Image Enhancement Techniques

Image Enhancement

- Image Enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application.
- Image enhancement can be done in :
  - Spatial Domain
  - Frequency Domain
- Spatial Domain Transformation are :
  - Point operations
  - Mask Operations

Point Operation

- Operation deals with pixel intensity values individually.
- The intensity values are altered using particular transformation techniques as per the requirement.
- The transformed output pixel value does not depend on any of the neighbouring pixel value of the input image.

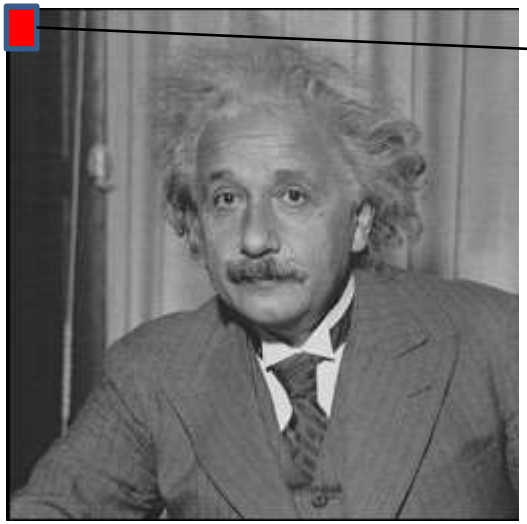
Examples:

- Image Negative.
- Contrast Stretching.
- Thresholding.
- Brightness Enhancement.
- Log Transformation.
- Power Law Transformation.

Mask Operation

- Mask is a small matrix useful for blurring, sharpening, edge-detection and more.
- New image is generated by multiplying the input image with the mask matrix.
- The output pixel values thus depend on the neighbouring input pixel values.
- The mask may be of any dimension 3X3 4X4 ….

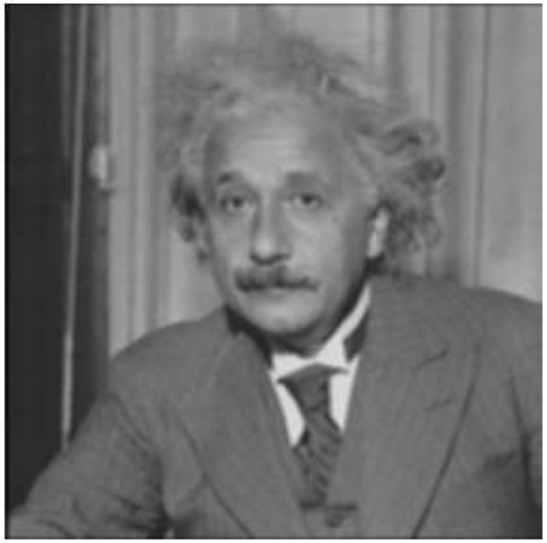
Mask Operation



0	3	0	0
0	6	1	16
0	0	2	46
0	0	2	43

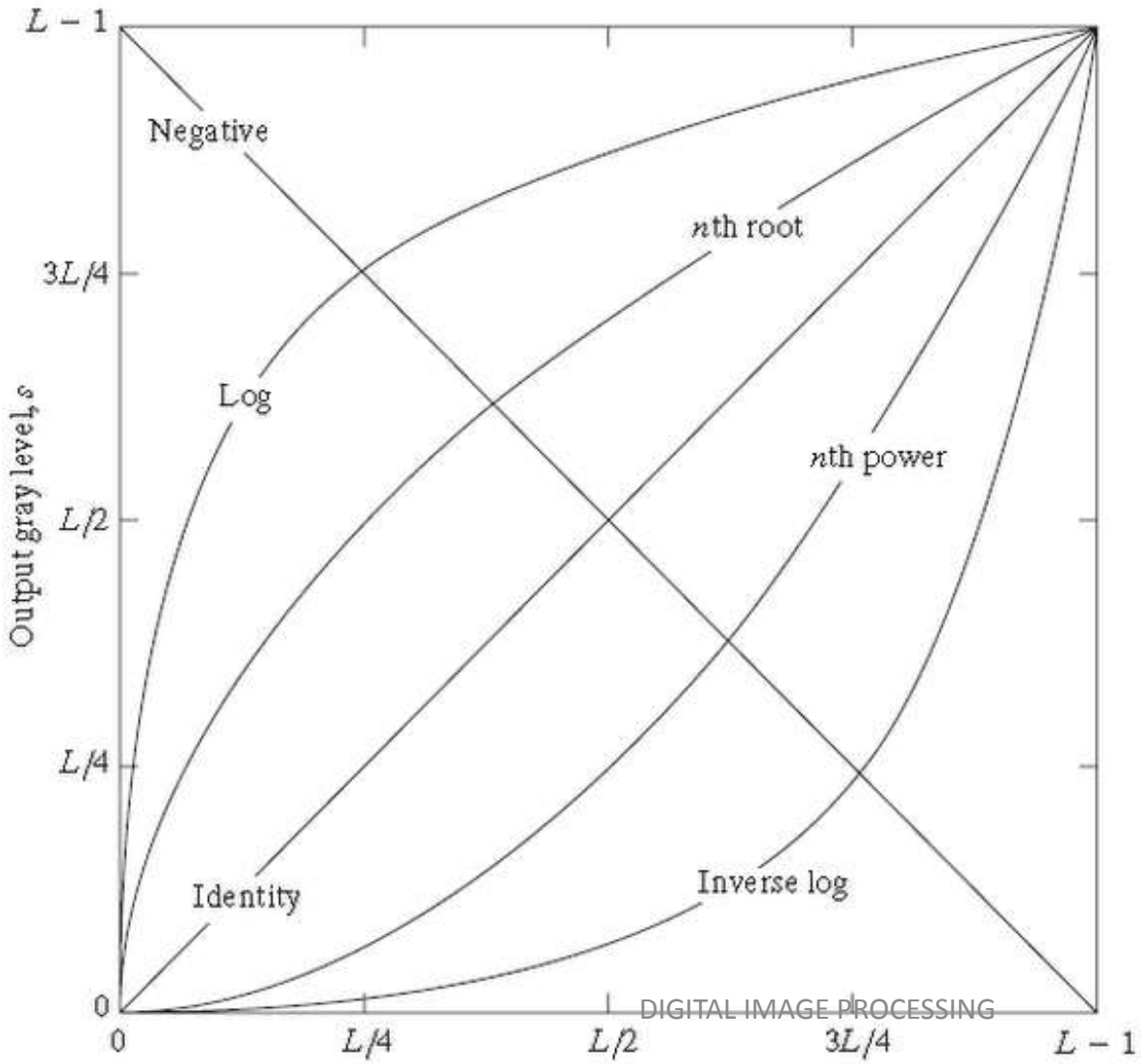
X

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Transfer function for different Intensity Transformations

$$s = L - 1 - r. \qquad (3.$$



Transfer function of

a) Negative
b) Log
c) Nth root
d) Identity
e) Nth power
f) Inverse log

Image Negative

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image.

- The negative of an image with gray levels in the range [0,L-1] is obtained by using the expression

$$s = L - 1 - r$$

L-1 = Maximum pixel value .

r = Pixel value of an image.

Image Negative

Original Image

Image negative

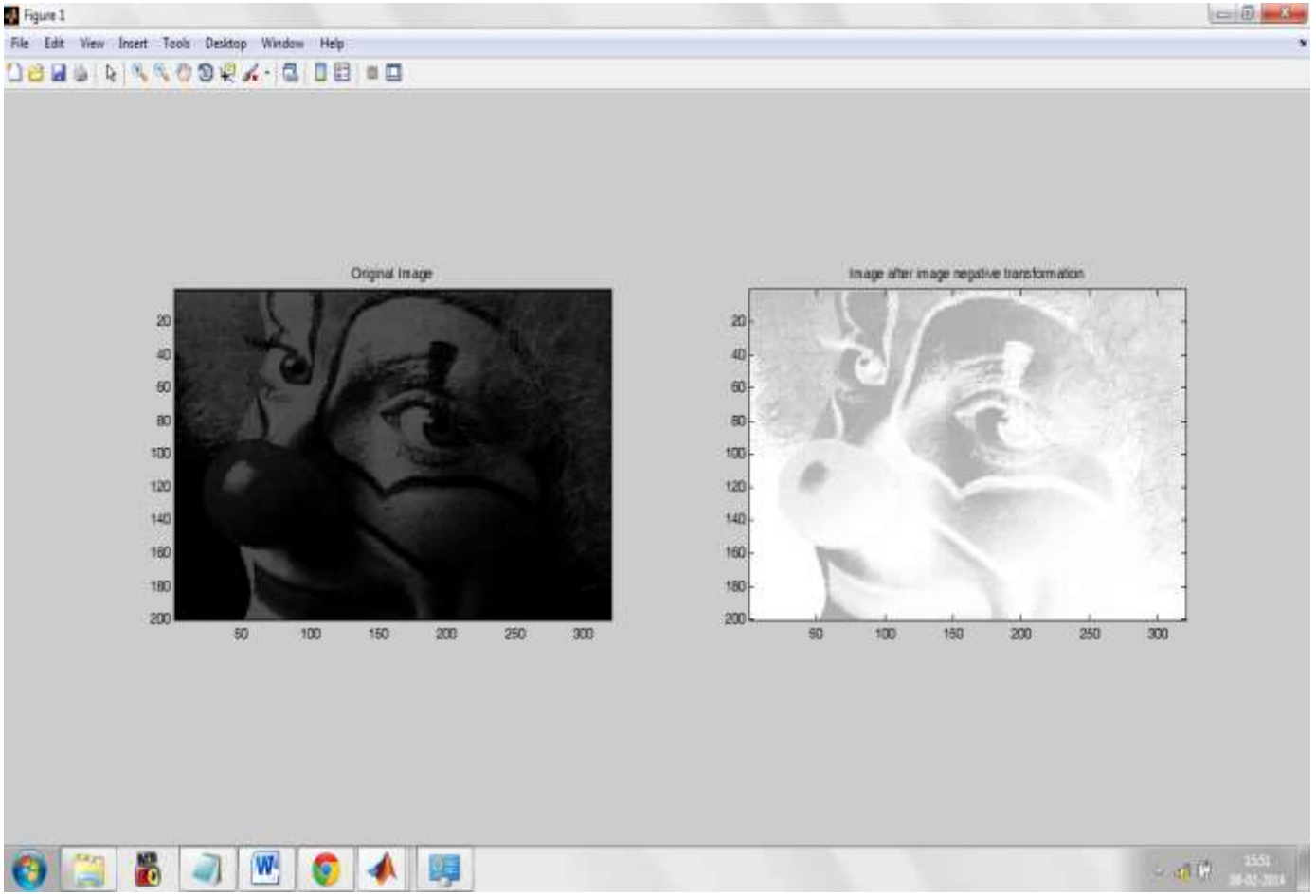
Image Negative

Matlab code :

```
% program for image enhancement using image negative
clear all
clc
close all
a=imread('clown.png');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      b(i,j)=255-a(i,j);
   end
end
subplot(1,2,1),subimage((a)),title('Original Image');
subplot(1,2,2),subimage((b)),title('Image after image
negative transformation')
```

Image Negative

Output screen:



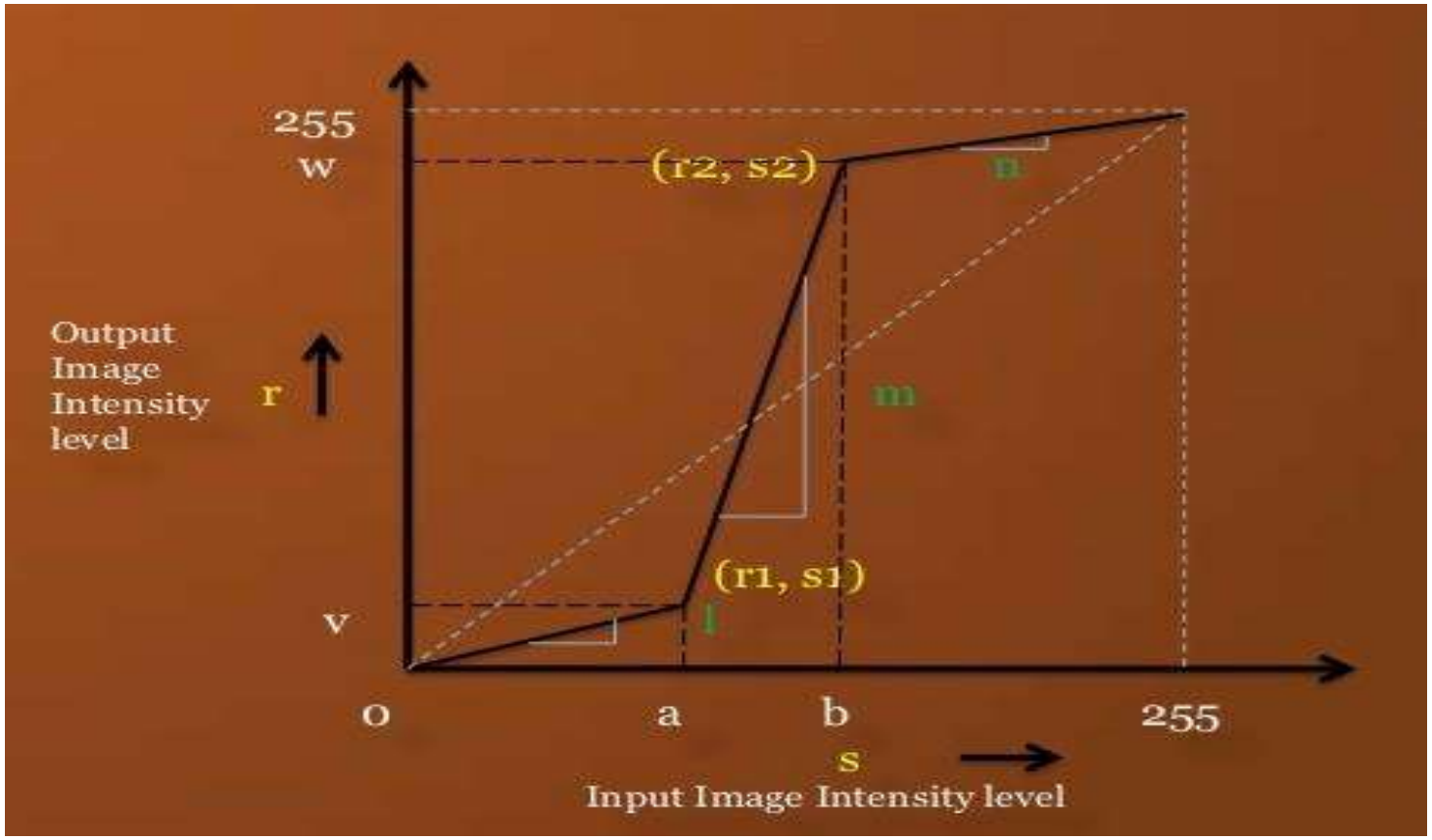
Contrast Stretching

- Contrast basically the difference between the intensity values of darker and brighter pixels .
- Contrast stretching expands the range of intensity levels in an image.

- Contrast stretching is done in three ways:
  - Multiplying each input pixel intensity value with a constant scalar.

    Example: s=2*r
  - Using Histogram Equivalent
  - Applying a transform which makes dark portion darker by assigning slope of < 1 and bright portion brighter by assigning slope of > 1.

Contrast Stretching (Using Transfer Function)



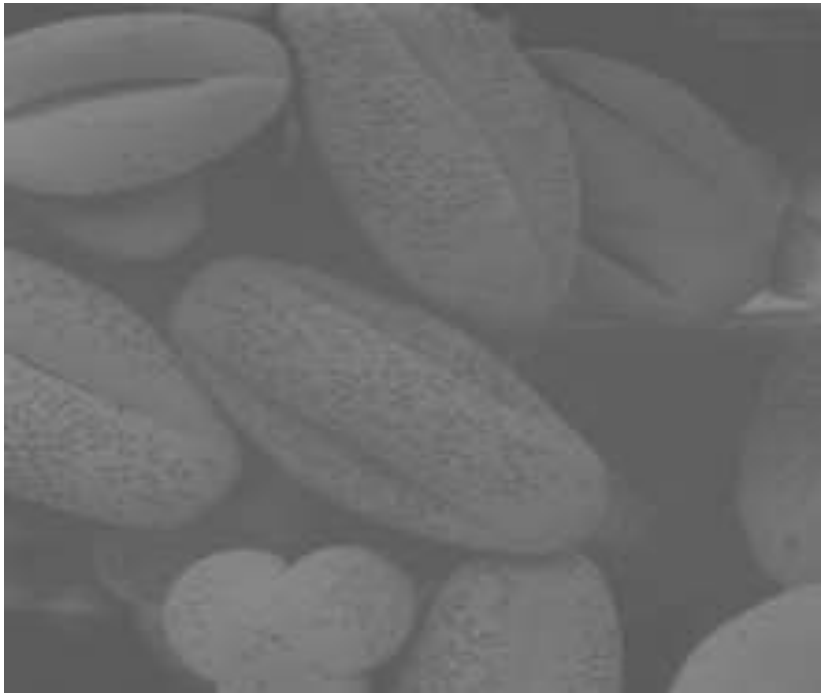
Formulation is given below:

s = l*r ; for 0 <= r <= a

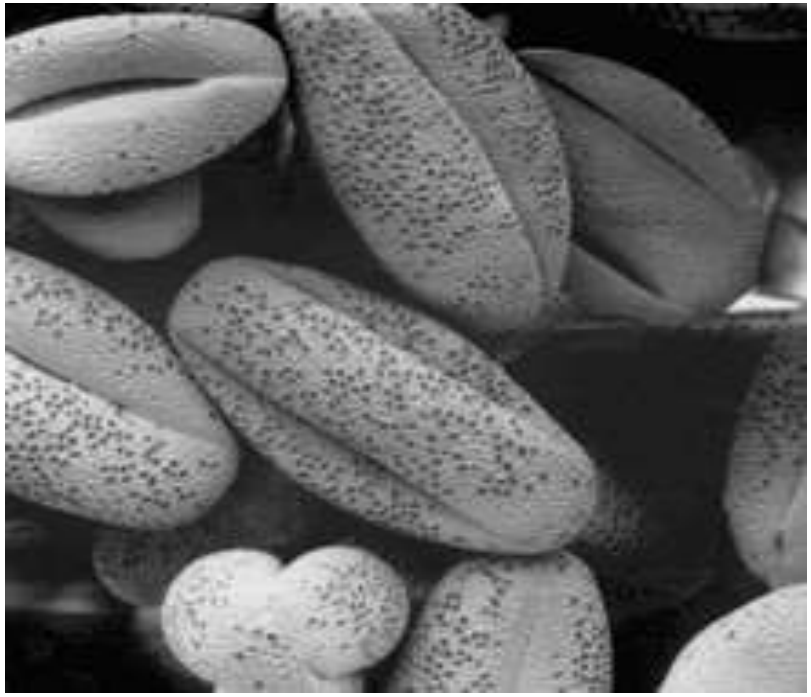
= m(r-a) + v ; for a < r <= b

= n(r-b) + w ; for b < r

Contrast Stretching



Original Image



Contrast Enhanced Image

Contrast Stretching

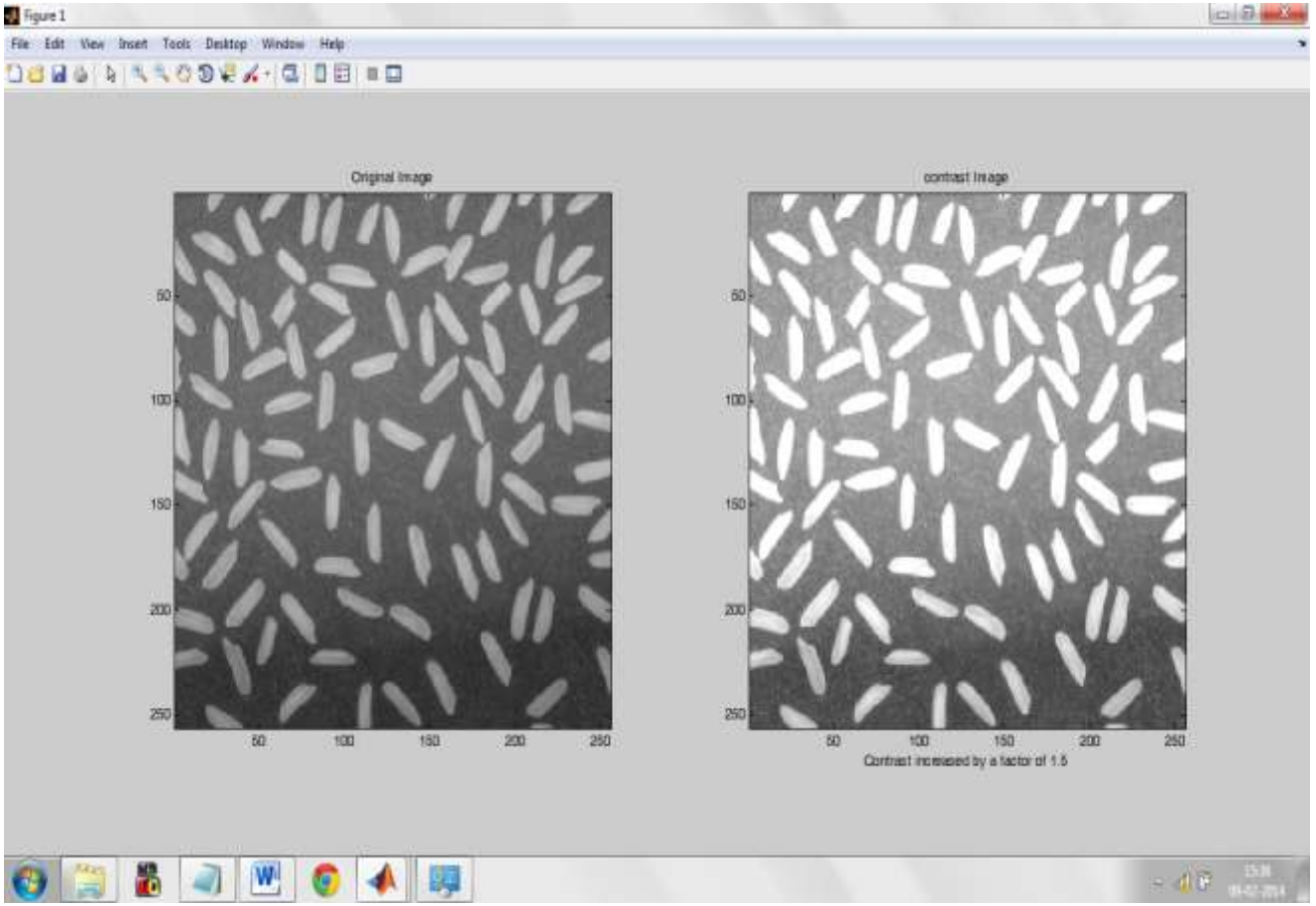
Contrast Stretching By Multiplication of each pixel with a scalar.

Matlab code:

```
% program to increase the contrast of an image
x=input('Enter the factor which contrast should be
increased');
a=imread('clown.png');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      b(i,j)=a(i,j)*x;
   end
end
subplot(1,2,1),subimage(a),title('Original Image');
subplot(1,2,2),subimage(b),title('contrast
Image'),xlabel(sprintf('Contrast increased by a factor of
%g',x));
```

Contrast Stretching

Output screen:



Contrast Stretching

Contrast Stretching by using threshold function.

Matlab code:

```
% program to increase the contrast of an image
x=input('Enter the factor which contrast should be
increased');
a=imread('clown.png');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      if(a(i,j)<50)
         b(i,j)=a(i,j);%slope of transfer function between i/p
and o/p is 1
      elseif (a(i,j)>200)
         b(i,j)=a(i,j);
         else
      b(i,j)=a(i,j)*x;%slope of transfer function between i/p
and o/p is x.Hence contrast of some particular pixel value
range increased
```

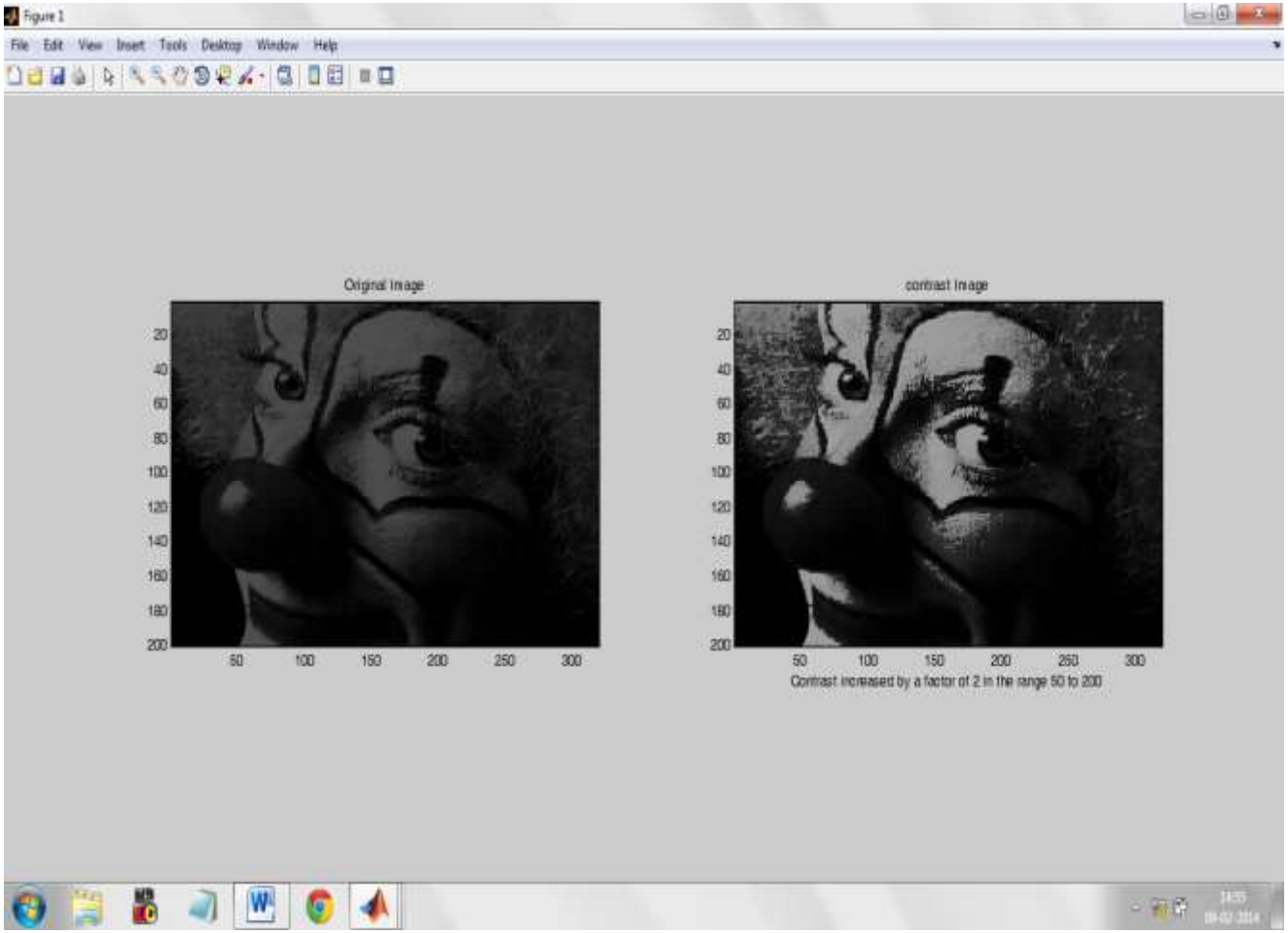
Contrast Stretching

Matlab code:

```
 end
   end
end
subplot(1,2,1),subimage(a),title('Original Image');
subplot(1,2,2),subimage(b),title('contrast
Image'),xlabel(sprintf('Contrast increased by a factor of %g
in the range 50 to 200',x));
```

Contrast Stretching

Output screen:



Contrast Stretching

Contrast Stretching By Histogram Equalisation.

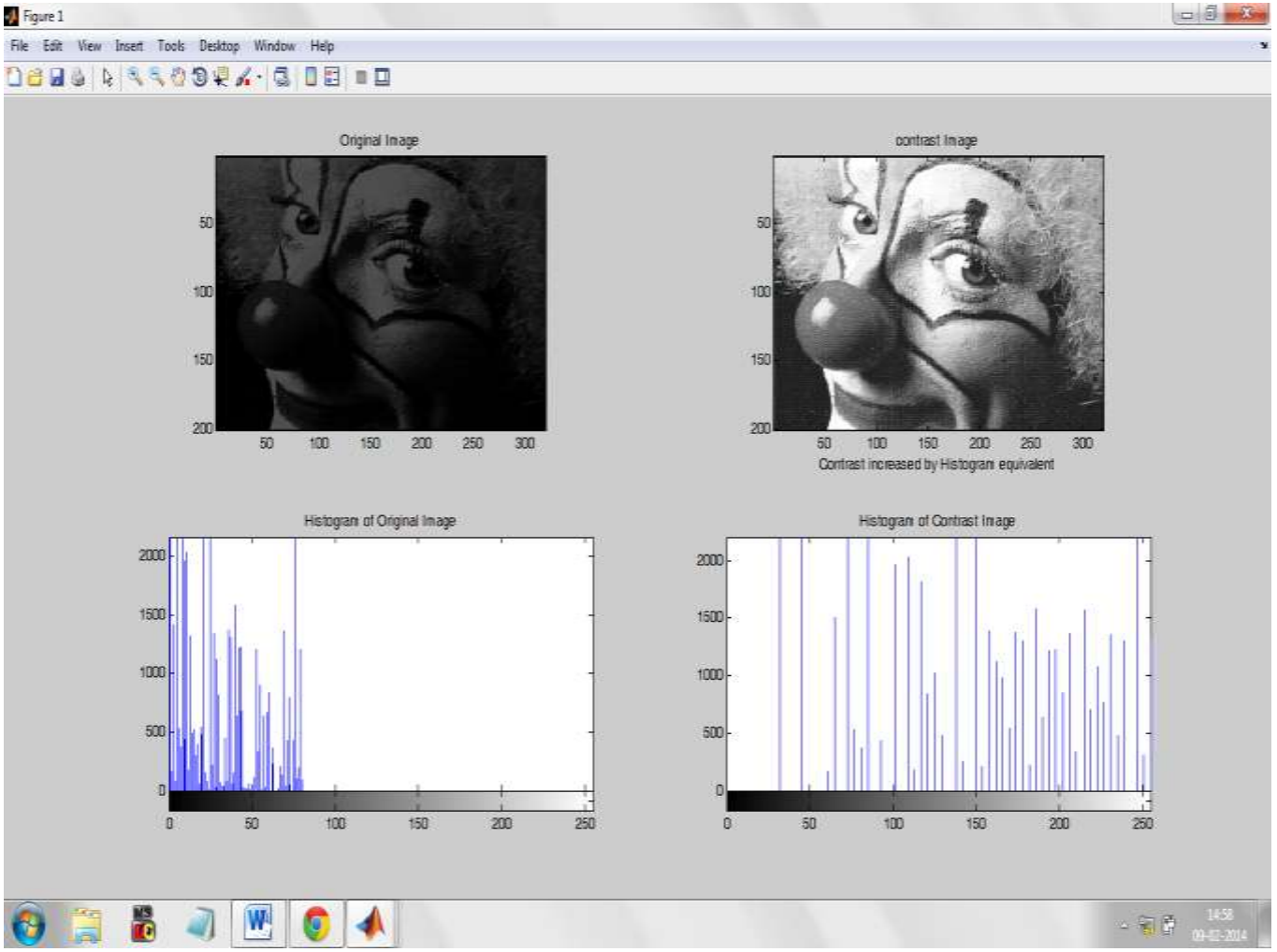
Matlab code :

```
% program to increase the contrast of an image by histogram
equivalent

a=imread('clown.png');
b=histeq(a);
subplot(2,2,1),subimage(a),title('Original Image');
subplot(2,2,2),subimage(b),title('contrast
Image'),xlabel(sprintf('Contrast increased by Histogram
equivalent'));
subplot(2,2,3),imhist(a),title('Histogram of Original Image');
subplot(2,2,4),imhist(b),title('Histogram of Contrast Image');
```

Contrast Stretching

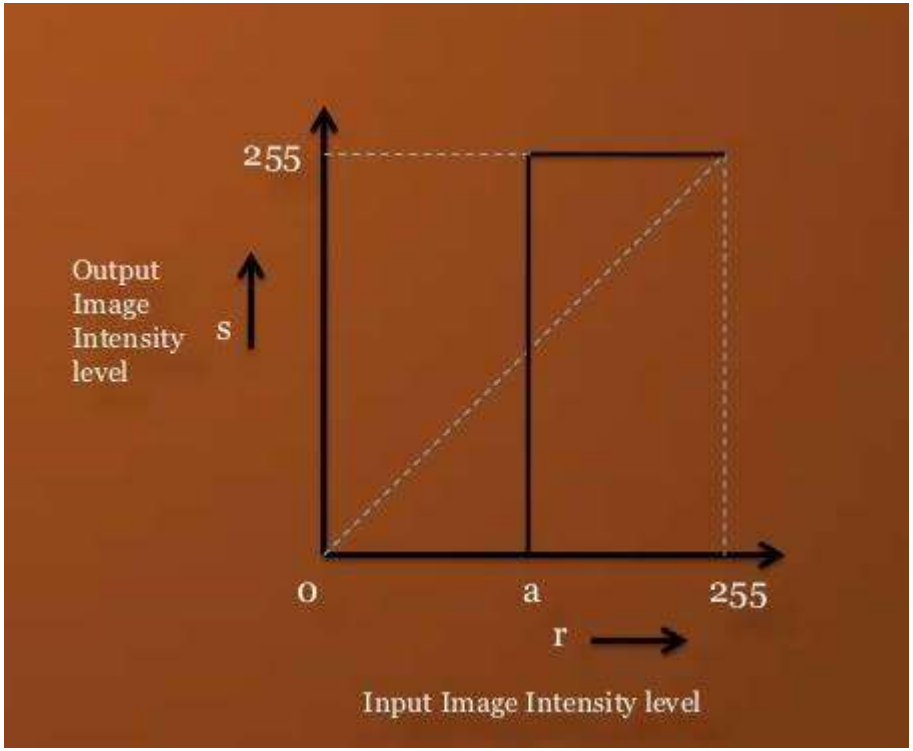
Output screen :



Thresholding

- Extreme Contrast Stretching yields Thresholding.
- Thresholded image has maximum contrast as it has only BLACK & WHITE gray values.
- In Contrast Stretching figure, if l & n slope are made ZERO & if m slope is increased then we get Thresholding Transformation
- If r1 = r2, s1 = 0 & s2 = L-1 ,then we get Thresholding function.

Thresholding Function



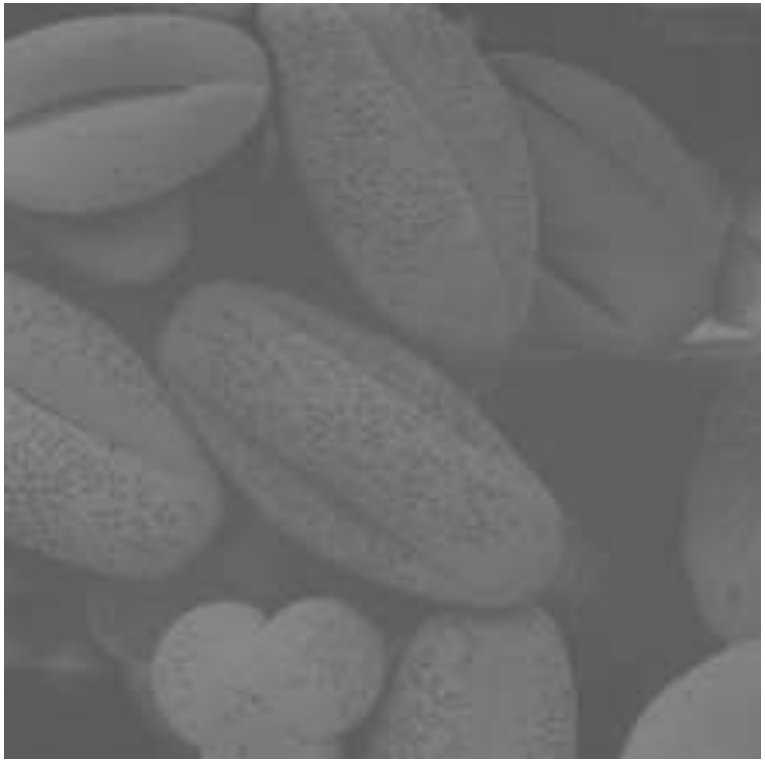
Expression goes as under:

$s = 0$; if $r = a$

$s = L - 1$; if $r > a$

where, L is number of gray levels.

Thresholding



Original Image

Transformed Image

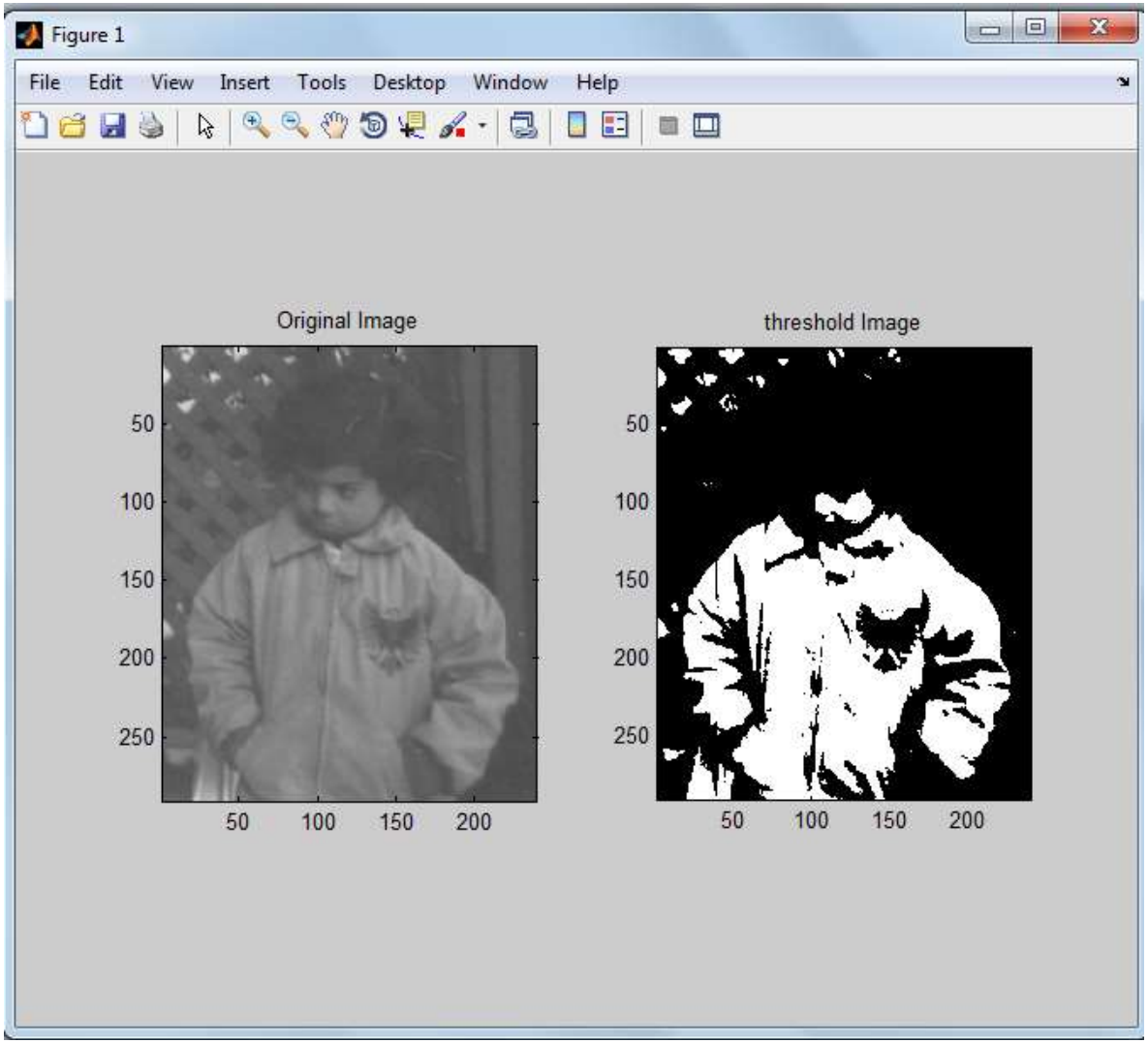
Thresholding

Matlab code:

```
% program for image thresholding
a=imread('pout.tif');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      if(a(i,j)<125)
         b(i,j)=0;%pixel values below 125 are mapped to zero
         else
      b(i,j)=255;%pixel values equal or above 125 are mapped to 255
      end
   end
end
subplot(1,2,1),subimage(a),title('Original Image');
subplot(1,2,2),subimage(b),title('threshold Image');
```

Thresholding

Output screen:



Brightness Enhancement

- Brightness Enhancement is shifting of intensity values to a higher level.
- The darker and the lighter pixels both get their values shifted by some constant value.
- Example : In x-ray images brightness can be enhanced to find the darker spots.

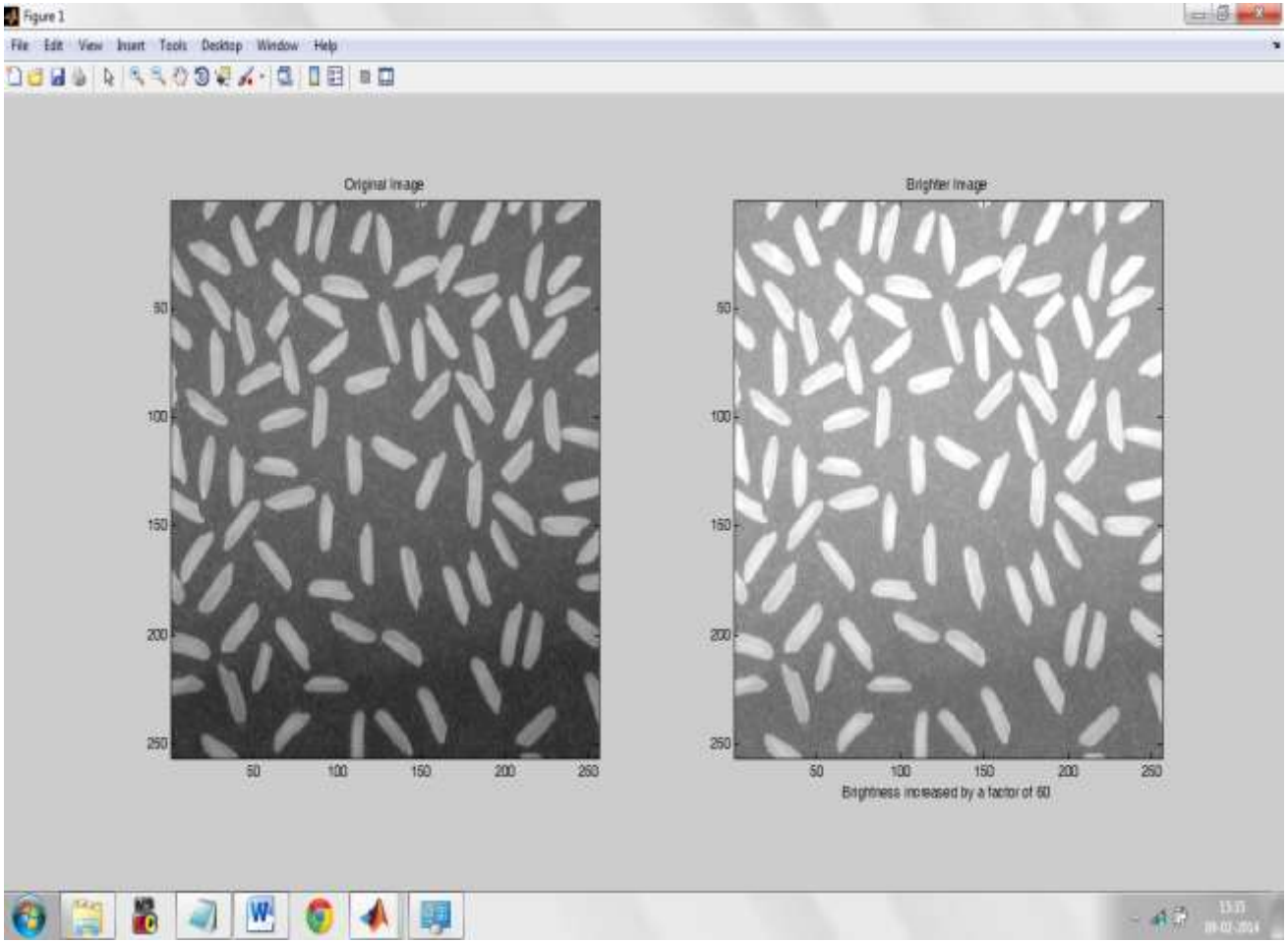
Brightness Enhancement

Matlab code :

```
% program to increase the brightness of an image
x=input('Enter the factor which brightness should be increased');
a=imread('clown.png');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      b(i,j)=a(i,j)+x;
   end
end
subplot(1,2,1),subimage(a),title('Original Image');
subplot(1,2,2),subimage(b),title('Brighter
Image'),xlabel(sprintf('Brightness increased by a factor of %g',x));
```

Brightness Enhancement

Output screen:



Log Transformation

- The log transformation is given by the expression

$$s = c \log(1 + r)$$

where c is a constant and it is assumed that $r \geq 0$.

- This transformation maps a narrow range of low-level grey scale intensities into a wider range of output values.
- Similarly maps the wide range of high-level grey scale intensities into a narrow range of high level output values.
- This transform is used to expand values of dark pixels and compress values of bright pixels.

Logarithmic Transformation Contd…

Original Image

Transformed Image

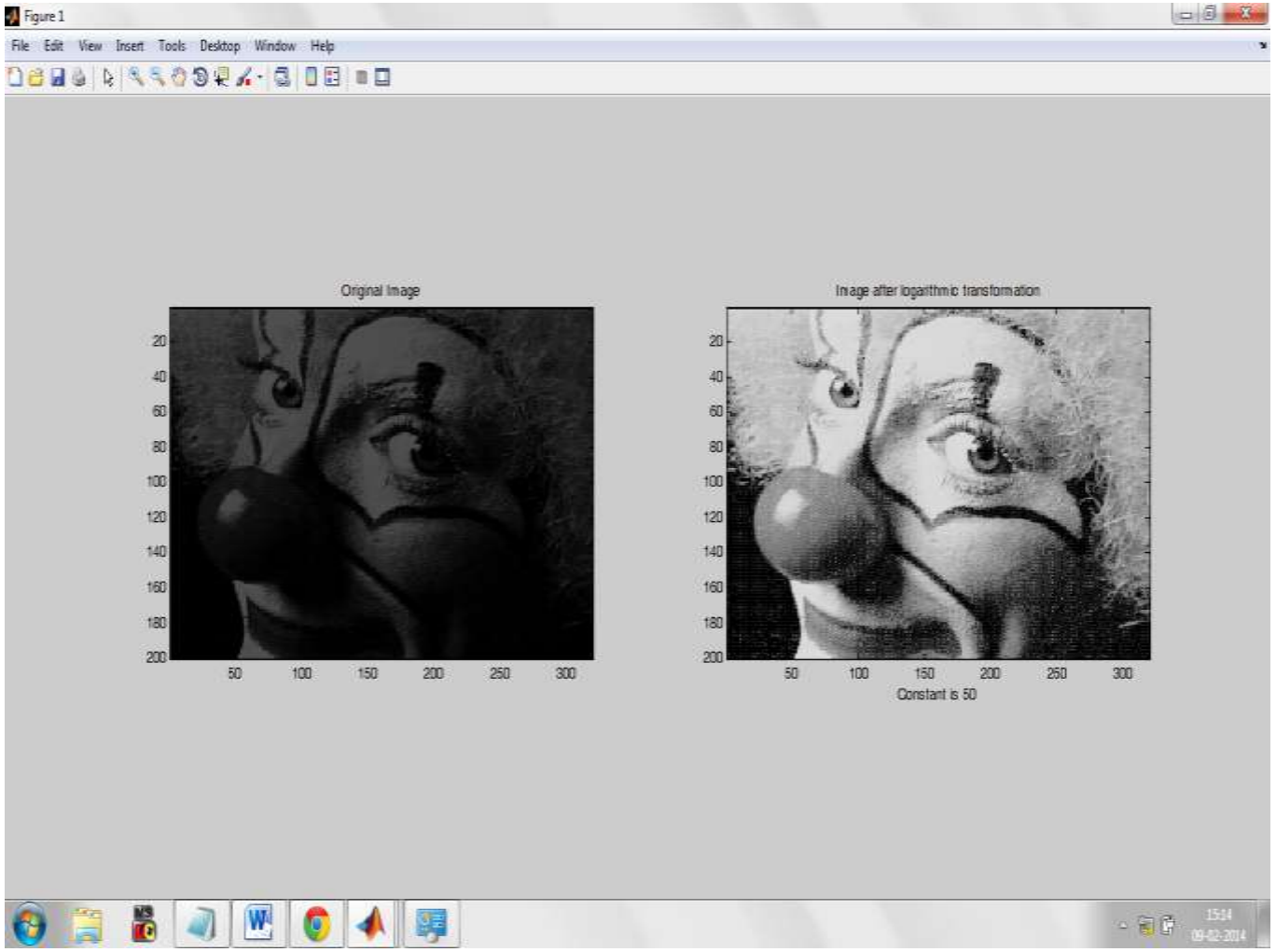
Log Transformation

Matlab code:

```
% program for image enhancement using logarithmic
transformation
A=input('Enter the value of constant A');
a=imread('clown.jpg');
a=rgb2gray(a);
[m,n]=size(a);
a=double(a);
for i=1:1:m
   for j=1:1:n
      b(i,j)=A*log(1+a(i,j));
   end
end
figure,subplot(1,2,1),subimage(uint8(a)),title('Original Image');
subplot(1,2,2),subimage(uint8(b)),title('Image after logarithmic
transformation'),xlabel(sprintf('Constant is %g',A));
```

Log Transformation

Output screen:



Power Law Transformation

Expression for power law transformation is given by:

$$s = c * (r^{\gamma})$$

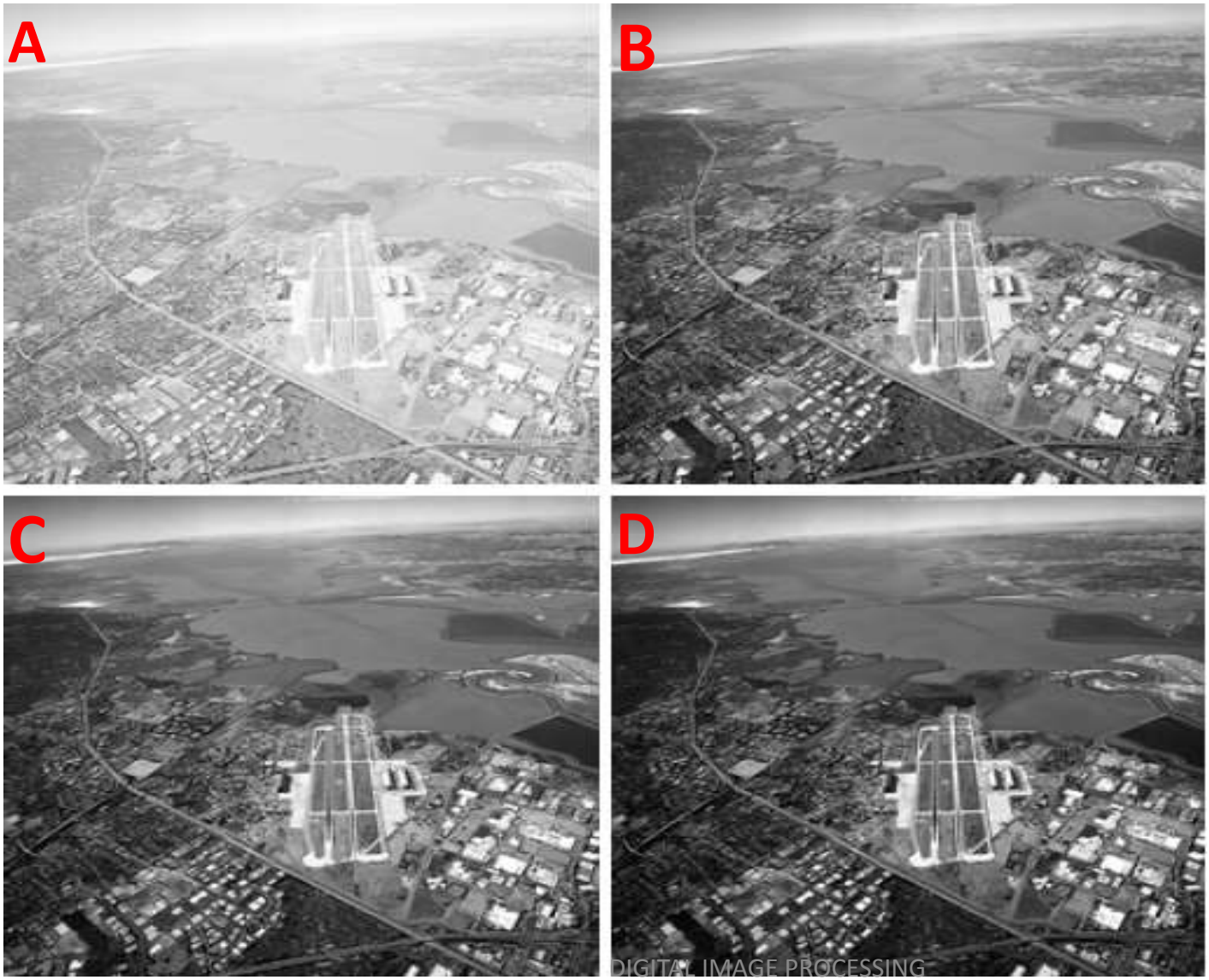
s is the output pixels value.
r is the input pixel value.
c and γ are real numbers.

- For various values of γ different levels of enhancements can be obtained.
- This technique is quite commonly called as *Gamma Correction ,* used in monitor displays.

Power Law Transformation

- Different display monitors display images at different intensities and clarity because every monitor has built-in gamma correction in it with certain gamma ranges .
- A good monitor automatically corrects all the images displayed on it for the best contrast to give user the best experience.
- The difference between the log-transformation function and the power-law functions is that using the power-law function a family of possible transformation curves can be obtained just by varying the γ .

Power Law Transformation



A : original image
For c=1
B : γ =3.0
C : γ =4.0
D : γ =5.0

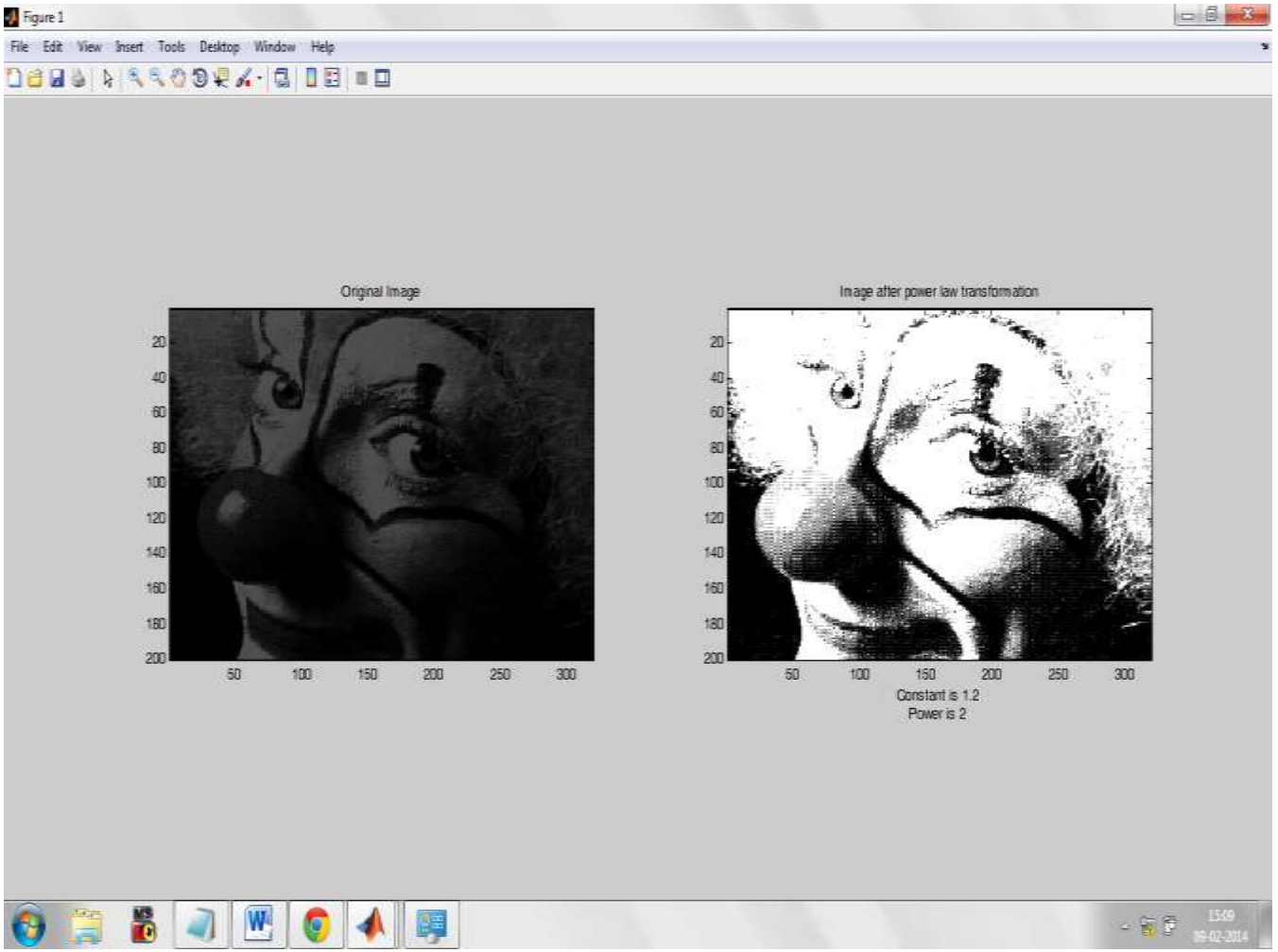
Power Law Transformation

Matlab code:

```
% program for image enhancement using power law
A=input('Enter the value of constant A');
x=input('Enter the value of power x');
a=imread('clown.png');
[m,n]=size(a);
for i=1:1:m
   for j=1:1:n
      b(i,j)=A*(a(i,j)^x);
   end
end
subplot(1,2,1),subimage(a),title('Original Image');
subplot(1,2,2),subimage(b),title('Image after power law
transformation'),xlabel(sprintf('Constant is %g\nPower is
%g',A,x));
```

Power Law Transformation

Output screen:



References:

- Digital Image Processing by Gonzalez And Woods.
- Wikipedia
- Matlab Help

THANK YOU