Lab 1 : Basic Implementation of Image Processing

Author : Siddharth Nahar , 2016CSB1043

1.Resize function : Implemented using Backward mapping and two options are provided:

Input Image:

a.Nearest Neigbourhood:

*ScaleX=0.5,ScaleY=0.5,PSNR=30.2672

*Inbuilt=resize(src,dst,Size(),fx ,fy,INTER_NEAREST);

*In inbuilt ,0.5 is rounded to 0 rather than 1 So PSNR is not exact 0.

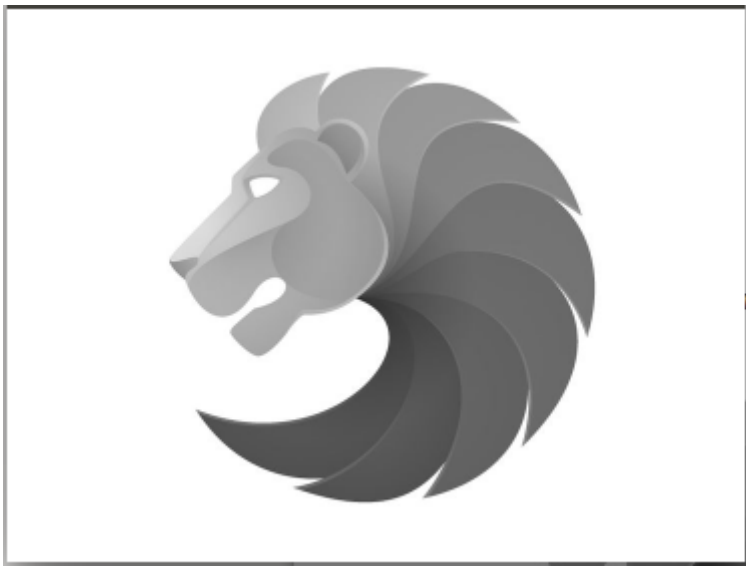
b.Bilinear:

Resize my implementation:

ScaleX = 0.5,ScaleY = 0.5,PSNR=0

Inbuilt=resize(.....,INTER_LINEAR)

Inbuilt function Output(BILINEAR):



**For Inter cubic PSNR=46.53*

****For Bilinear I observed :***

a.For Corner Points ,inbuilt not gives same value,it takes only fraction which is effective.

b.Inbuilt floors size if rows*fx is not integer,So PSNR is calculted for 0.5

2.Translate the image:

By Implementation:

Traverse x = 20,y=20;

**PSNR = 0,As in case of both forward or backward*

Result remains same.

By Inbuilt:

Function = warpAffine(src,dst,m,Size(dst));

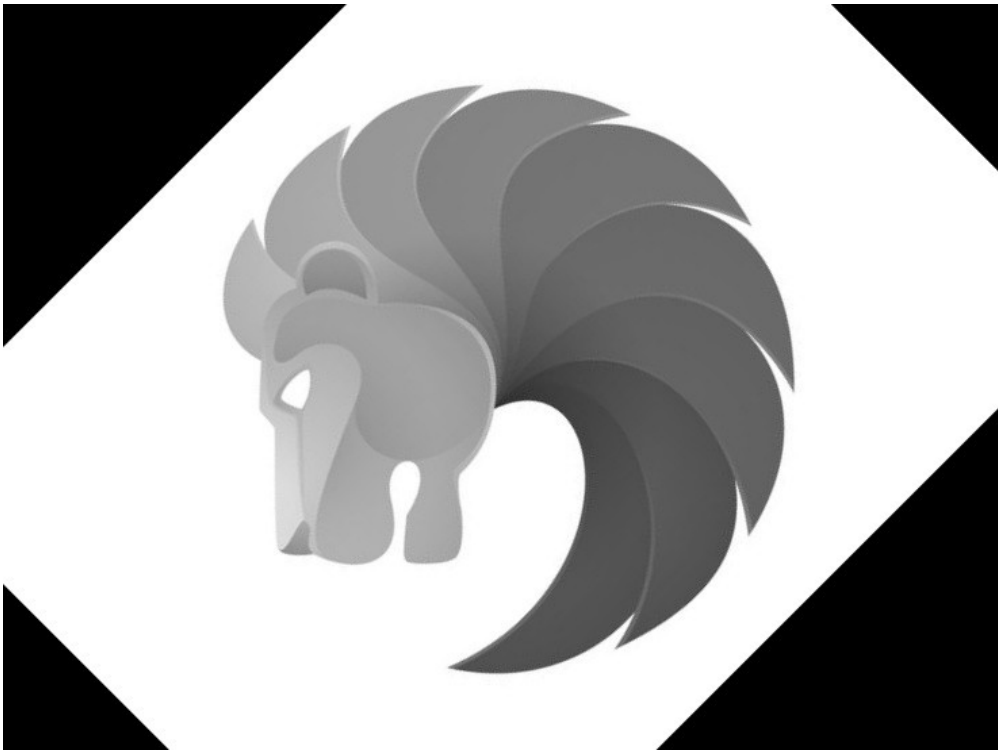
*m=2*3 matrix of translation*

Output:

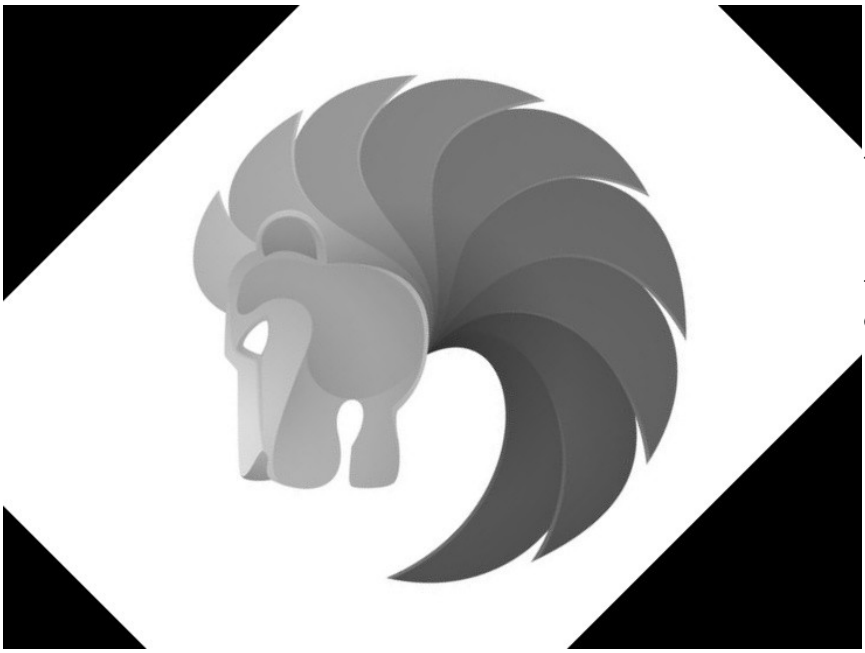
3.Rotation of Image:

By Implemnetation:

Angle = 45,Bilinear,PSNR = 65.5325



By Inbuilt:



**Backward Mapping is very important.*

**Dots appear In image,because rotation changes orientation*

**In bilinear implementation,corner cases are considered differently.ex. For only 2 nearest ,it takes interpolation for both then multiply effective ration to result.*

4.Shear:

By implementation:

shear x = 0.5,PSNR=26.452

Inbuilt:

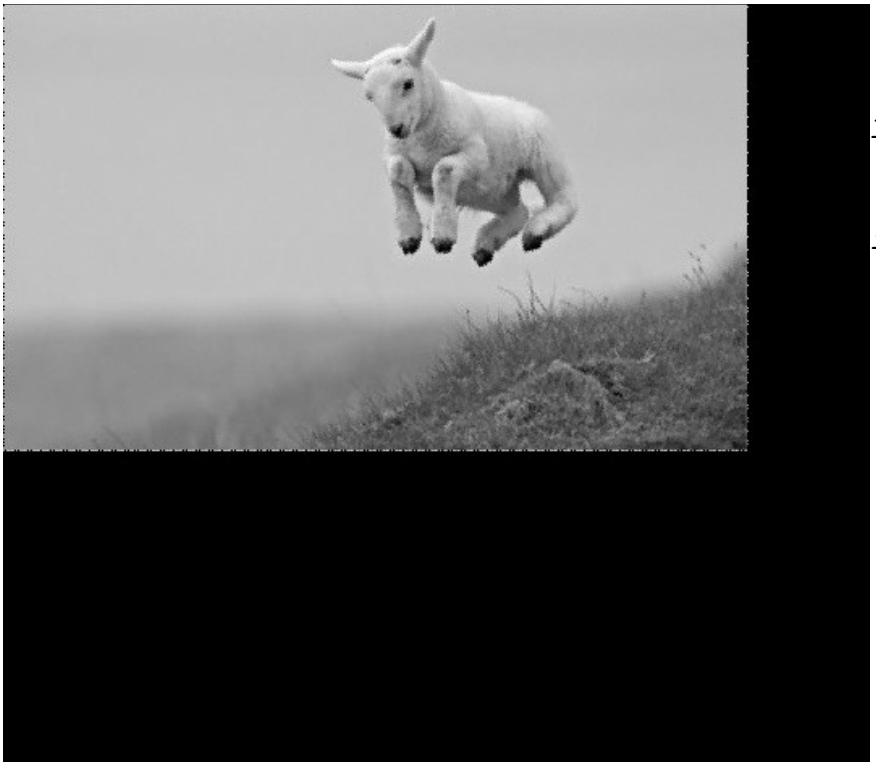
5.Recounstruction with Tie points:

Input Image:

Points of distoted :(13,395),(233,14),(235,523),(455,142)

Point of original:(10,450),(10,10),(266,450),(266,10)

Output Image:



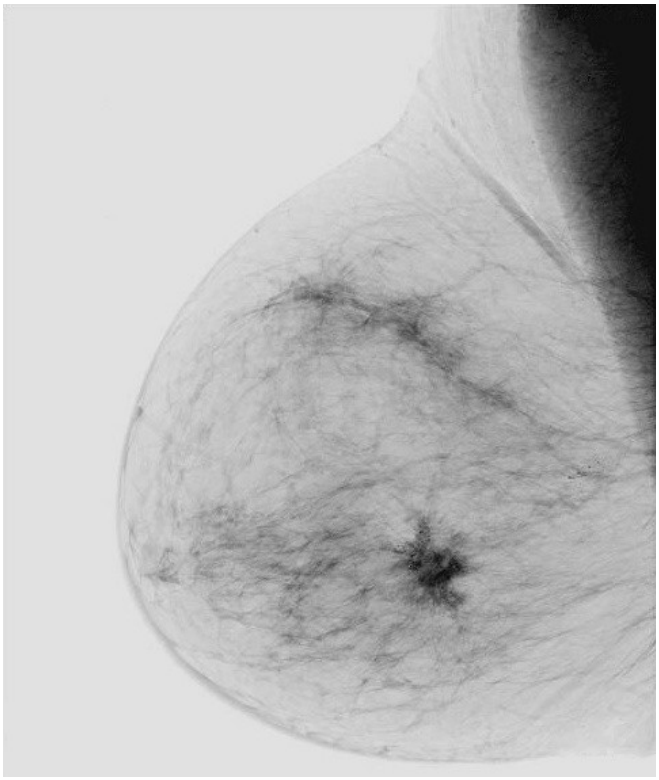
**Kept the size same as Input image.*

**It has reconstructed the image which was rotated.*

6.Image Negative:

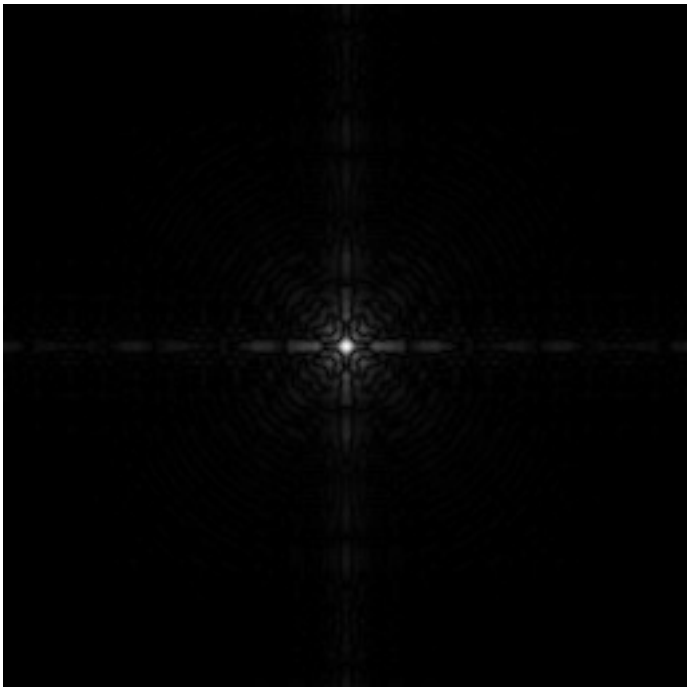
Input Image:

Output Image:

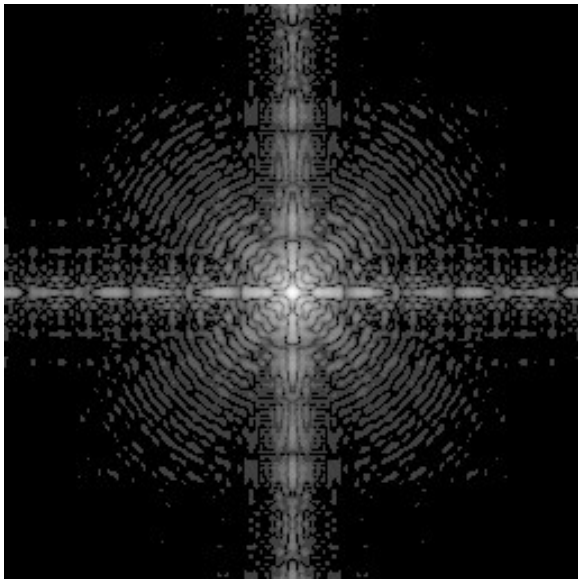


7.Log Transformation:

Input Image:



Output Image:



8.Power Transformation:

Input Image:

Gamma = 3.0

Output Image:

9.PieceWise Linear Transformation:

Input Image:

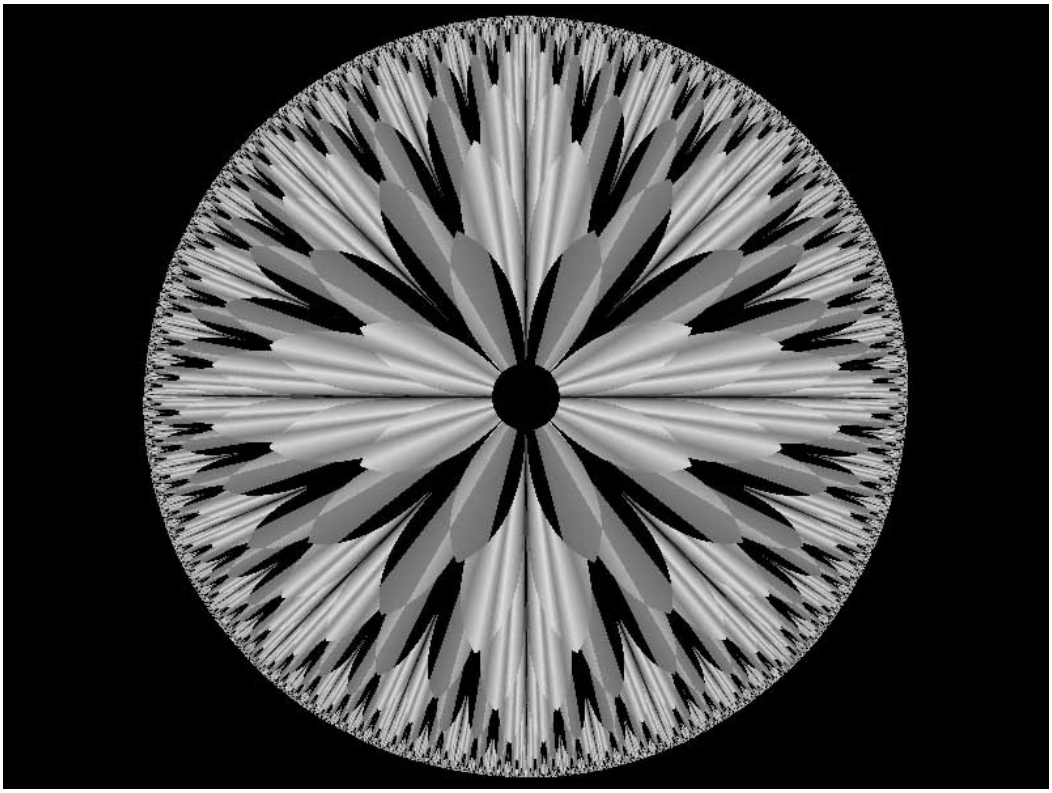
r1 = rmin,r2=rmax,s1=0,s2=255

Output Image:

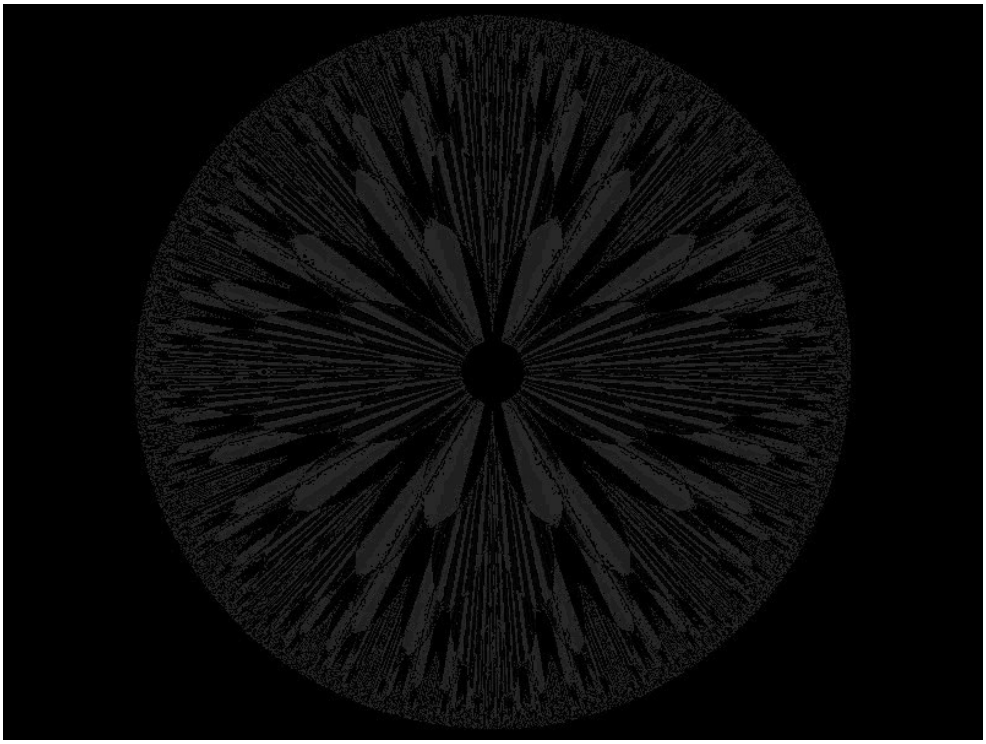
10.BitSlicing Transformation:

Input Image:

Planes = 3,5

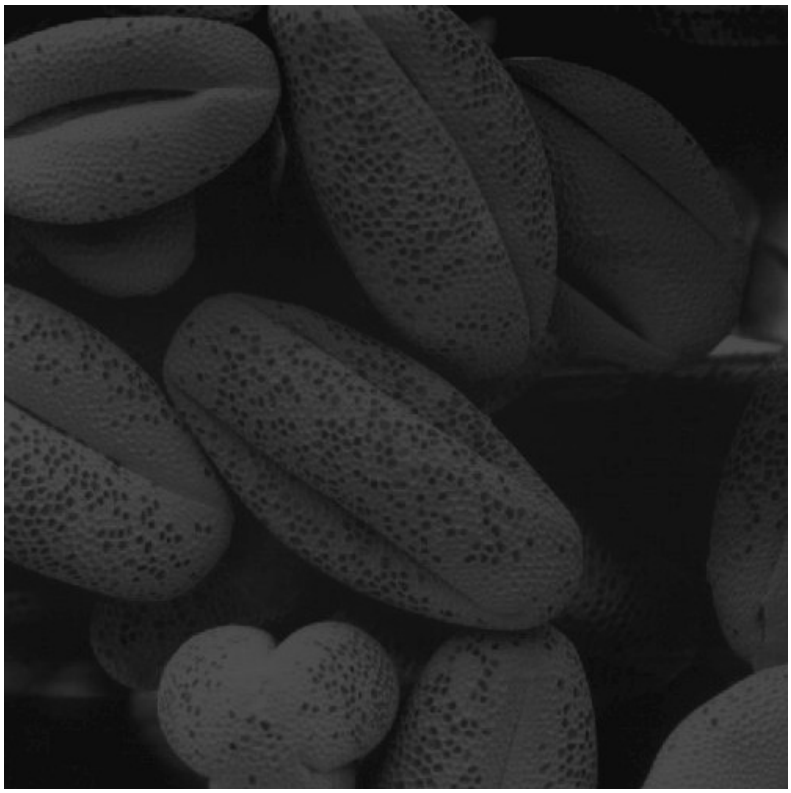


Output Image:



11.Histogram Equalization:

Input Image:



**PSNR = 0;*

**Normalization used :*

255(y-ymin)/(1-ymin)*

For more distribution of values

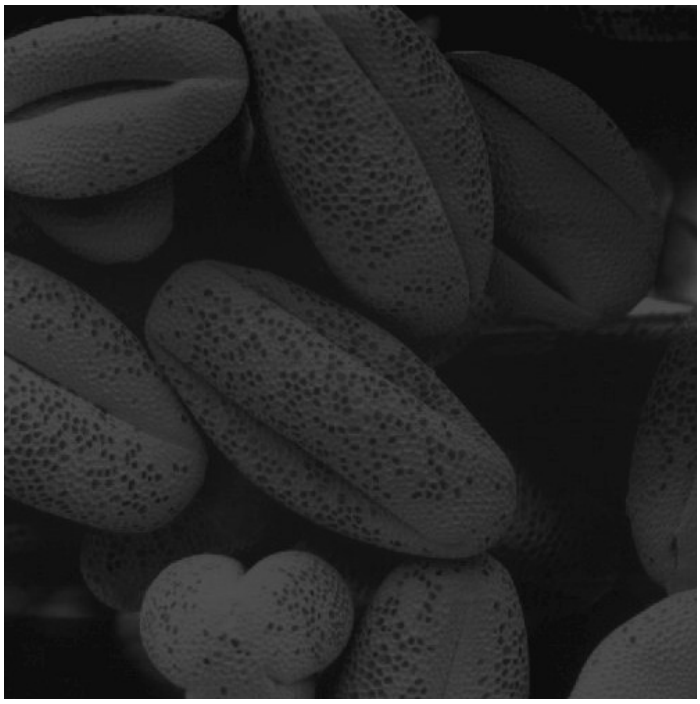
**equalizehist(src,dst);*

Output Image implementation:

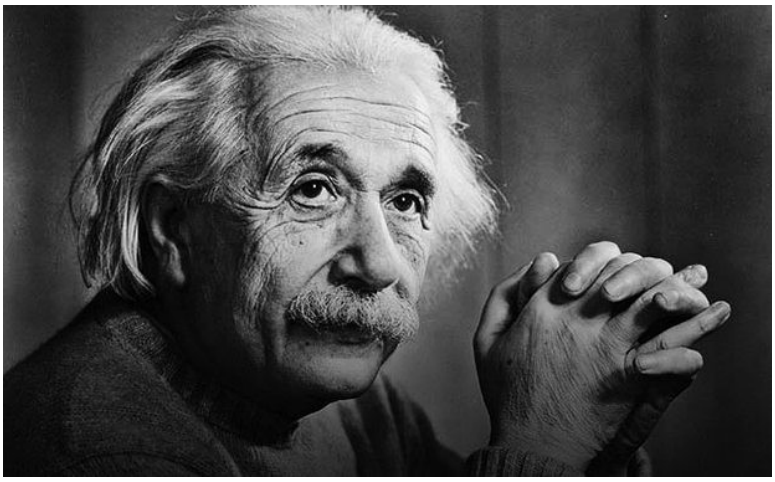
Inbuilt Implementation:

12.Matched Equalization:

Input Image 1:



Input Image 2:

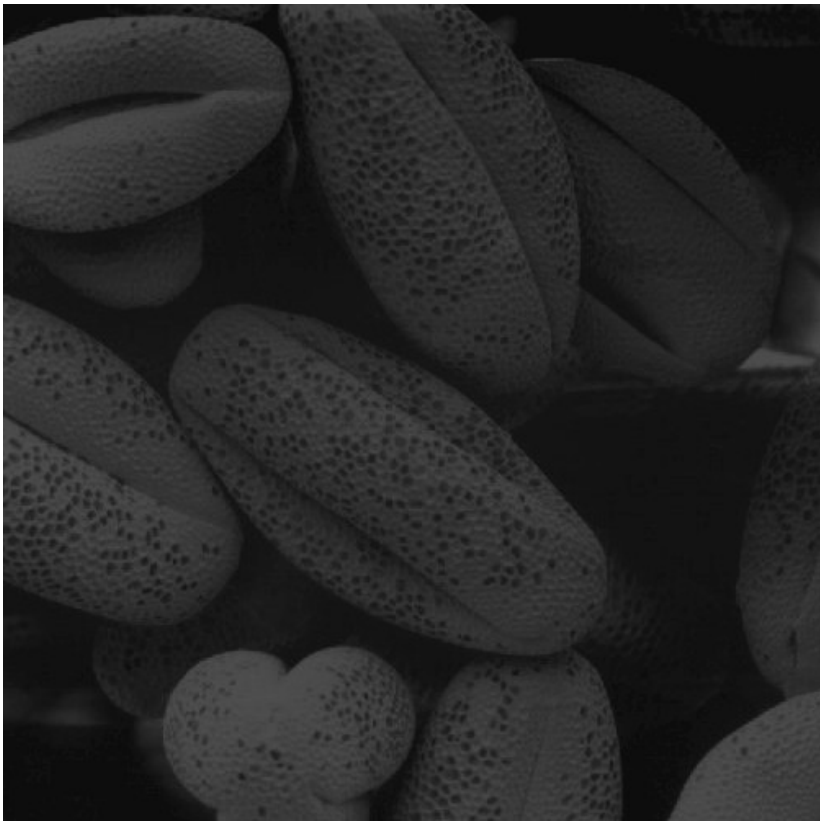


Outptut Image:

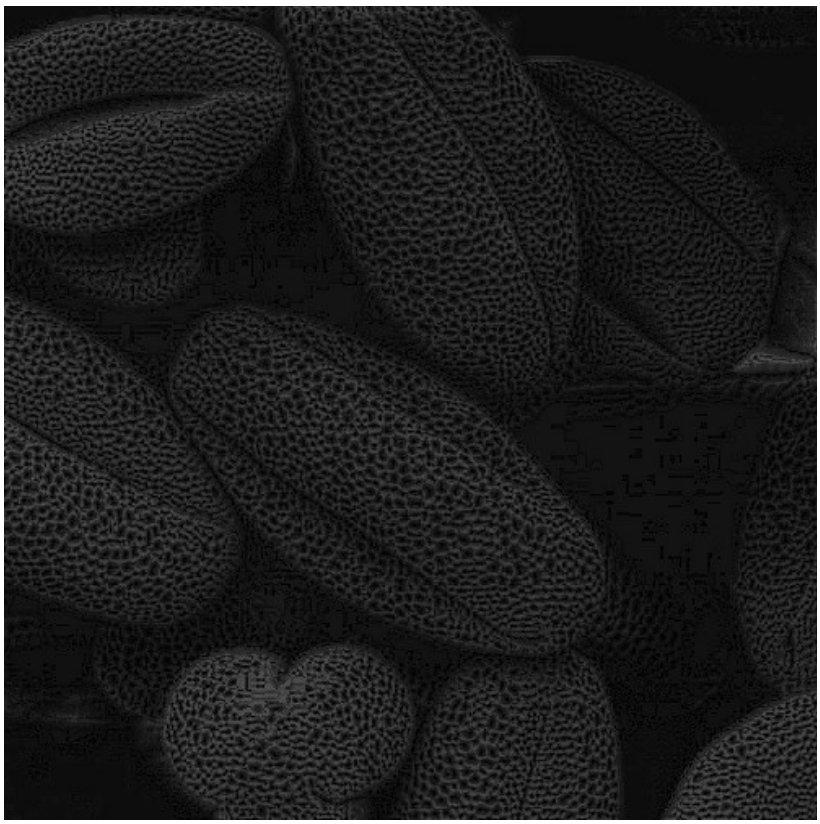
13.Adaptive Histogram Eqalization:

Input Image:

Congruence Square=3



Output Image:



**Use Mirror neighbours for Locaslized Equalization*

**Used Equalization in local region of3*3*

**Due to small region Enhancement is not proper.*

**Most effective to Enhance regions but time complexity m2*n2*