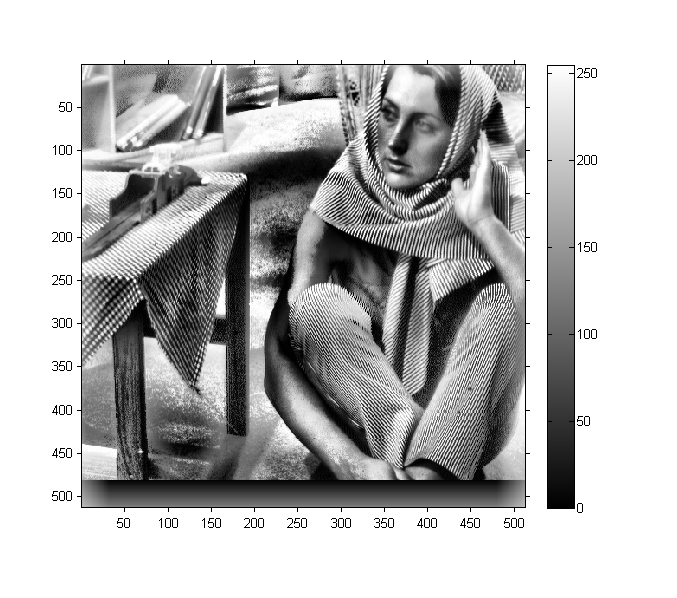
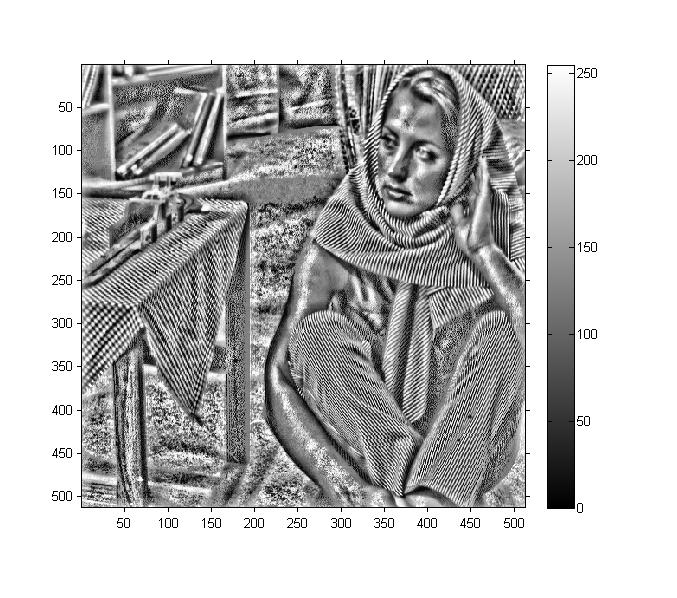
sum = sum + bins(i);

cdf(i,1) = sum;

end

outputValue = uint8(255.0\*cdf(inputImage(int32(sizeX/2), int32(sizeY/2))+1));

end



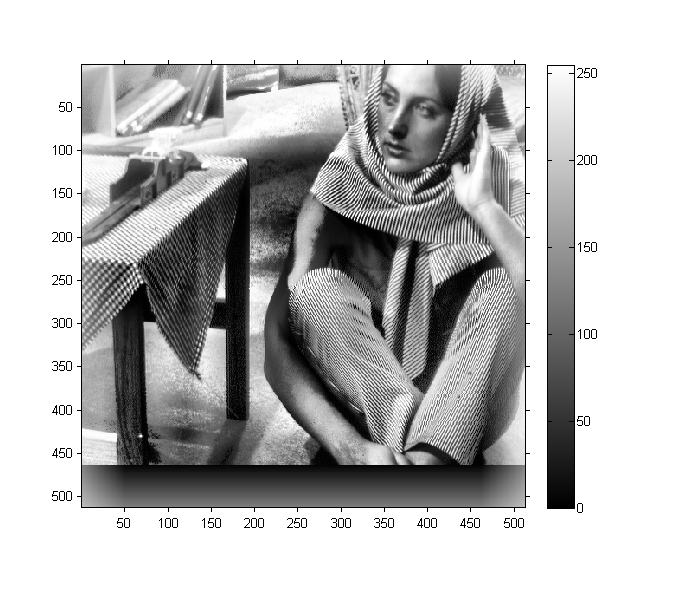


Figure. barbara after Adaptive Histogram Equalization: N= 25 (top), N = 64 (center), N=100 (bottom)

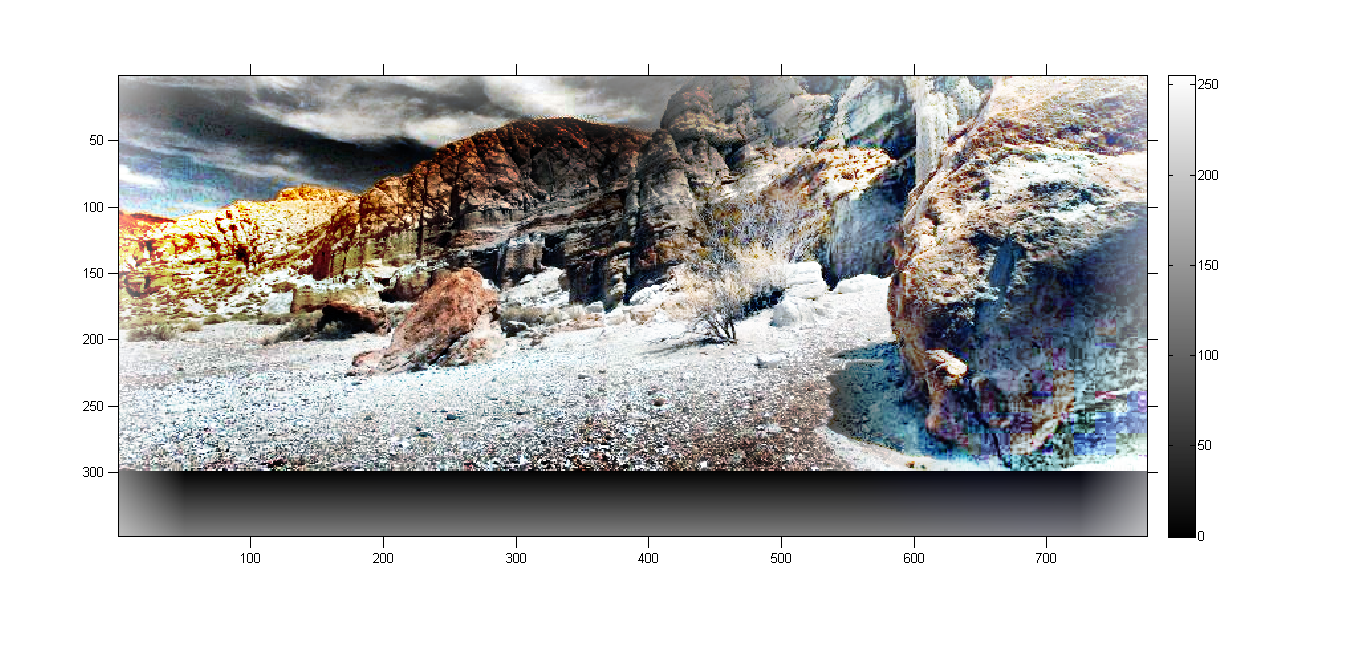
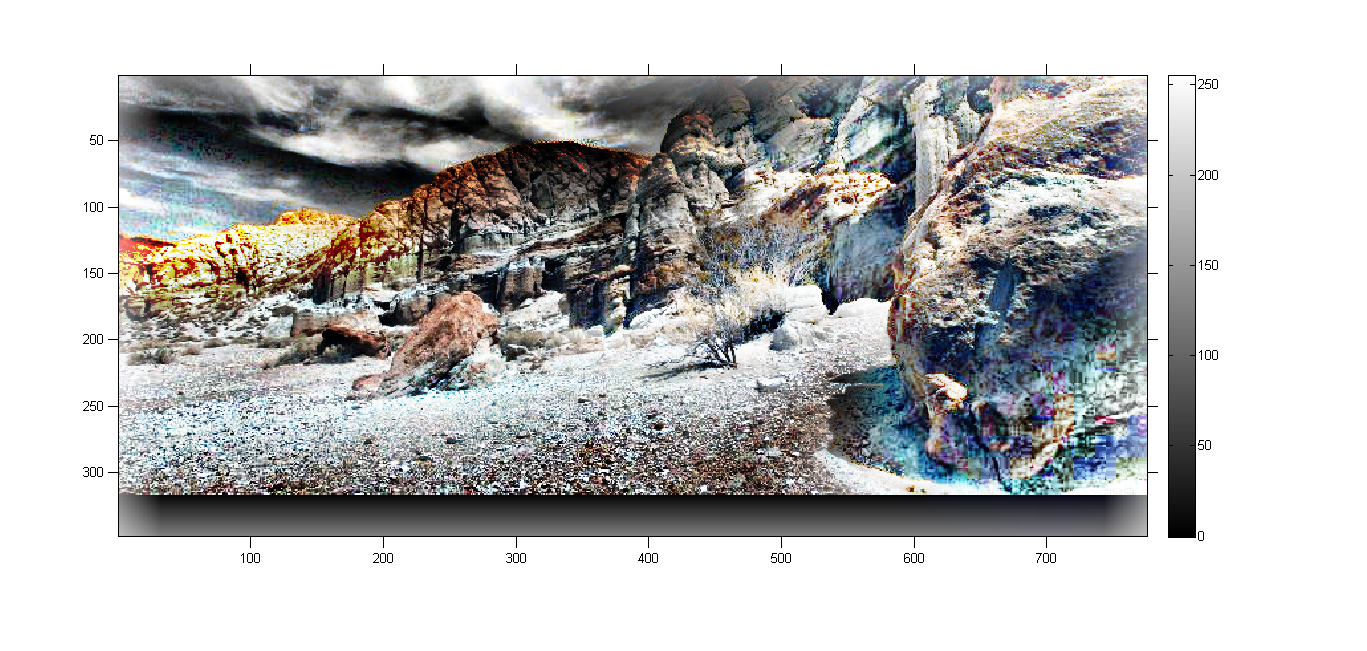
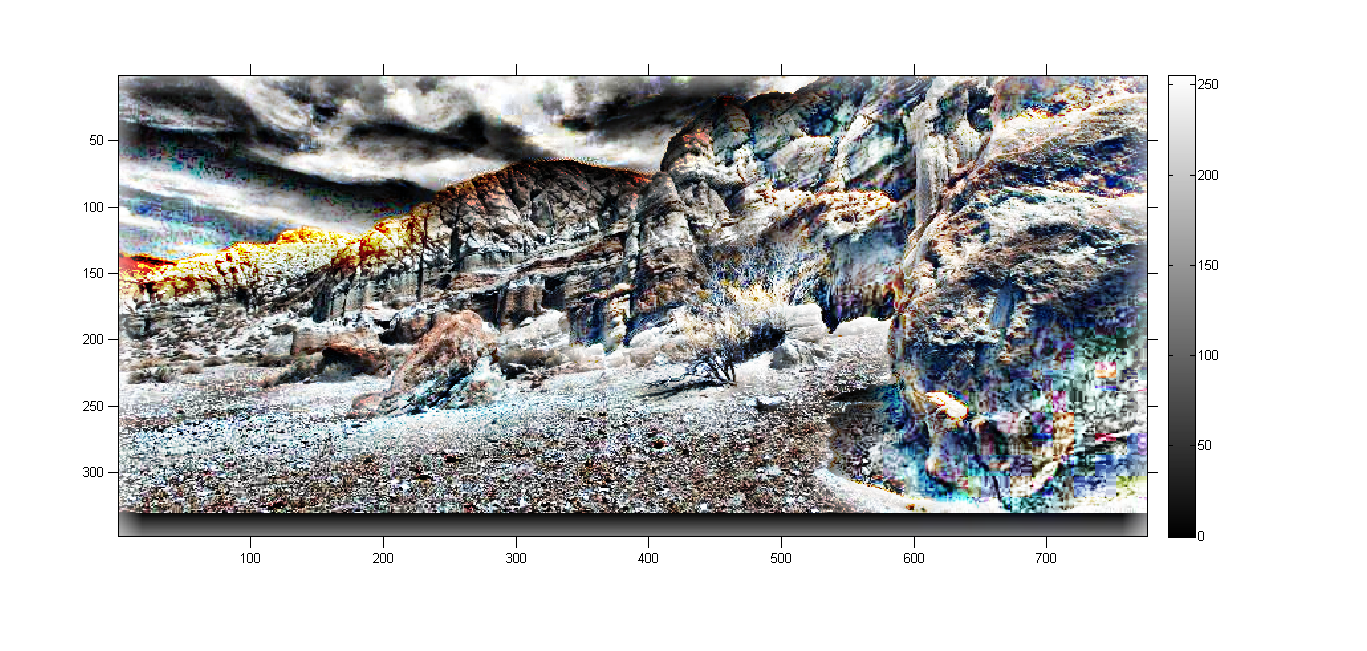


Figure. canyon after Adaptive Histogram Equalization: N= 36 (top), N = 64 (center), N=100 (bottom)

(c) Contrast Limited Adaptive Histogram Equalization

function [outputValue] = myCLAHEHelper(inputImage, C)

% myAHE takes the inputImage, 8 bit single channel image as the input and

% enhances it using Histogram equalisation

[sizeX, sizeY] = size(inputImage);

ColorNumber = 256;

bins = zeros(ColorNumber, 1);

cdf = zeros(ColorNumber, 1);

for i = 1:sizeX

for j = 1:sizeY

bins(int32(inputImage(i,j))+1, 1) = bins(int32(inputImage(i,j))+1, 1) + 1;

end

end

C = C \* sizeX \* sizeY;

points = 0;

for i = 1:ColorNumber

if bins(i) > C

points = points + (bins(i) - C);

bins(i) = C;

end

end

points = points ./ ColorNumber;

bins = (bins + points) ./ (sizeX \* sizeY);

sum = 0;

for i = 1:ColorNumber

sum = sum + bins(i);

cdf(i,1) = sum;

end

outputValue = uint8(255.0\*cdf(inputImage(int32(sizeX/2), int32(sizeY/2))+1));

end