Faculty of Computers &
Artificial Intelligence

Benha University

Digital Image Processing Project

By

Abdallah Tamer Mohamed El Ghamry

SEC. 3

CS

January 2021

Graphical User Interface (GUI)

File

- **Open:** to browse an image.
- **Save:** to save the modified image.
- **Reset:** to undo all the operations performed on the image.

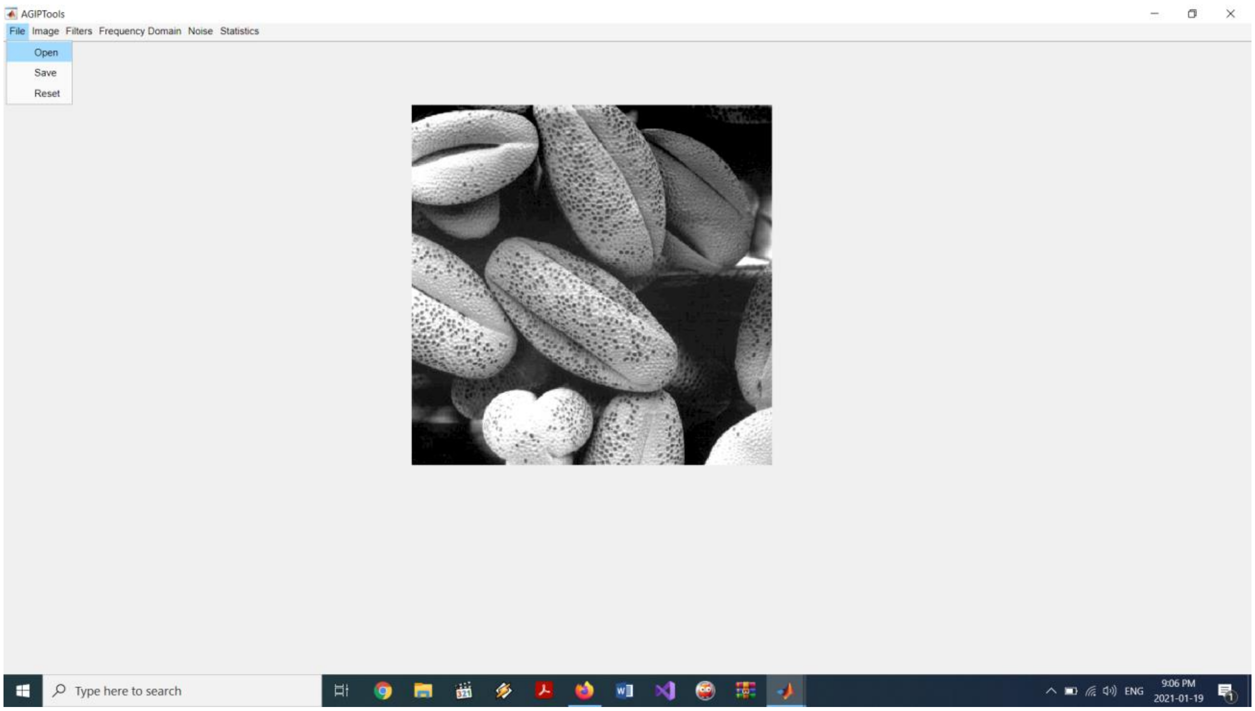


Image → Colors

Extracts red, green or blue channel from a RGB image.

- **Red**
- **Green**
- **Blue**

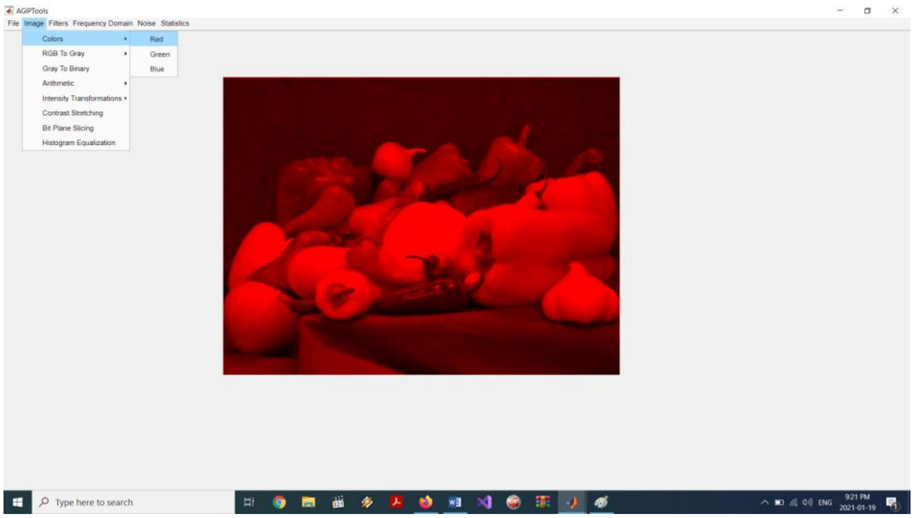
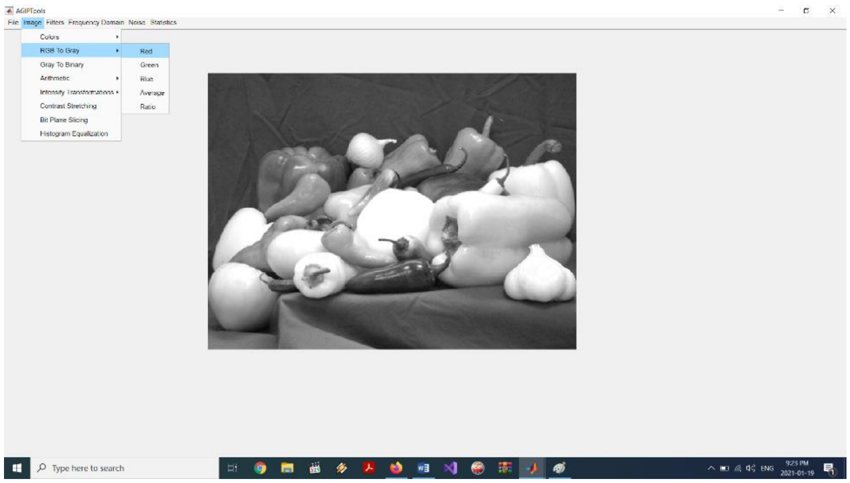


Image → RGB to Gray

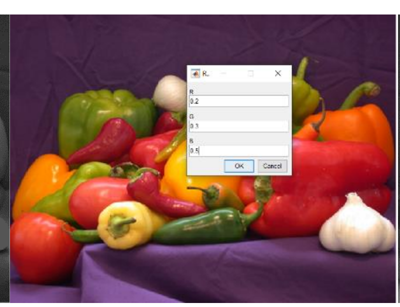
Convers a RGB image to a grayscale image using 5 methods.

- **Red:** Selects the red channel. RGB
- **Green:** Selects the green channel. RGB
- **Blue:** Selects the blue channel. RGB
- **Average:** Selects the red channel. (R + G + B) / 3
- **Ratio:** Selects the red channel. (r*R + g*G + b*B) such that (r + g + b) = 1





Red Green Blue

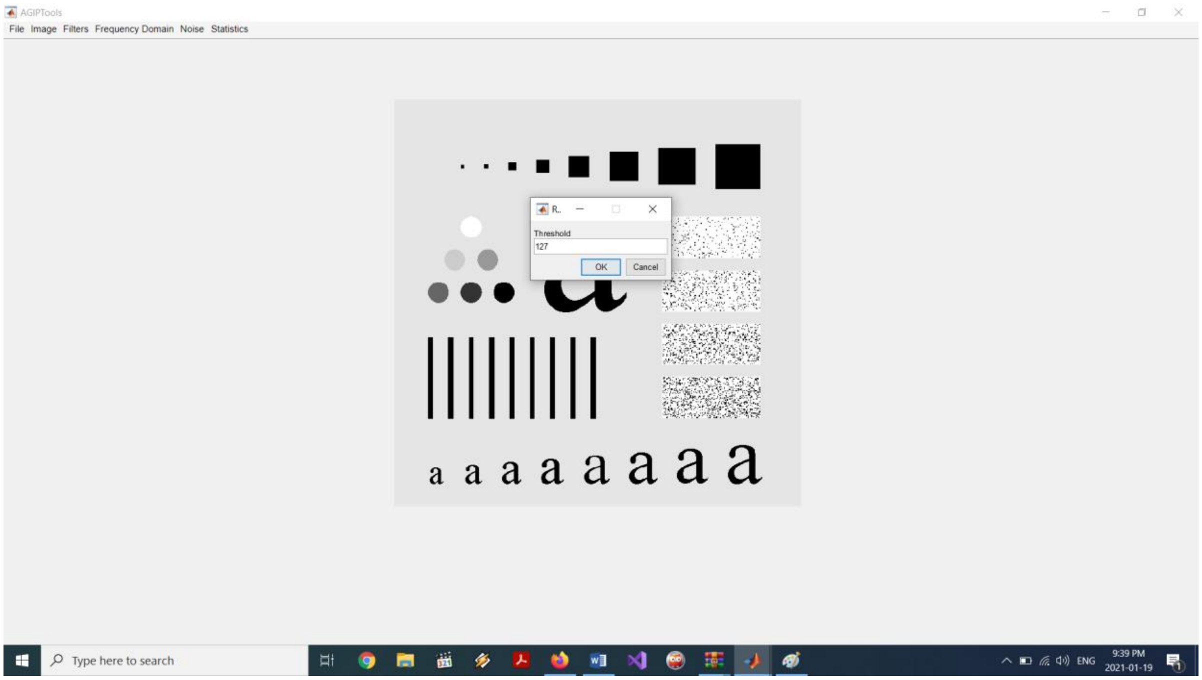




Average Ratio of r=0.2, g=0.3, b=0.5

Image → Gray to Binary

Convers a grayscale image to a binary (0, 1) image with a specific threshold.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}$$



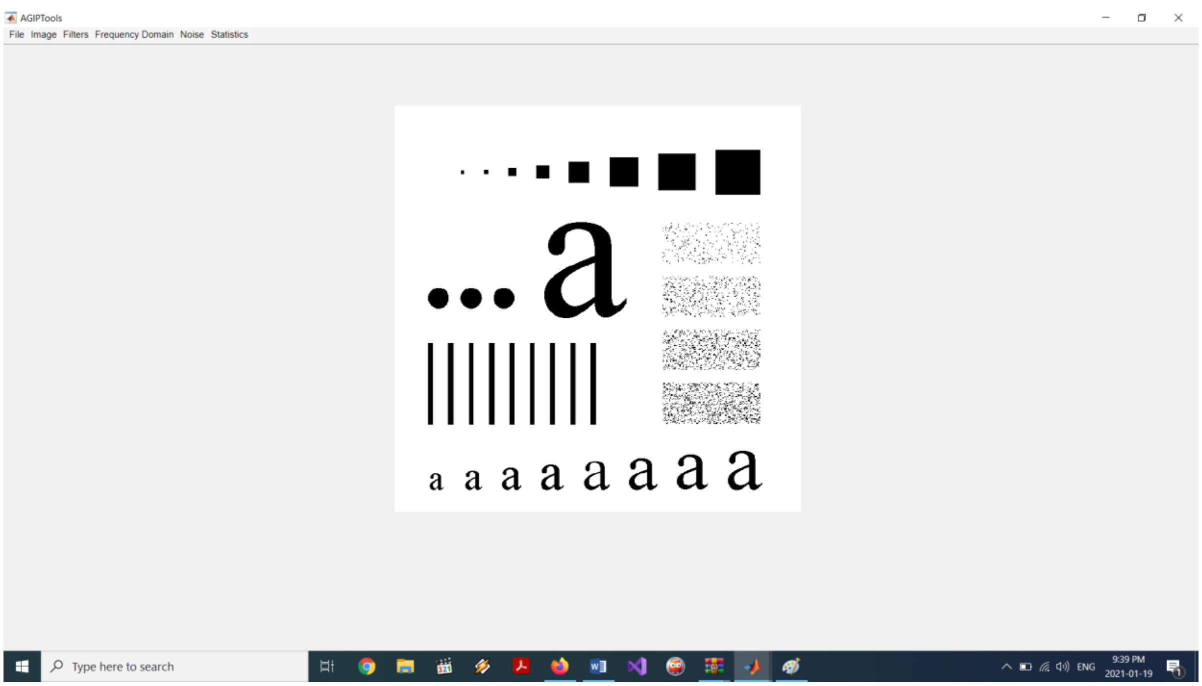
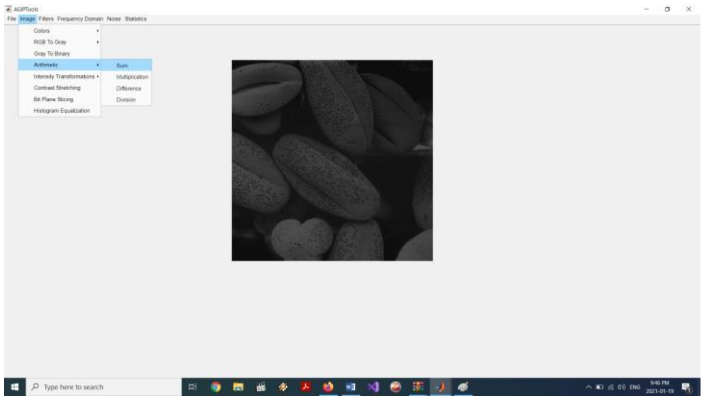
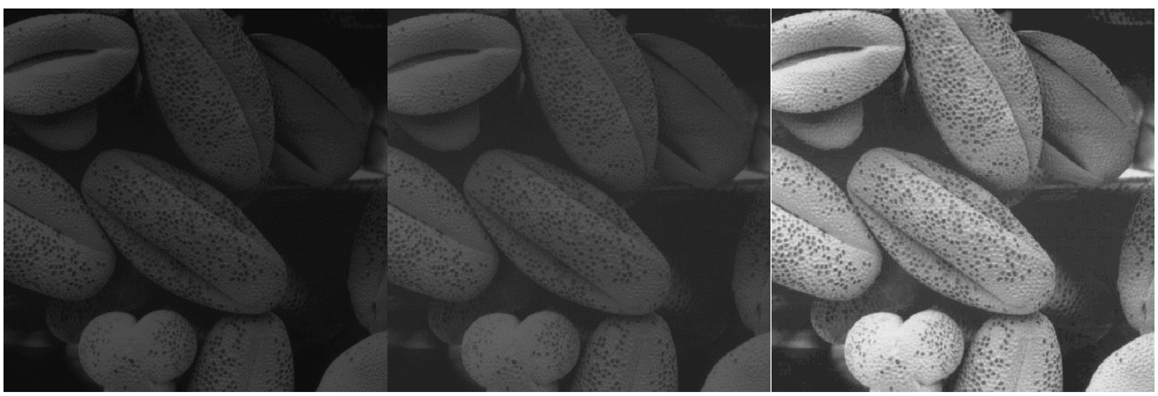


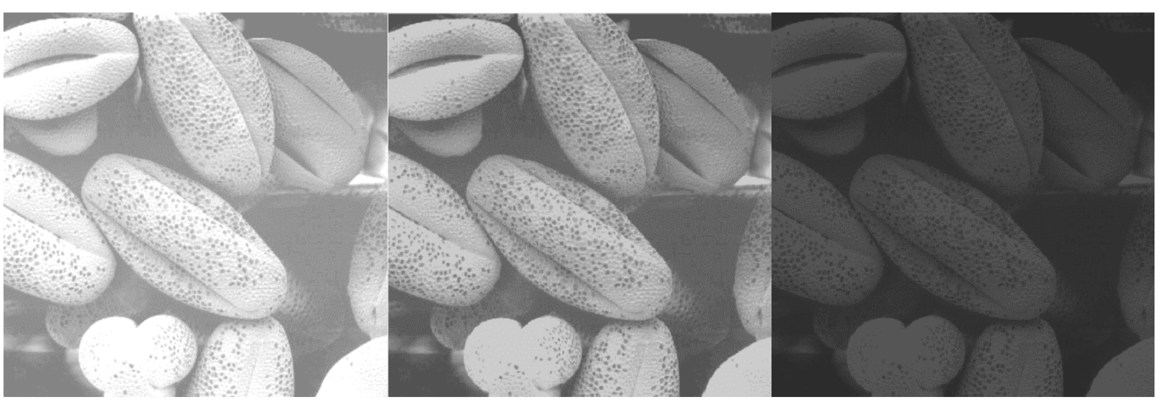
Image → Arithmetic

- **Sum:** Adds a specific value to the image.
- **Multiplication:** Multiply the image with a specific value.
- **Difference:** Subtract a specific value from the image.
- **Division:** Divide the image with a specific value.

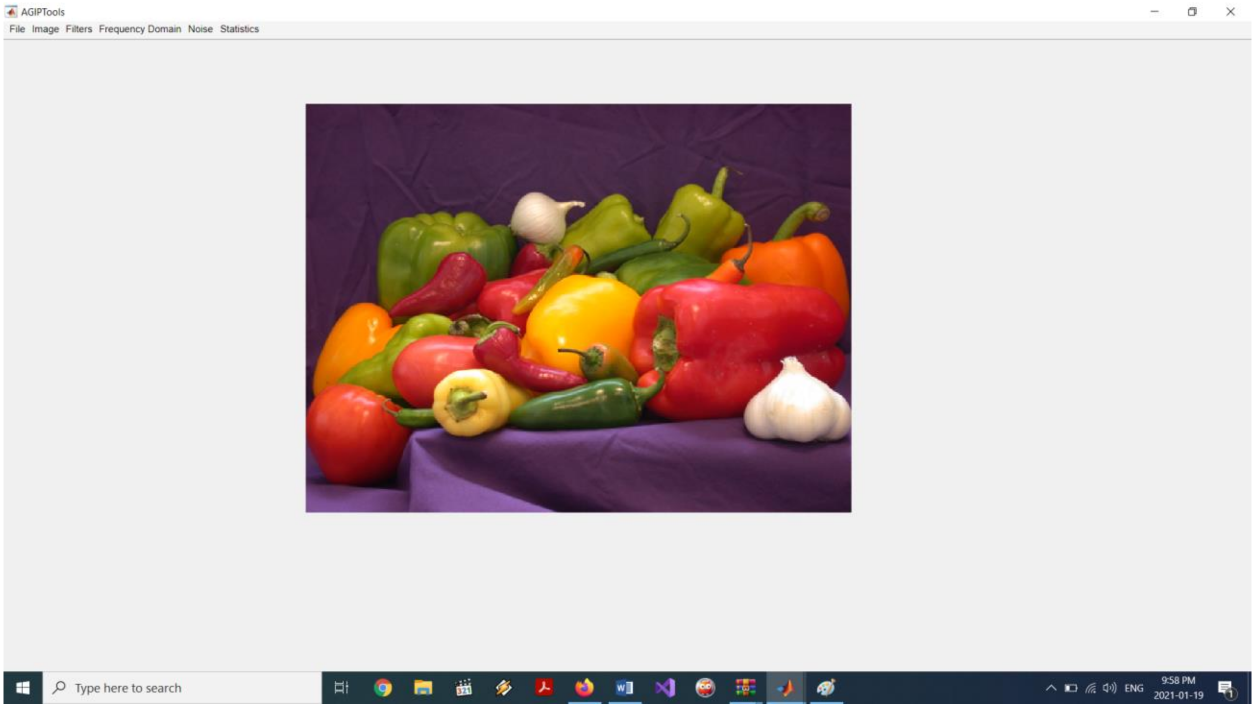


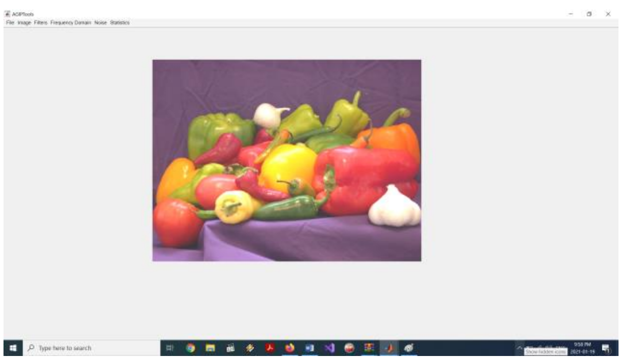


Dark Image | Add 20 | Multiply 3

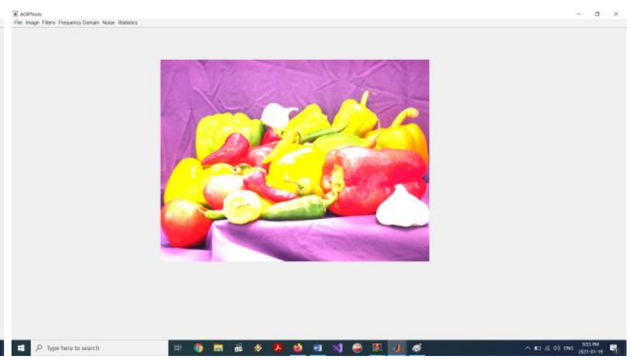


Light Image | Subtract 30 | Divide 3

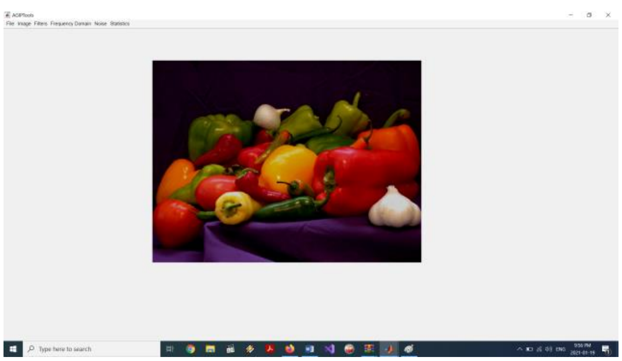




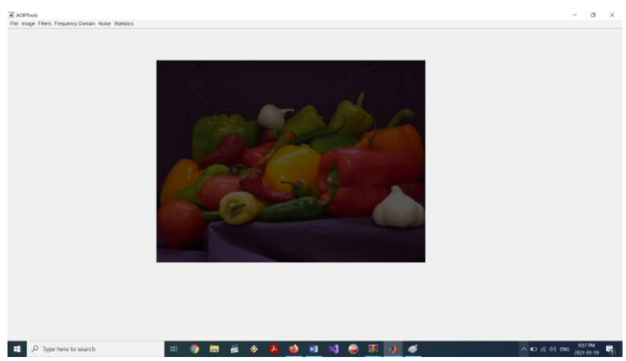
Add 50



Multiply 3



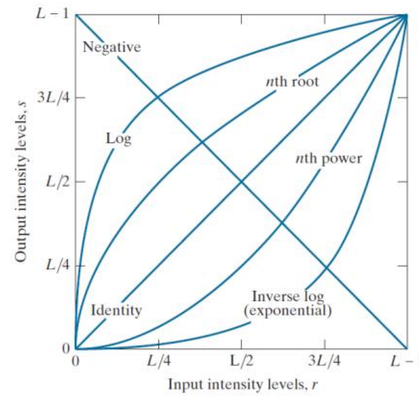
Subtract 50

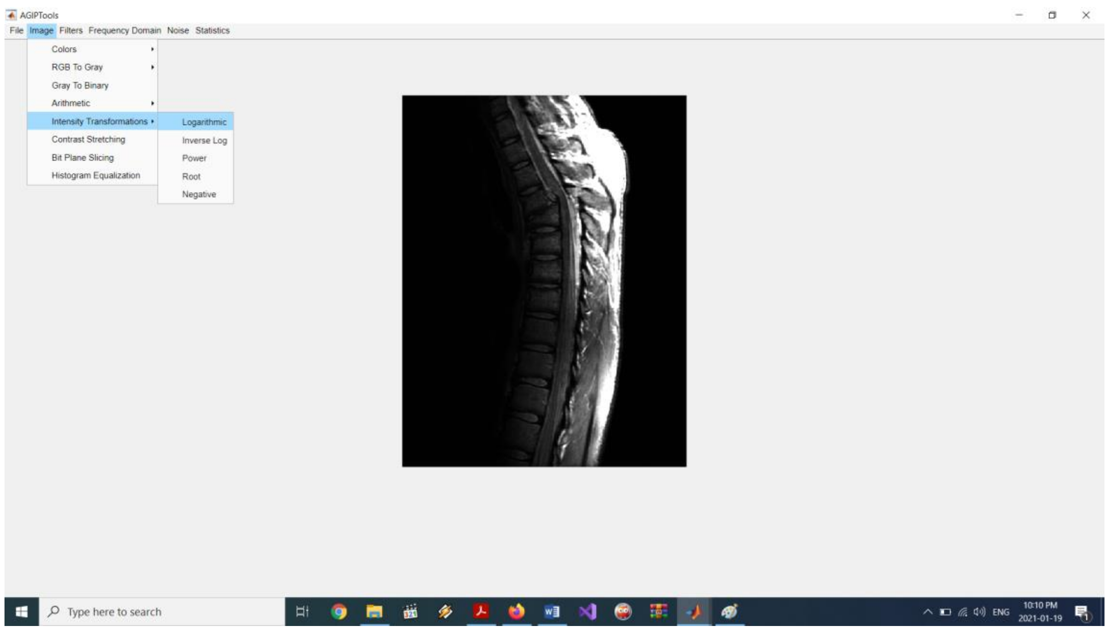


Divide 3

Image → Intensity Transformations

- **Log** $s = c*\log_2(1 + r)$
- **Inverse Log** $s = 2^{r/c} - 1$
- **Power** $s = c*r^p$
- **Root** $s = c*r^{1/p}$
- **Negative** $s = (L - 1) - r = 255 - r$





Original

Log

Original

Inv Log

Original

Root 2

Root 3

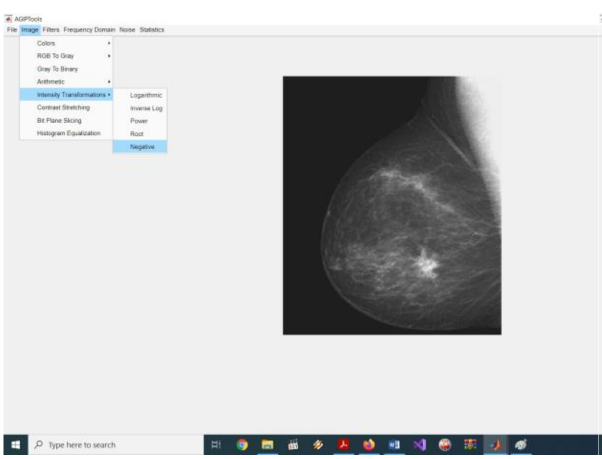
Original

Power 2

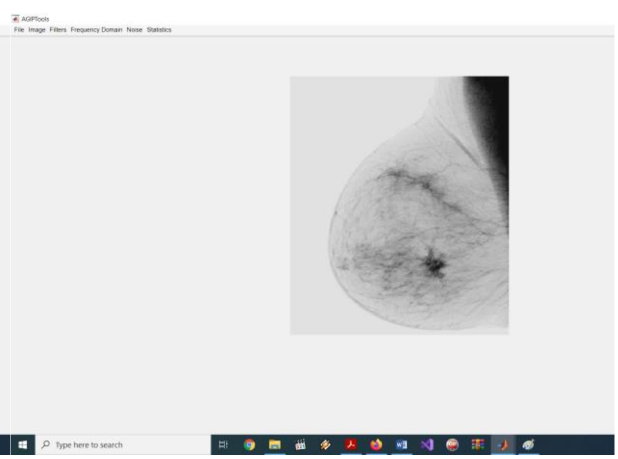
Power 3

Original

Negative



Original



Negative

Original

Log

Original

Inv Log

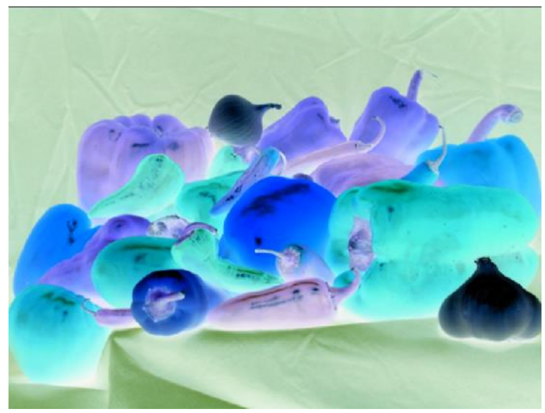
Original

Root 2

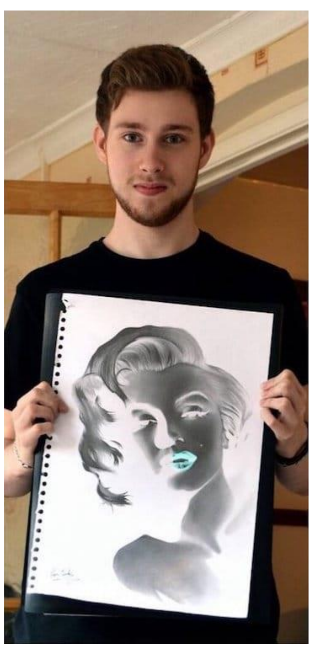
Original

Power 2

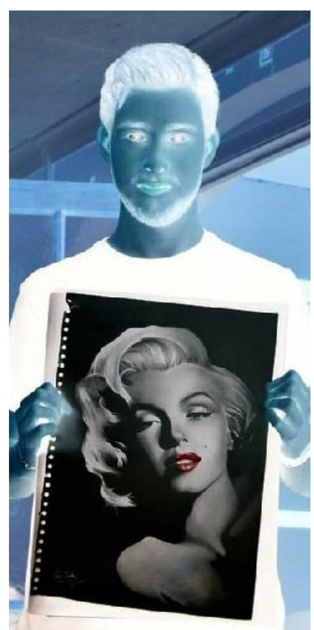
Original



Negative

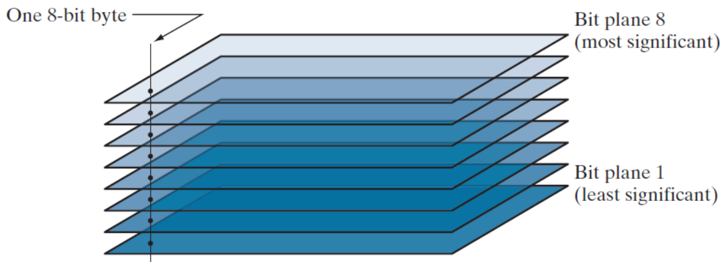


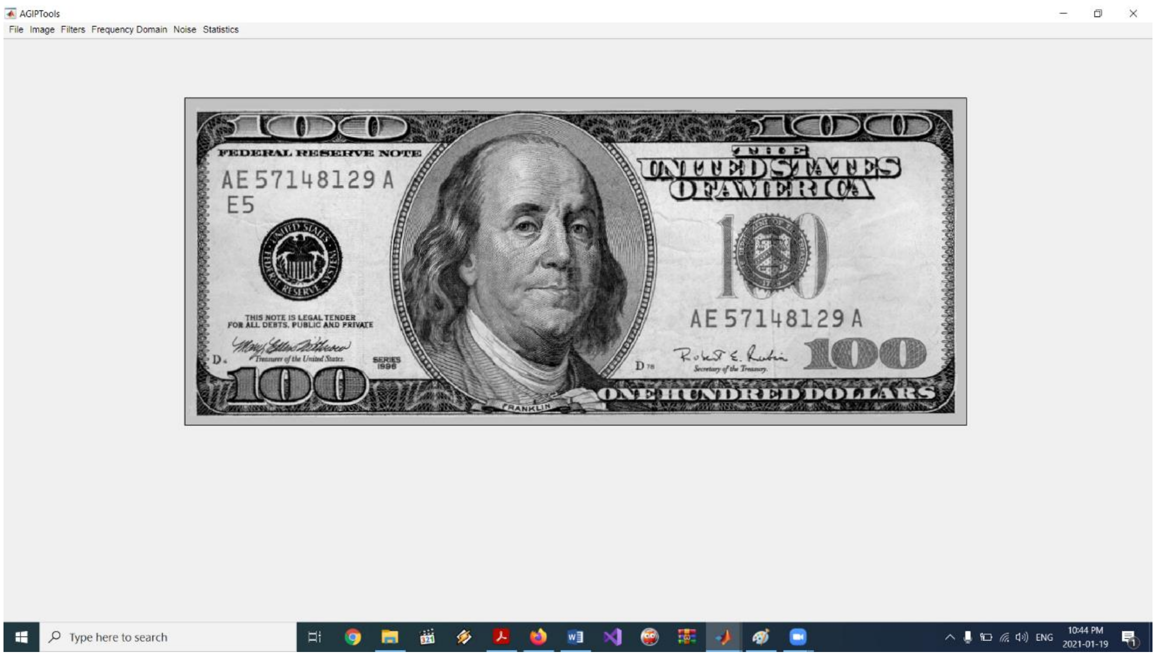
Original

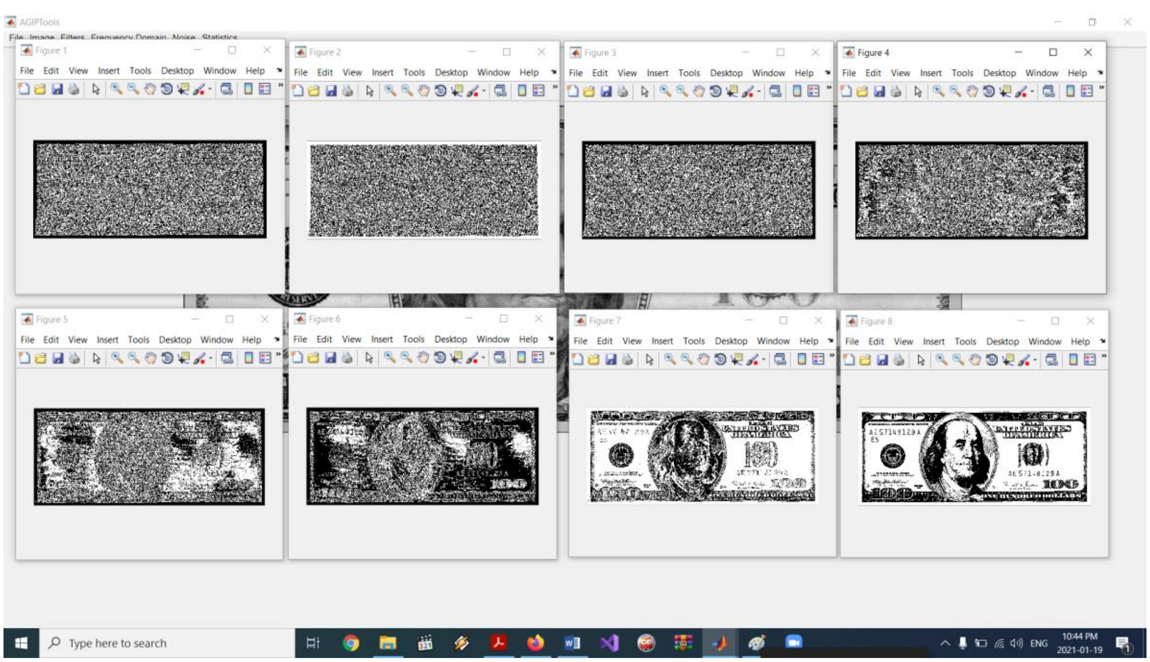


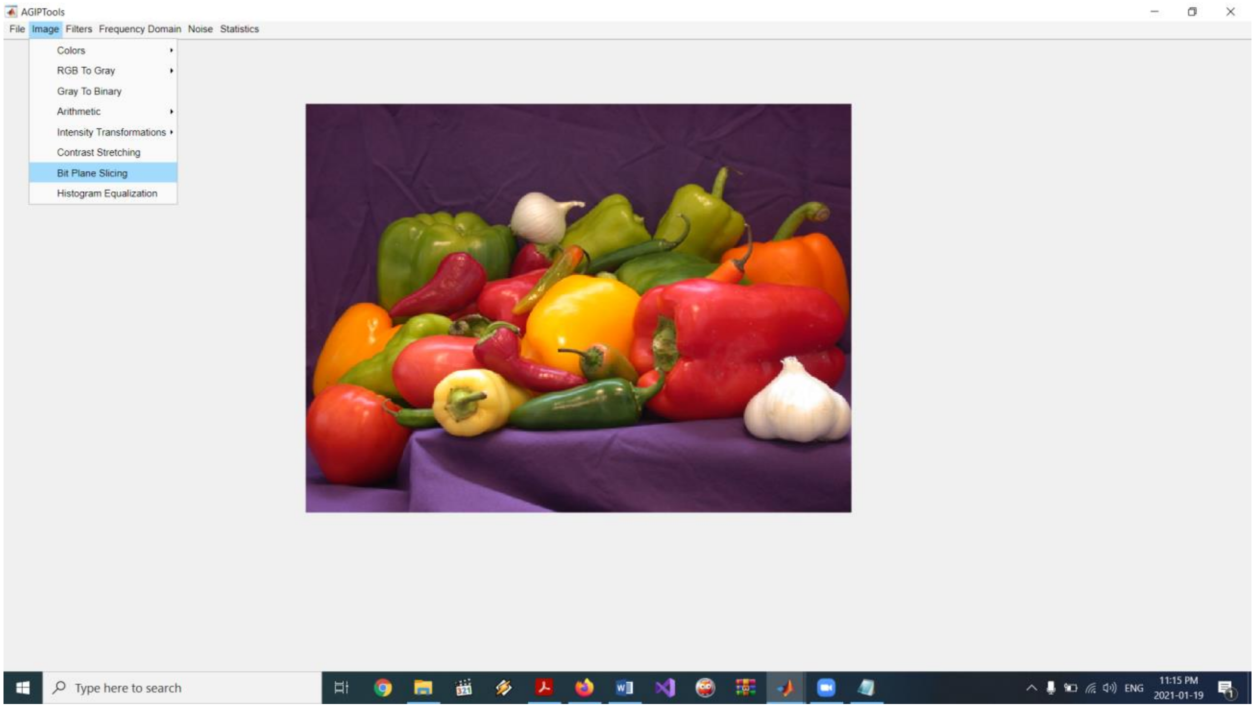
Negative

Image → Intensity Transformations → Bit Plane Slicing

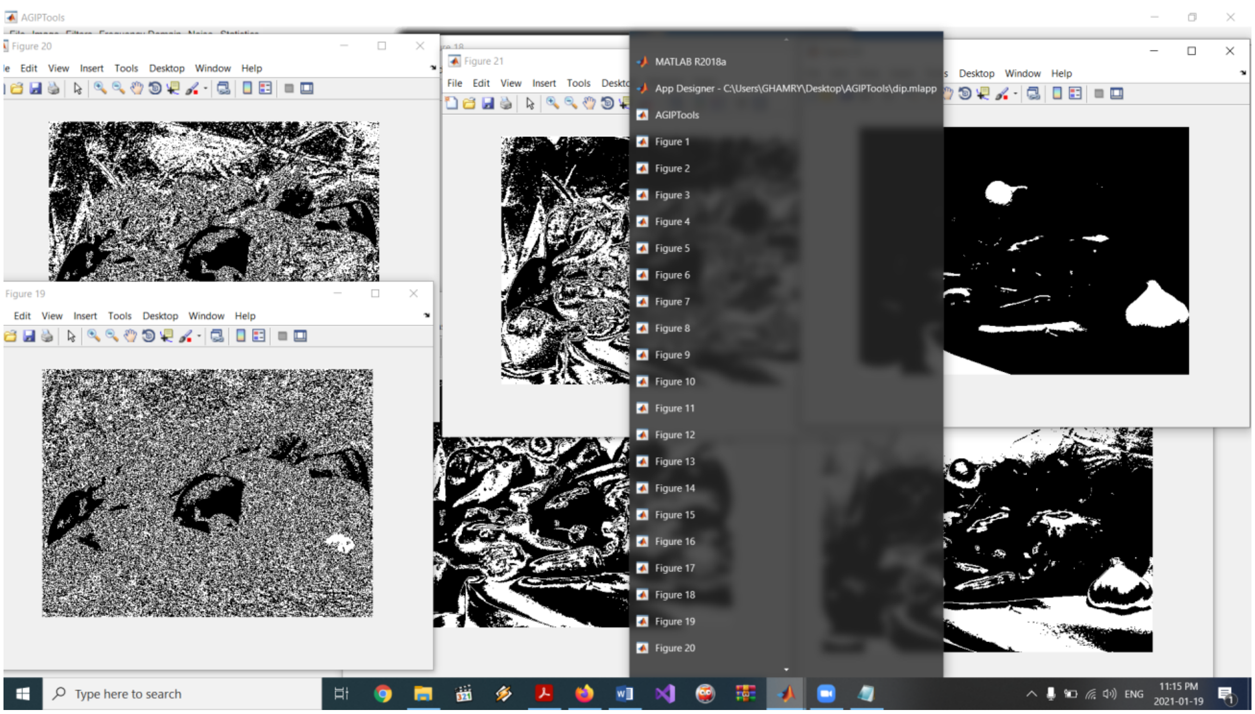






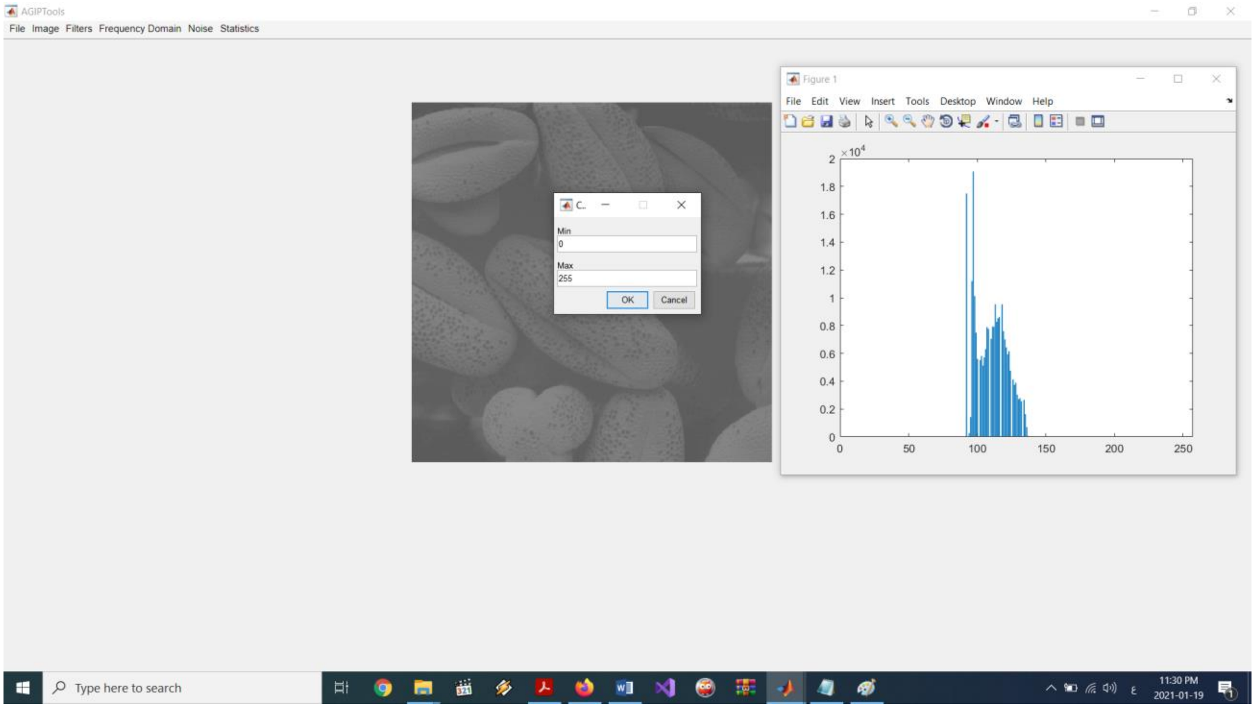


AGIPTools
File Image Filters Frequency Domain Noise Statistics
Colors
RGB To Gray
Gray To Binary
Arithmetic
Intensity Transformations
Contrast Stretching
Bit Plane Slicing
Histogram Equalization
Type here to search
ENG
11:15 PM
2021-01-19



AGIPTools
Figure 20
Figure 19
Figure 21
MATLAB R2018a
App Designer - C:\Users\GHAMRY\Desktop\AGIPTools\dip.mlapp
AGIPTools
Figure 1
Figure 2
Figure 3
Figure 4
Figure 5
Figure 6
Figure 7
Figure 8
Figure 9
Figure 10
Figure 11
Figure 12
Figure 13
Figure 14
Figure 15
Figure 16
Figure 17
Figure 18
Figure 19
Figure 20
Type here to search
ENG
11:15 PM
2021-01-19

Image → Contrast Stretching



Low contrast image

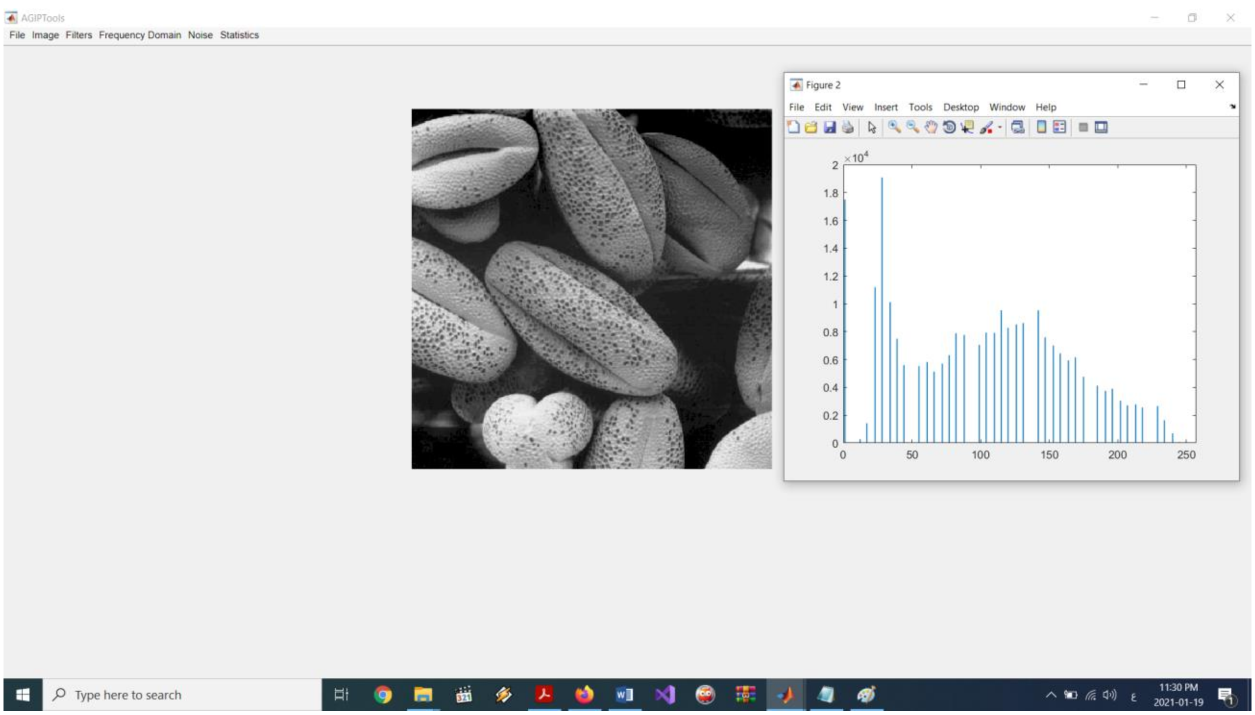
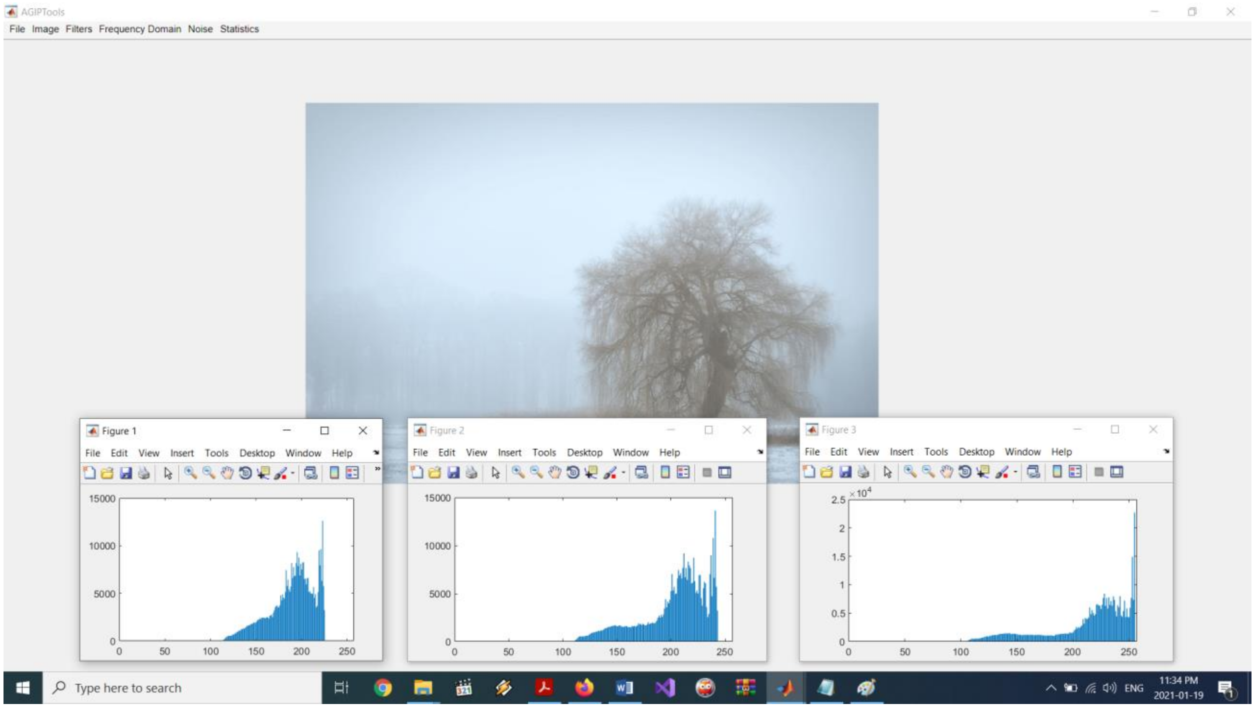


Image after contrast stretching



Light image

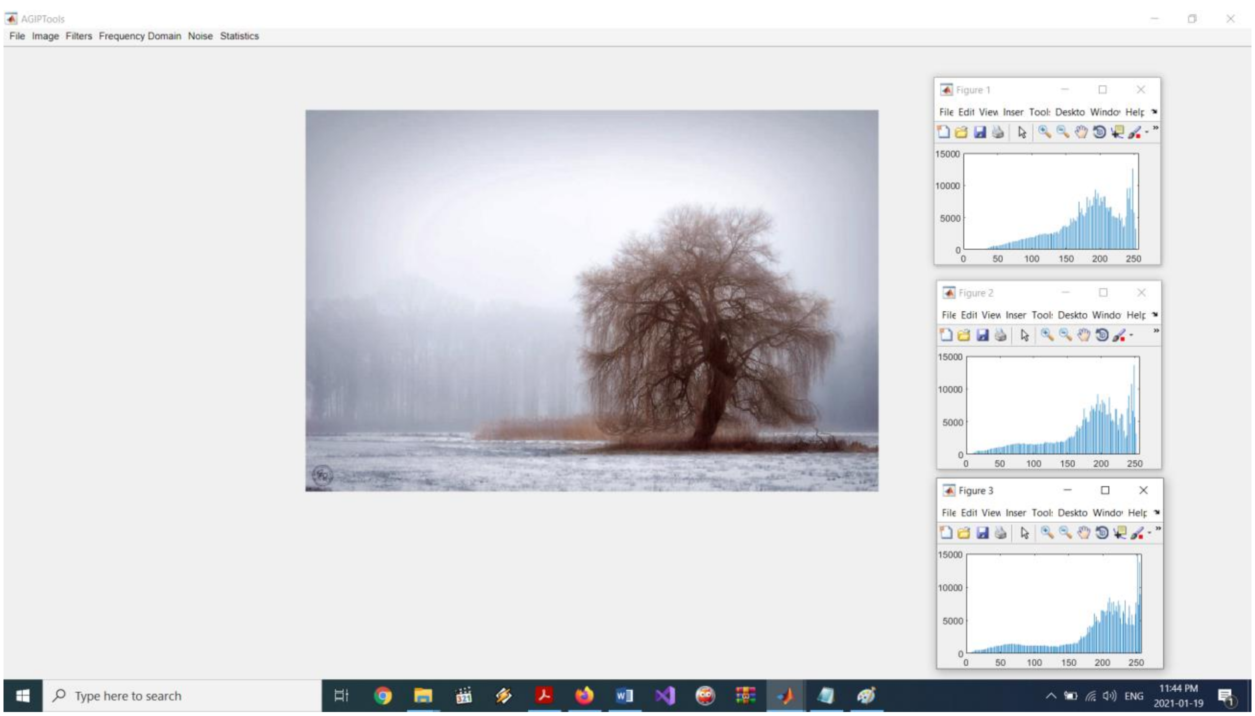
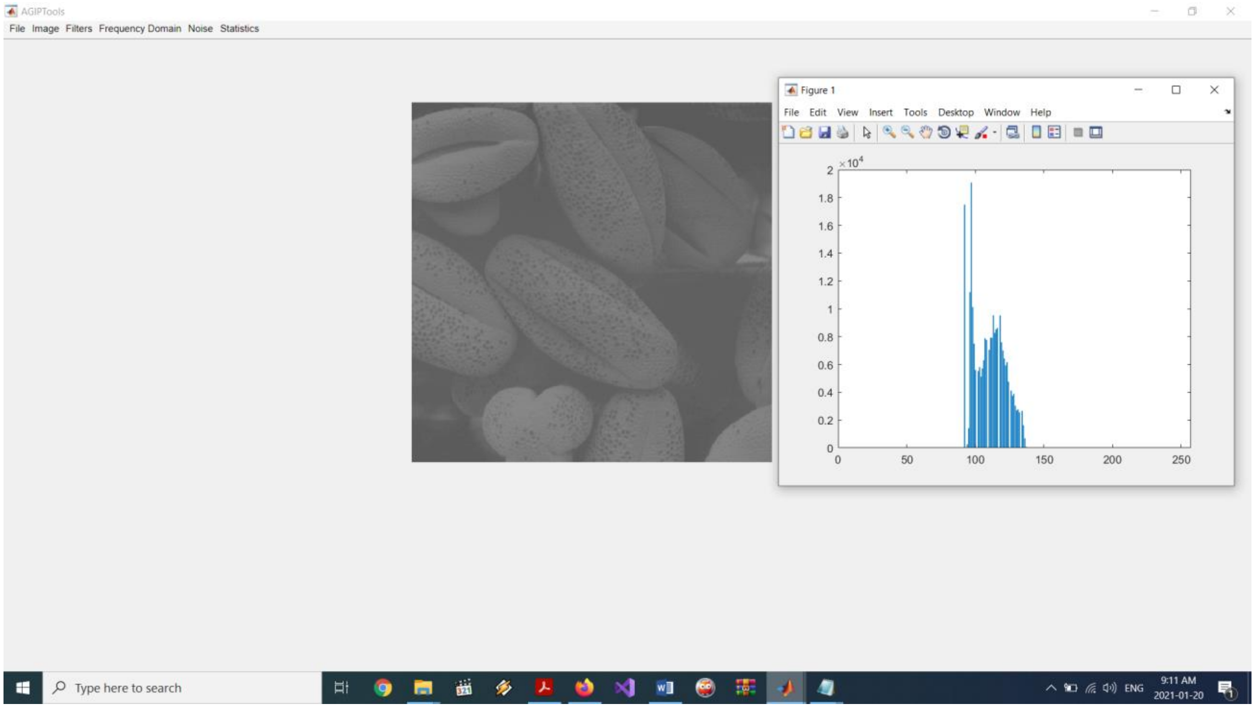


Image after contrast stretching

Image → Histogram Equalization



Low contrast image

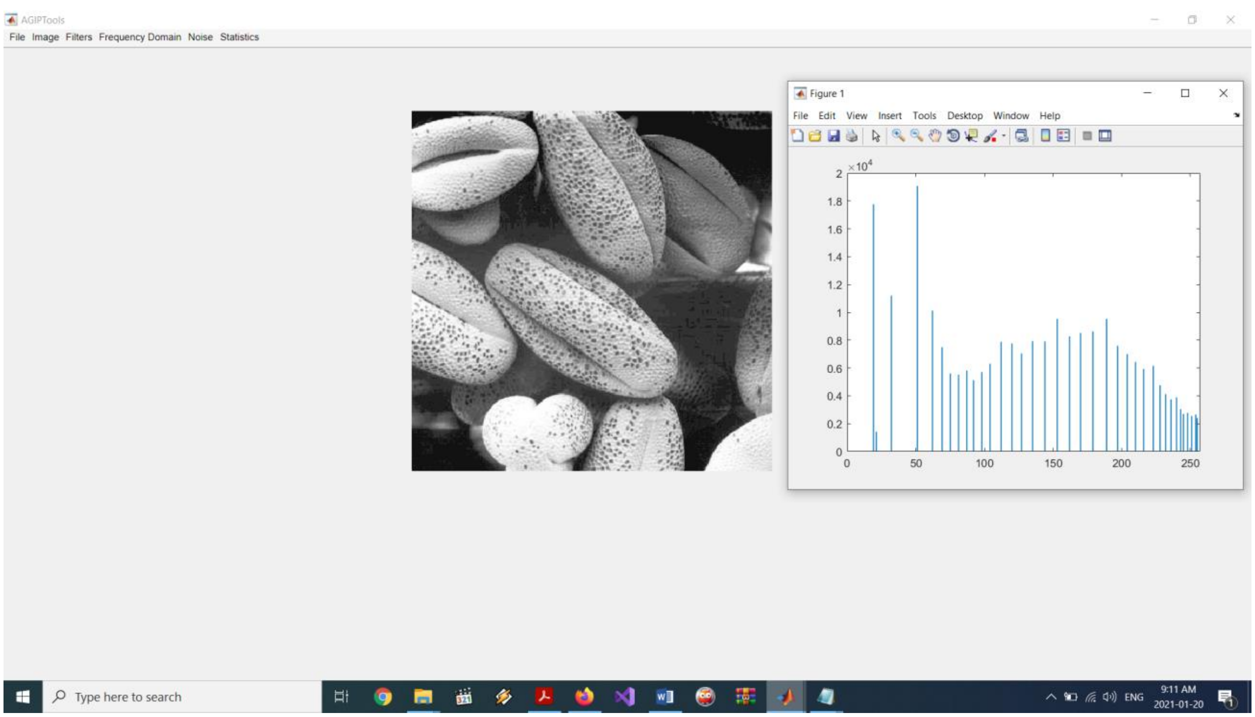
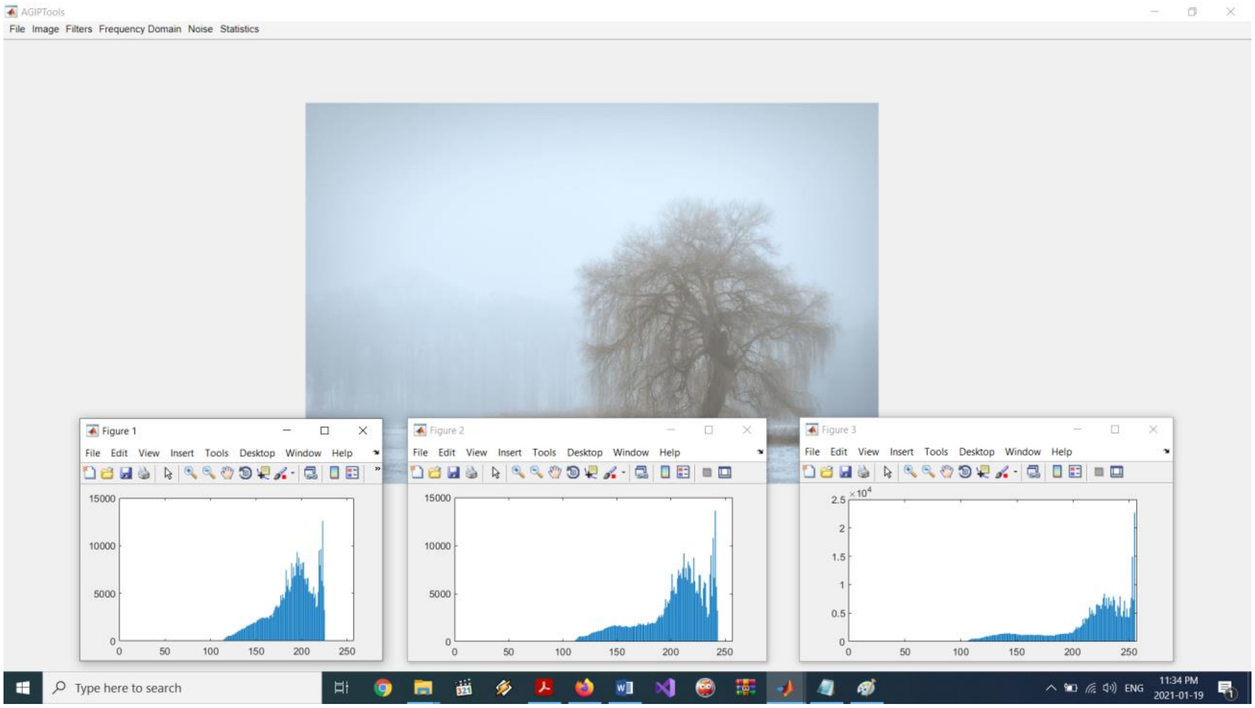


Image after histogram equalization



Light image

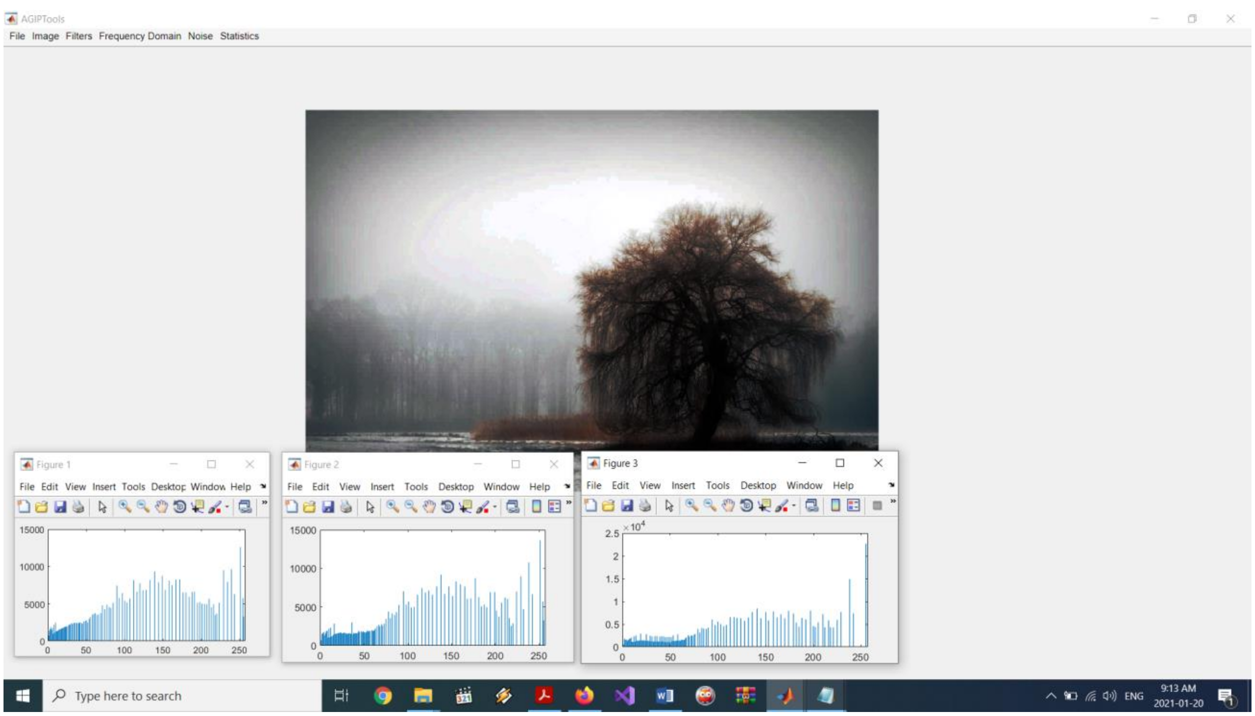


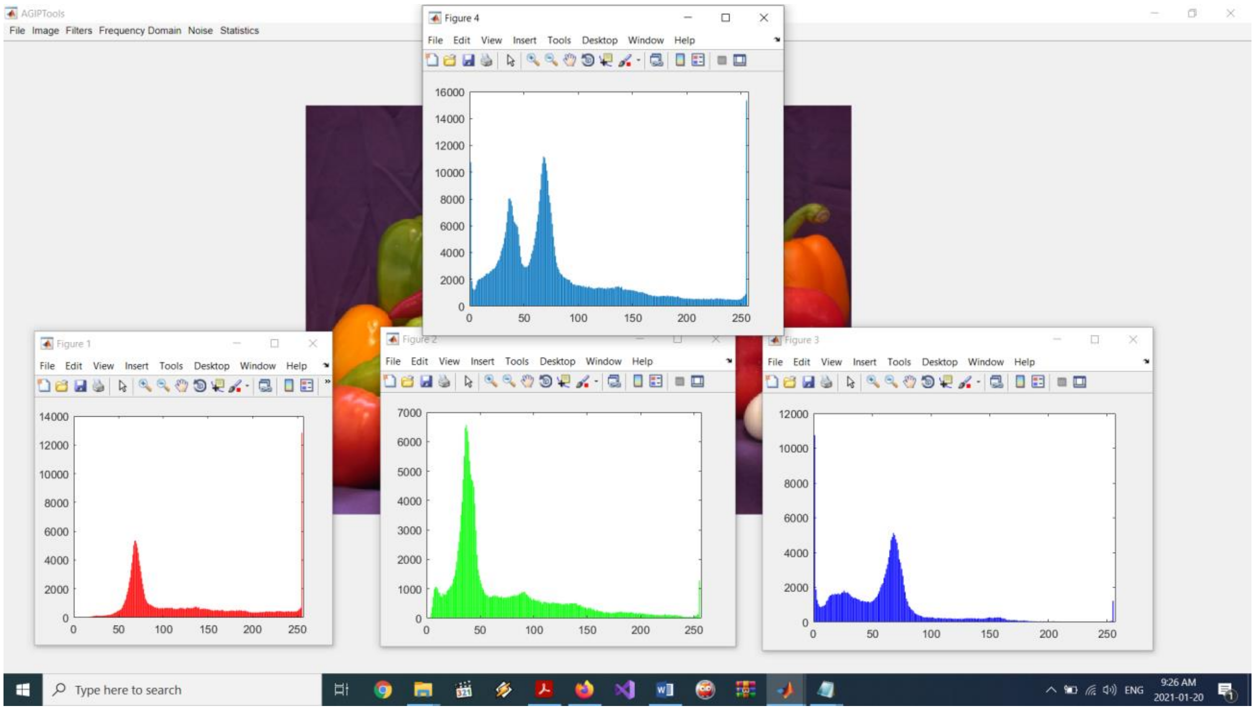
Image after histogram equalization

Statistics → Histogram

If the image is RGB, it displays 3 channels.

Statistics → One Histogram

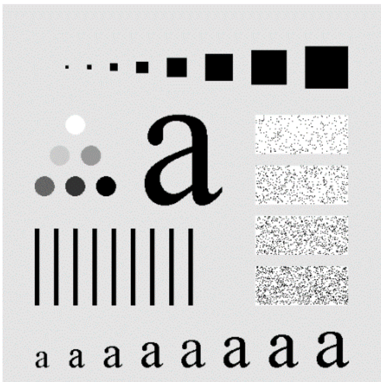
If the image is RGB, it combines the 3 channels.



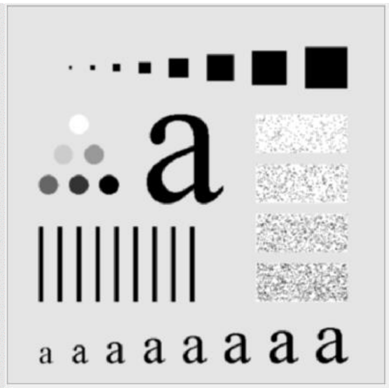
Filters → Linear Spatial Filtering → Mean

- **Arithmetic Mean**

$$\hat{f}(x,y)=\frac{1}{mn}\sum_{(r,c)\in S_{xy}} g(r,c)$$

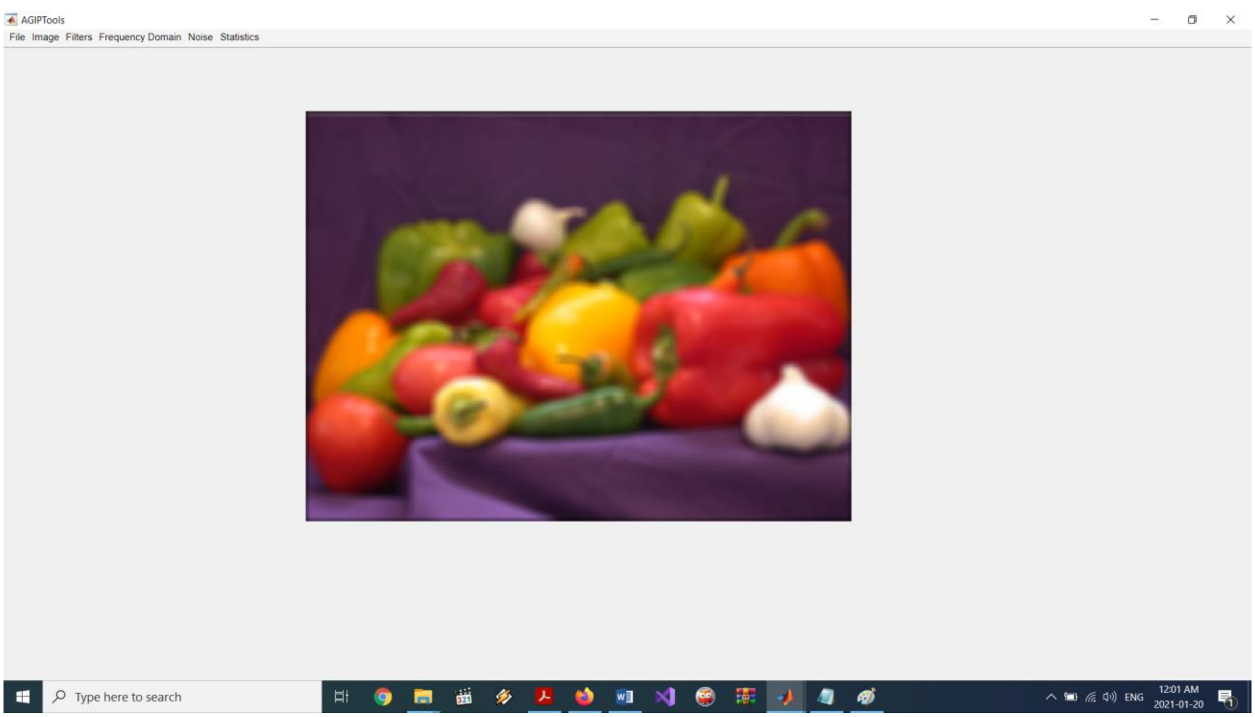


Original



3*3 Arithmetic Mean

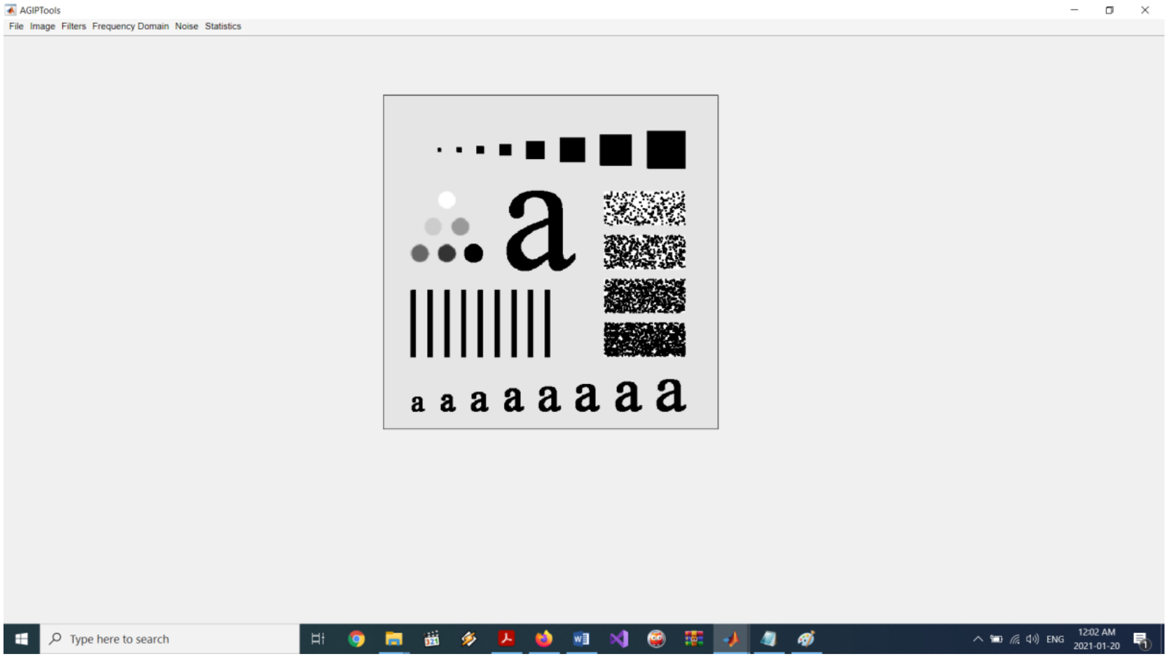
7*7 Arithmetic Mean



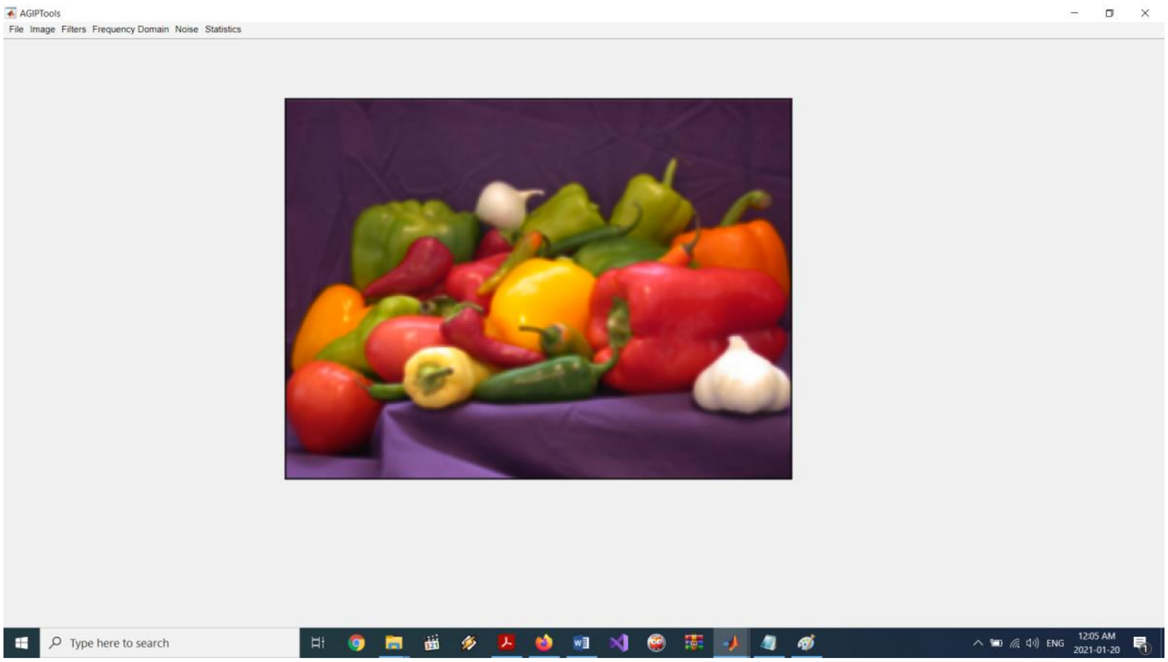
9*9 Arithmetic Mean

- **Geometric Mean**

$$\hat{f}(x,y) = \left[\prod_{(r,c)\in S_{xy}} g(r,c) \right]^{\frac{1}{mn}}$$



3*3 Geometric Mean

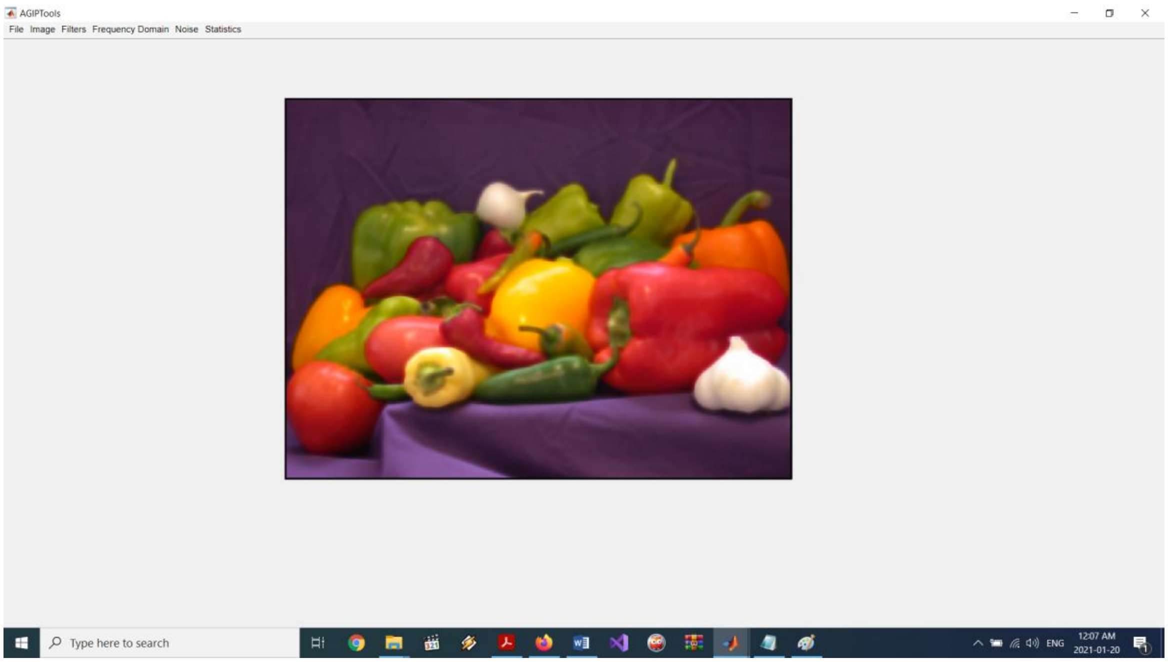


5*5 Geometric Mean

- **Harmonic Mean**

$$\hat{f}(x,y) = \frac{mn}{\sum_{(r,c)\in S_{xy}} \frac{1}{g(r,c)}}$$

5*5 Harmonic Mean



5*5 Harmonic Mean

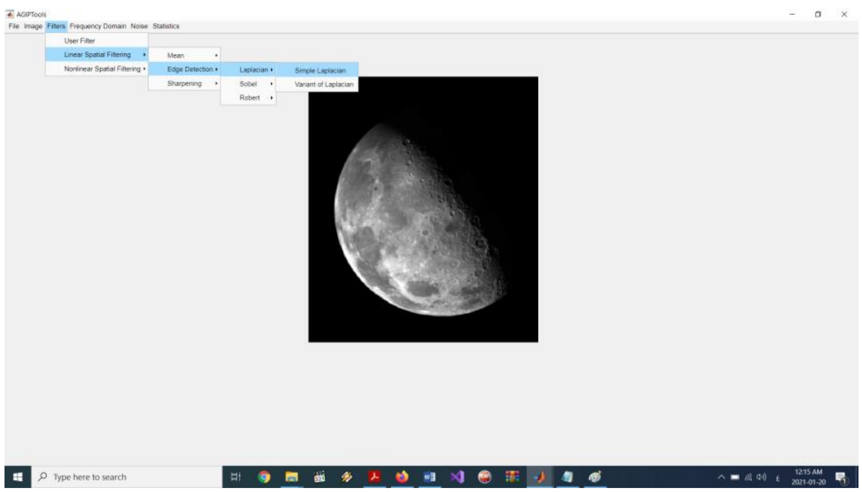
Filters → Linear Spatial Filtering → Edge Detection → Laplacian

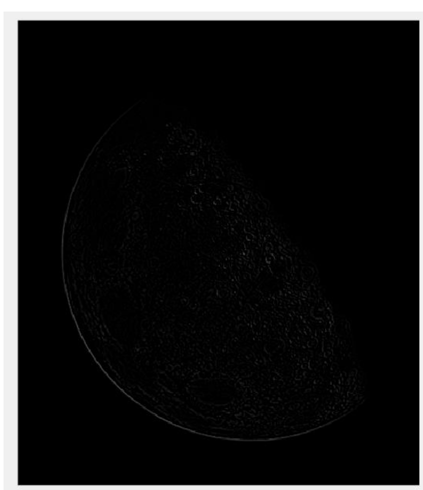
0	1	0
1	−4	1
0	1	0

Simple Laplacian

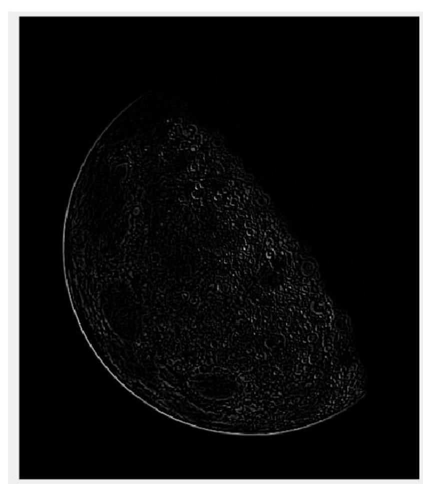
1	1	1
1	−8	1
1	1	1

Variant of Laplacian

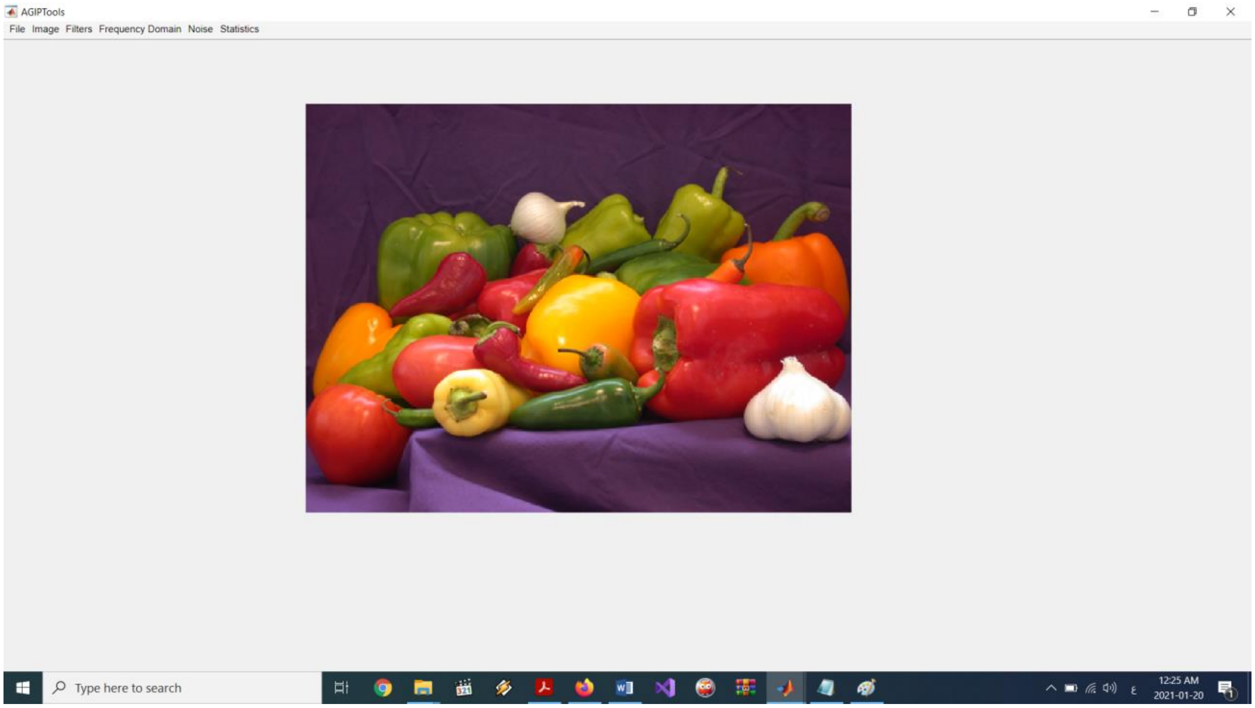


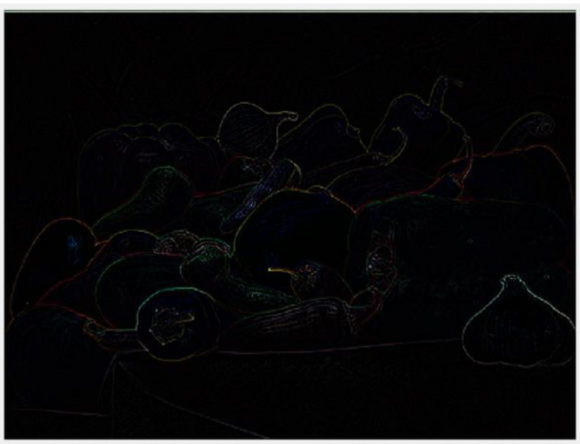


Simple Laplacian

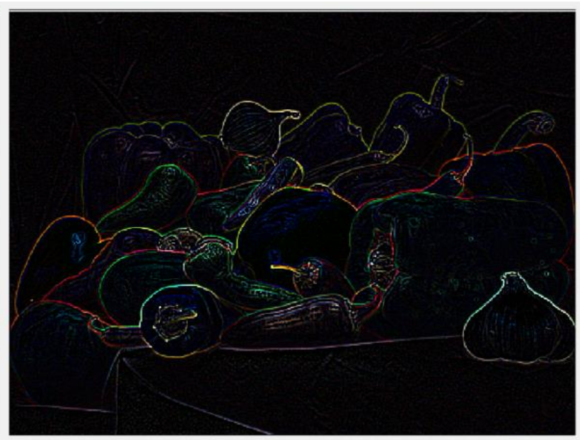


Variant of Laplacian





Simple Laplacian



Variant of Laplacian

Filters → Linear Spatial Filtering → Edge Detection → Sobel

-1	-2	-1
0	0	0
1	2	1

H

-1	0	1
-2	0	2
-1	0	1

V

0	-1	-2
1	0	-1
2	1	0

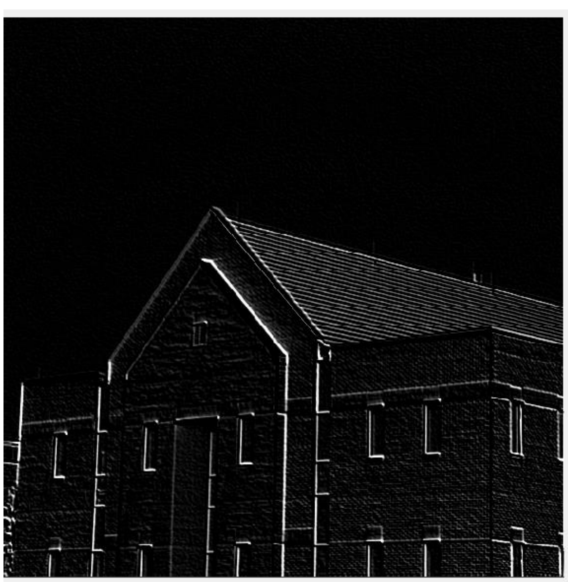
LD

2	1	0
1	0	-1
0	-1	-2

RD

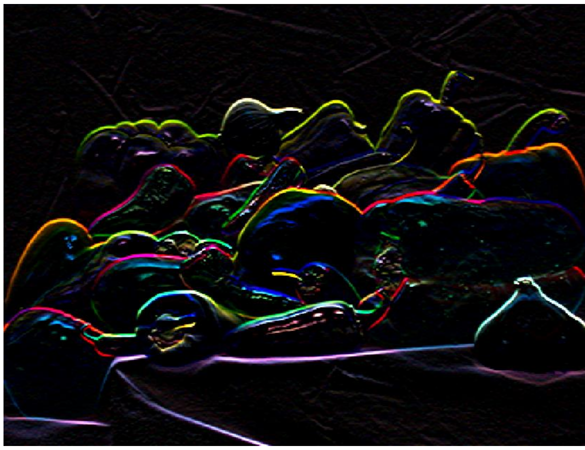
H

V



LD

RD



H

V

LD

RD

Filters → Linear Spatial Filtering → Edge Detection → Robert

0	1	0
0	0	0
0	-1	0

H

0	0	0
1	0	-1
0	0	0

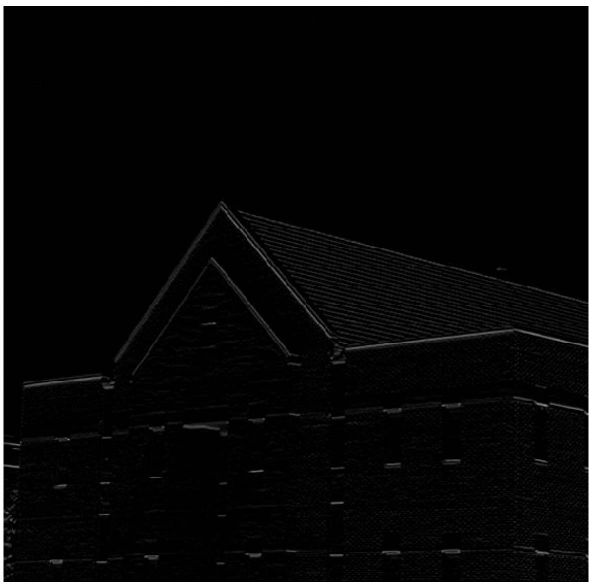
V

0	0	1
0	0	0
-1	0	0

LD

1	0	0
0	0	0
0	0	-1

RD

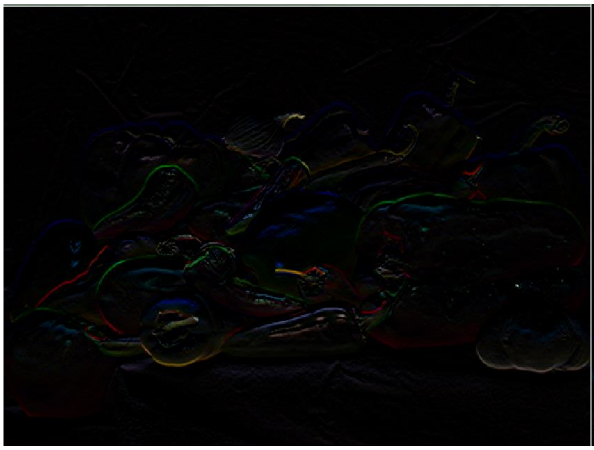


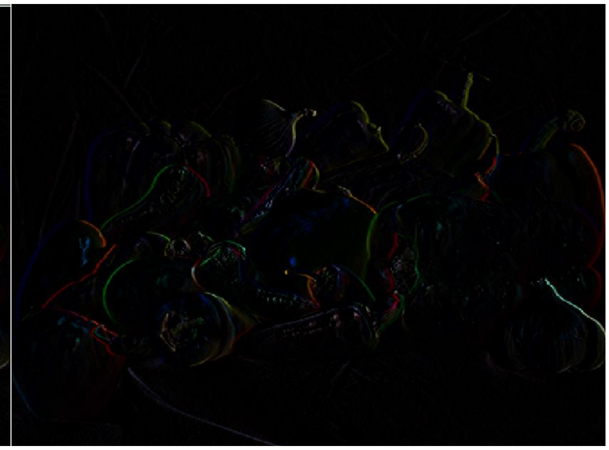
H

V

LD

RD





H

V

LD

RD

Filters → Linear Spatial Filtering → Sharpening

0	-1	0
-1	5	-1
0	-1	0

Laplacian

0	1	0
0	1	0
0	-1	0

H

0	0	0
1	1	-1
0	0	0

V

1	0	0
0	1	0
0	0	-1

LD

0	0	1
0	1	0
-1	0	0

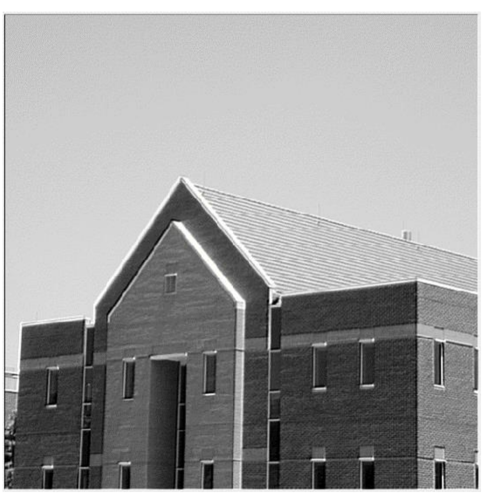
RD

Oiginal

Laplacian

H

V



LD

RD

Oiginal

Laplacian

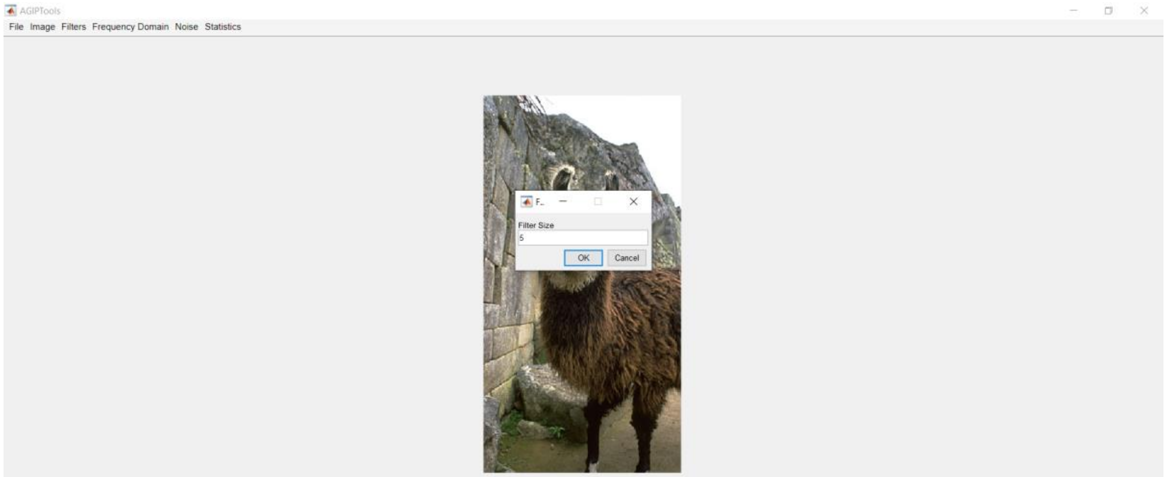
H

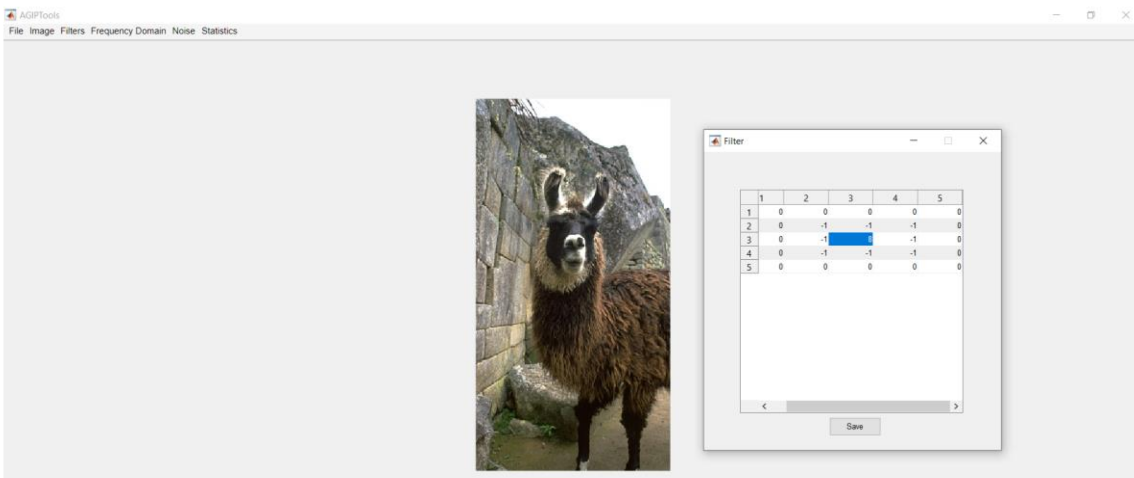
V

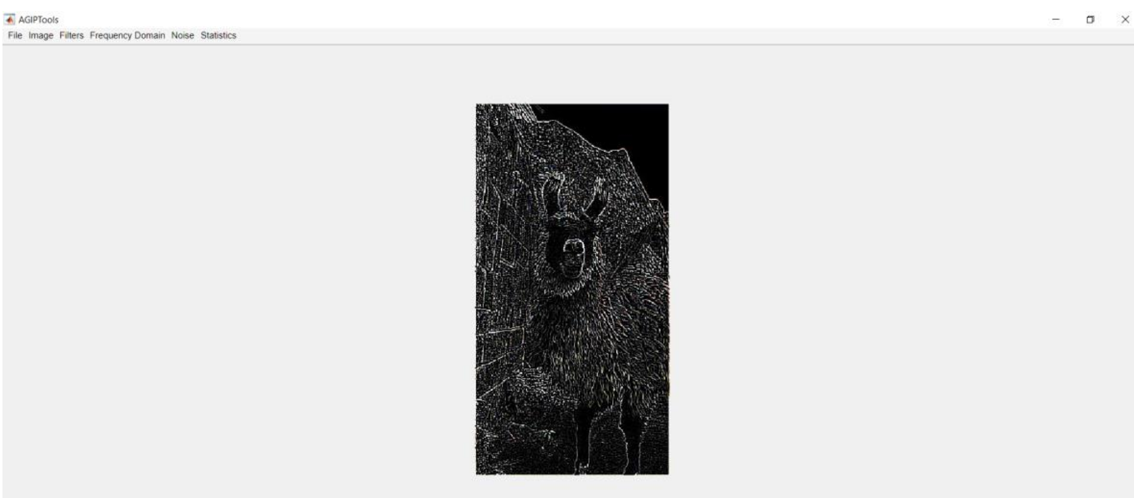
LD

RD

Filters → Linear Spatial Filtering → User Filter







Filters → Nonlinear Spatial Filtering → Min

This filter is useful for finding the darkest points in an image.

$$\hat{f}(x,y) = \min_{(r,c)\in S_{xy}} \{g(r,c)\}$$



Original 3*3 Min Filter 5*5 Min Filter


Original 3*3 Min Filter

Filters → Nonlinear Spatial Filtering → Max

This filter is useful for finding the brightest points in an image.

$$\hat{f}(x,y) = \max_{(r,c)\in S_{xy}} \{g(r,c)\}$$



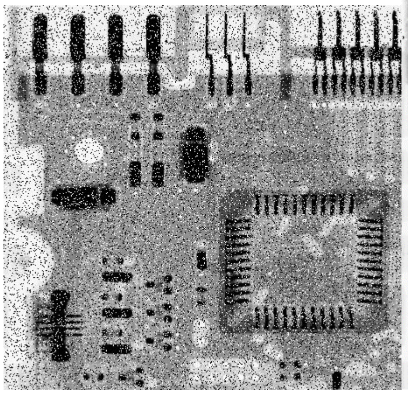
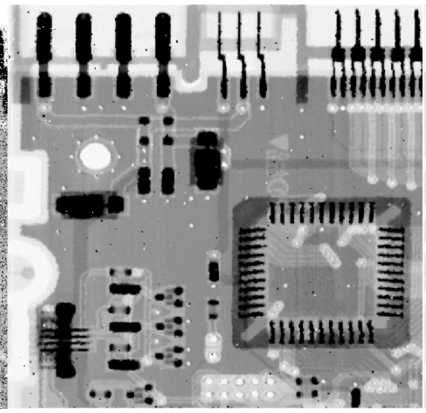
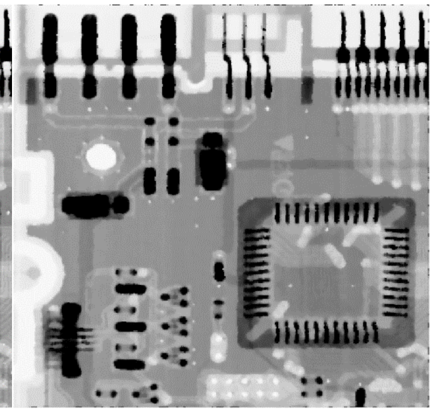
Original 3*3 Max Filter 5*5 Max Filter


Original 3*3 Max Filter

Filters → Nonlinear Spatial Filtering → Median

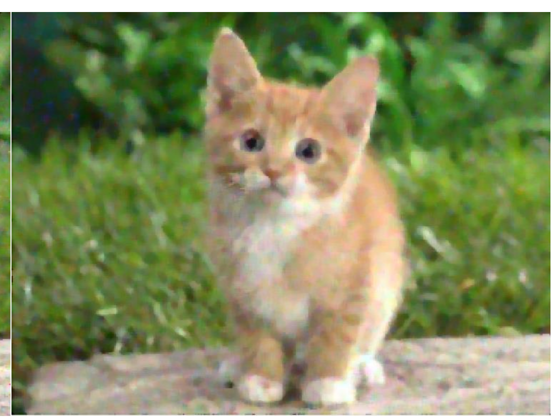
This filter is useful for removing salt and pepper noise in an image.

$$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}}\{g(r,c)\}$$

Original 3*3 Median Filter 5*5 Median Filter

Original

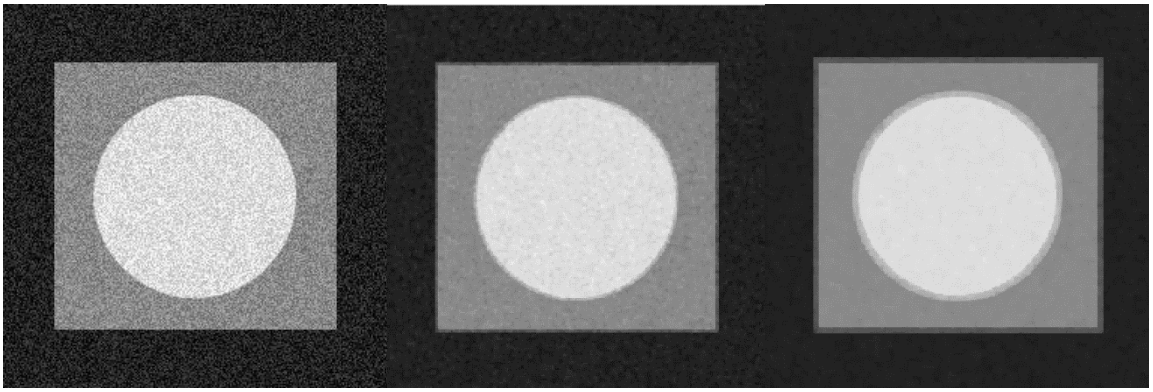



3*3 Median Filter 5*5 Median Filter

Filters → Nonlinear Spatial Filtering → Midpoint

It works best for randomly distributed noise, like Gaussian or uniform noise.

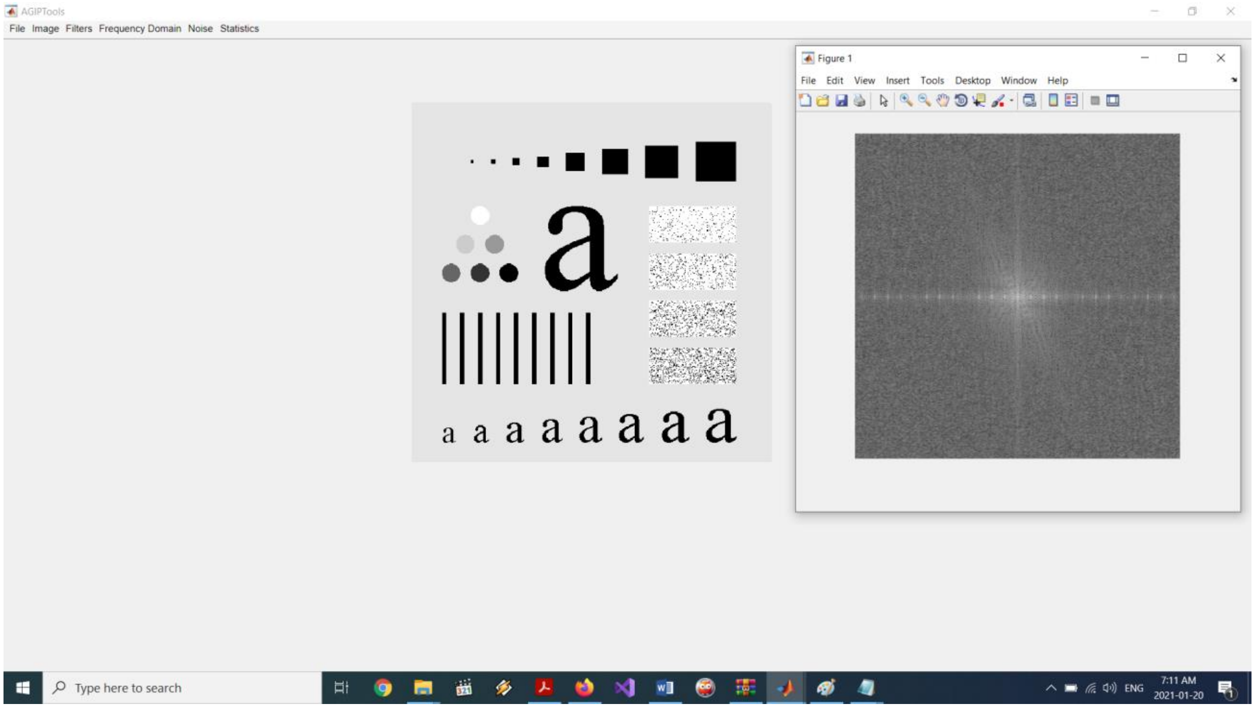
$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(r,c)\in S_{xy}}\{g(r,c)\} + \min_{(r,c)\in S_{xy}}\{g(r,c)\}\right]$$

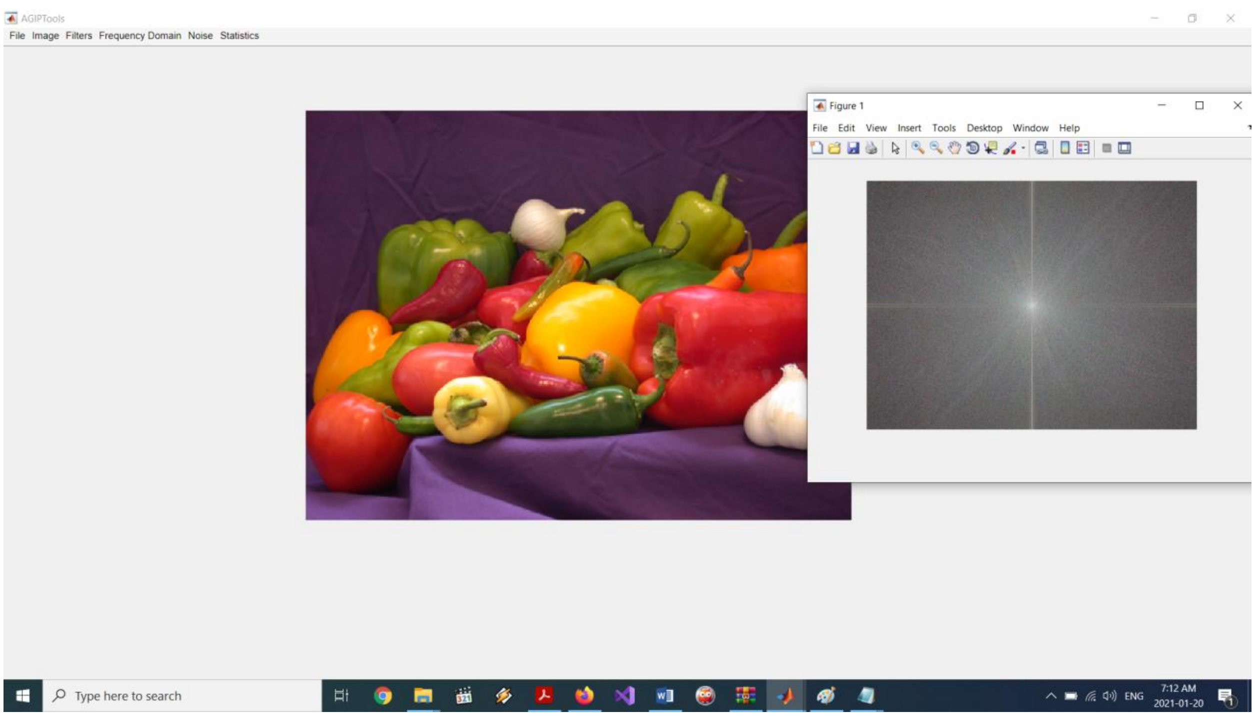


Original 3*3 Midpoint Filter 5*5 Midpoint Filter

Original 3*3 Midpoint Filter 5*5 Midpoint Filter

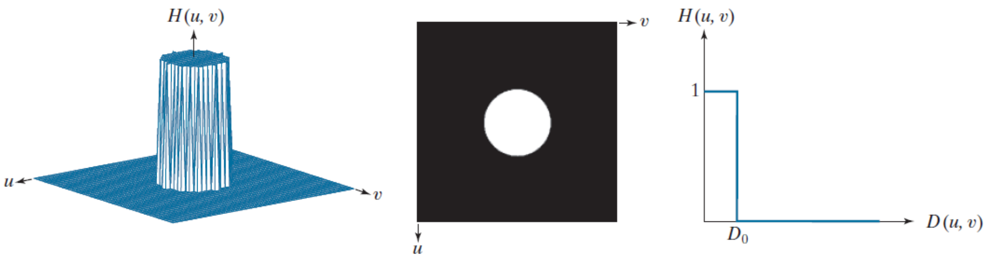
Frequency Domain → Log DFT

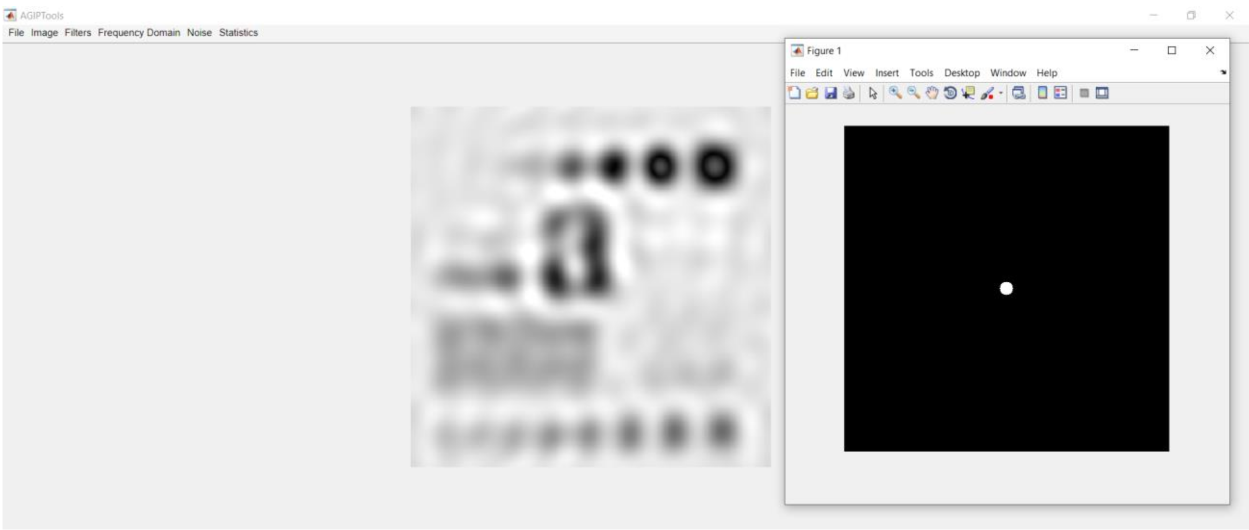




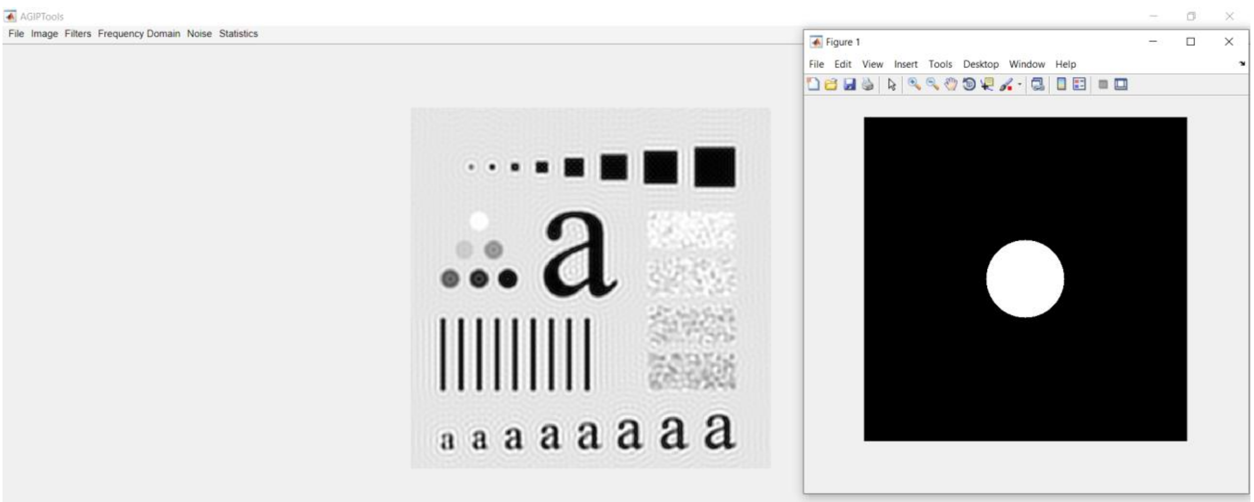
Frequency Domain → Low pass Filters → Ideal

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

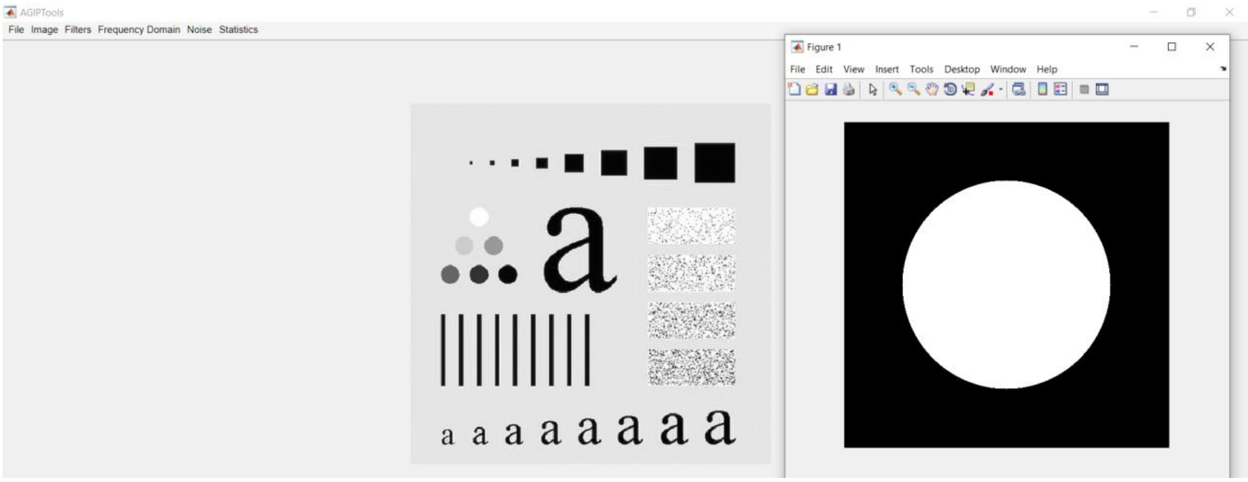




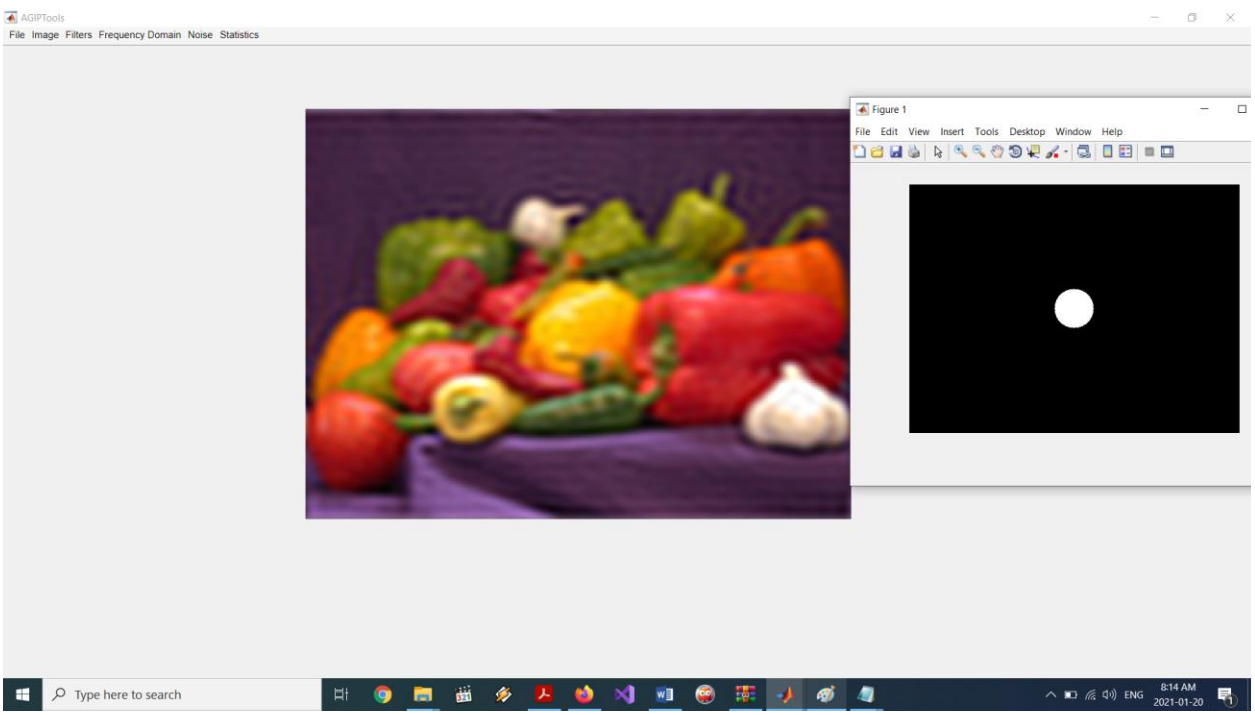
$D_0 = 10$



$D_0 = 60$


AGIPTools
File Image Filters Frequency Domain Noise Statistics
Figure 1
File Edit View Insert Tools Desktop Window Help

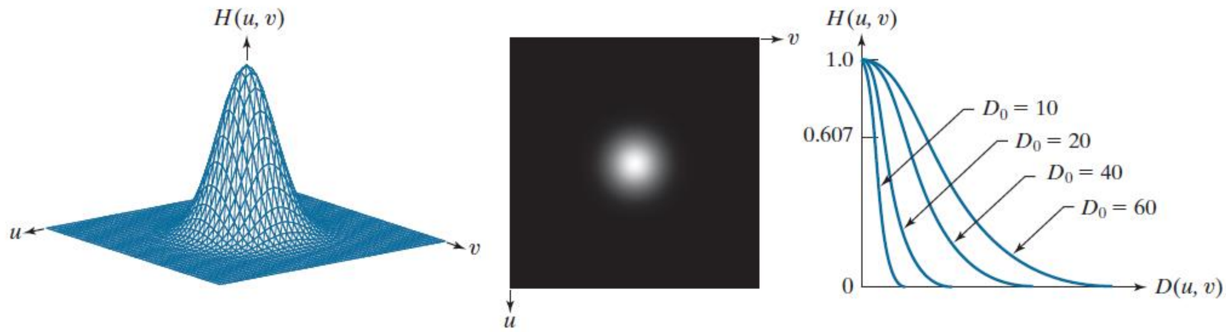
$D_0 = 160$

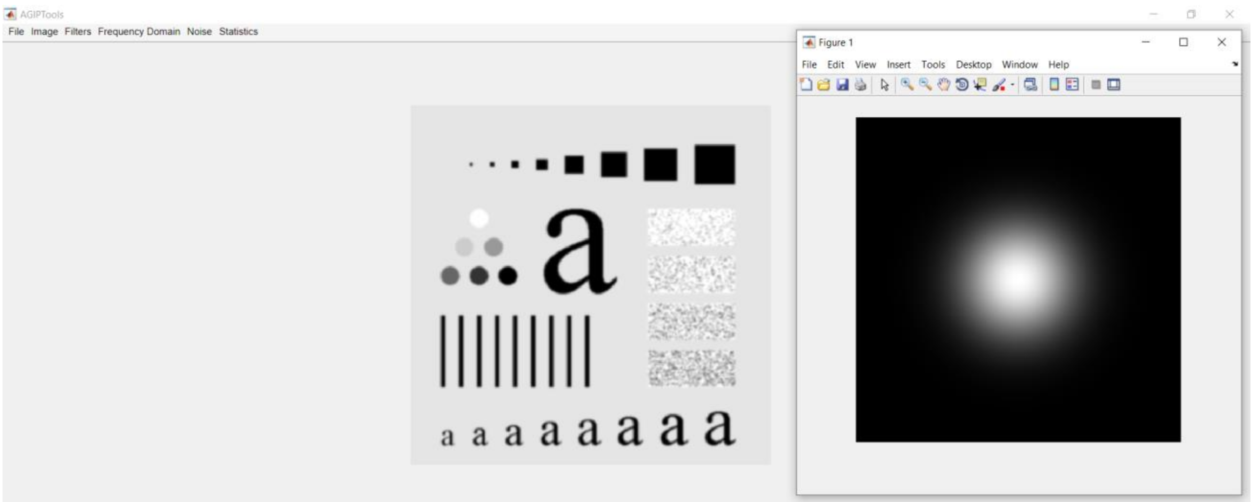

AGIPTools
File Image Filters Frequency Domain Noise Statistics
Figure 1
File Edit View Insert Tools Desktop Window Help
Type here to search
ENG
8:14 AM
2021-01-20

$D_0 = 30$

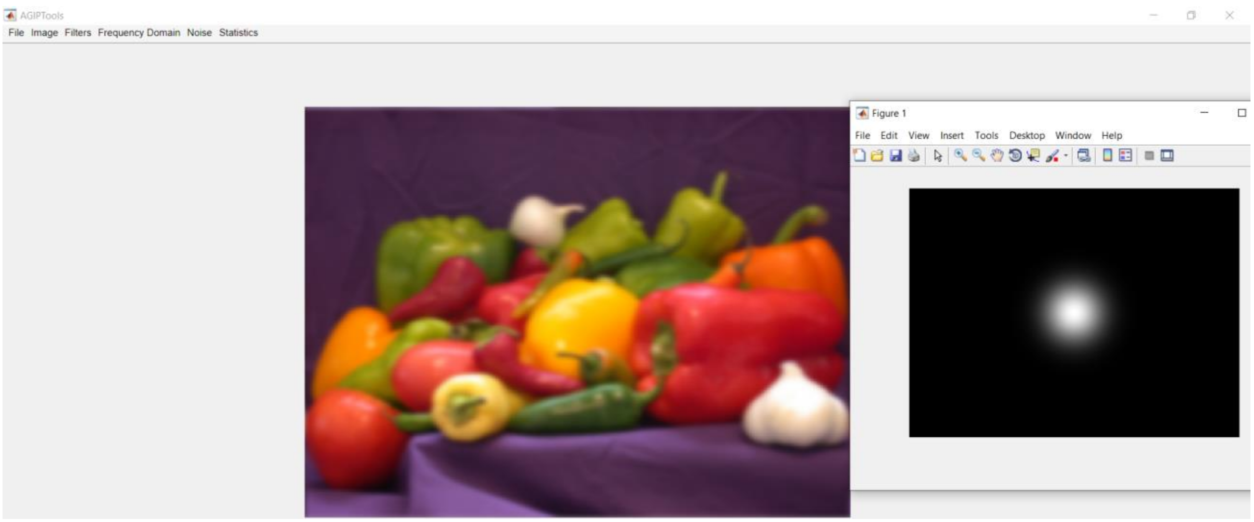
Frequency Domain → Low pass Filters → Gaussian

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$





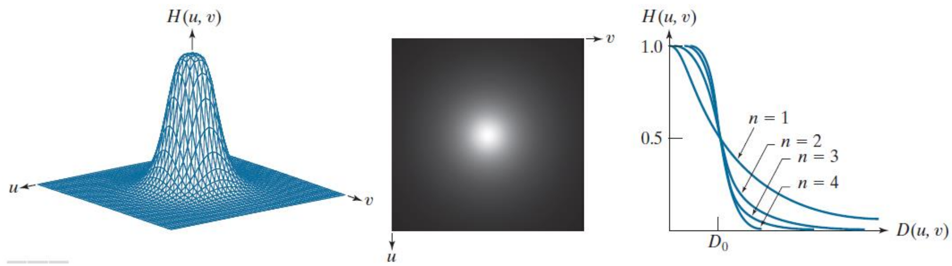
$D_0 = 60$

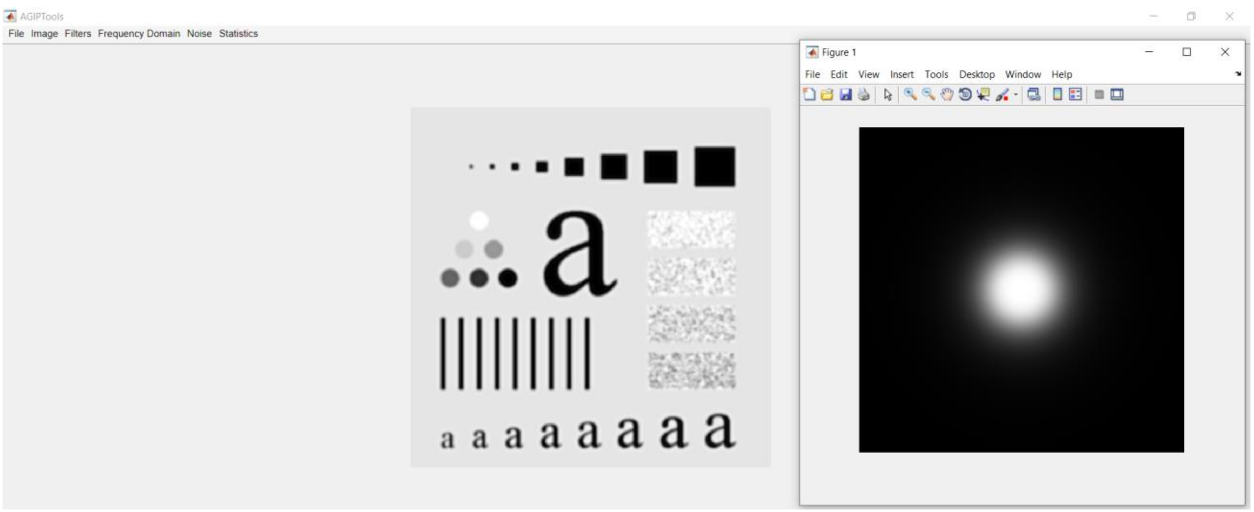


$D_0 = 30$

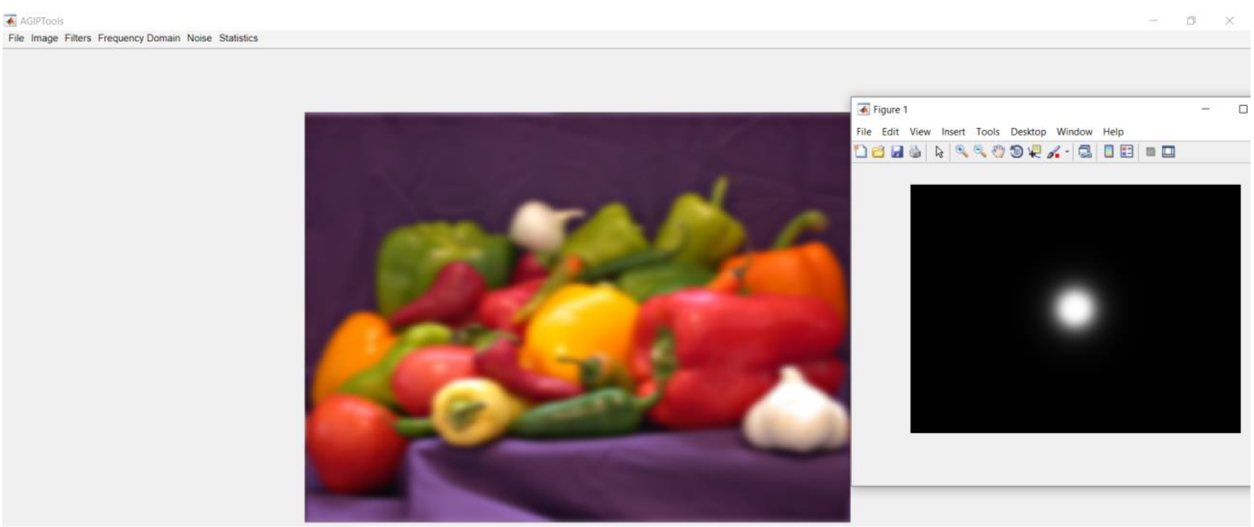
Frequency Domain → Low pass Filters → Butterworth

$$H(u,v) = \frac{1}{1 + \left[D(u,v)/D_0 \right]^{2n}}$$





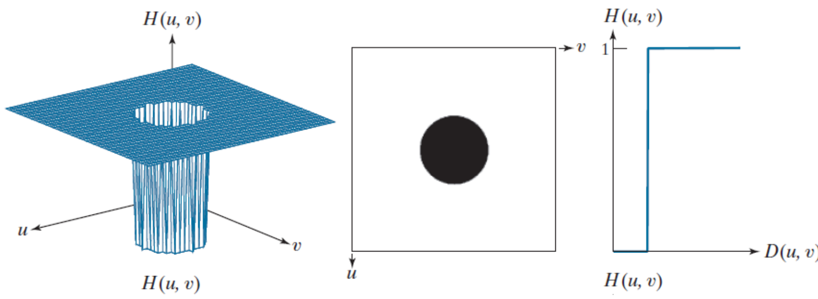
D_0 = 60, n = 2

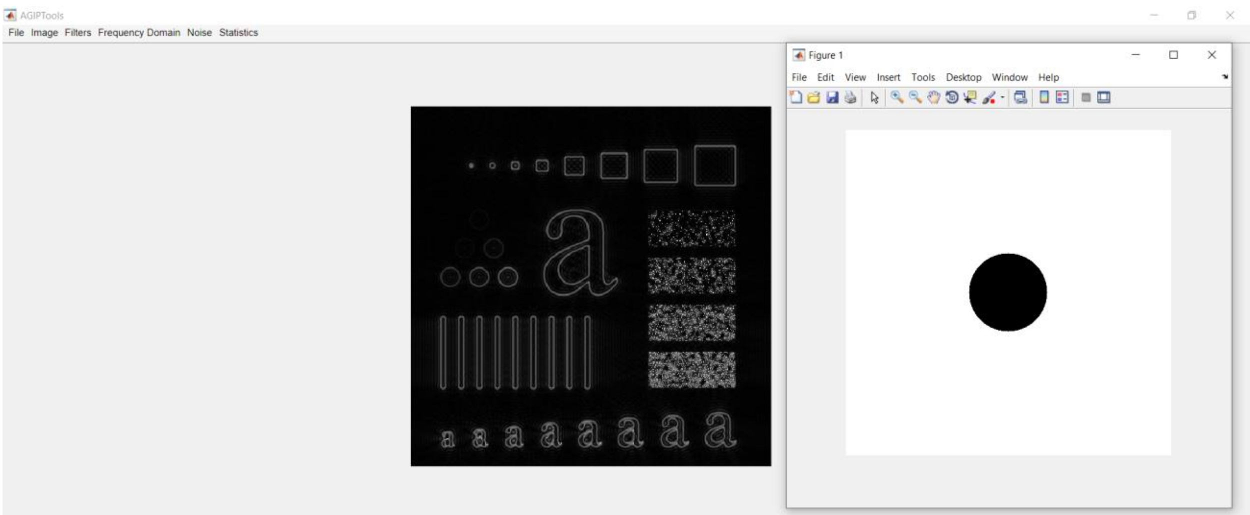


D_0 = 30, n = 2

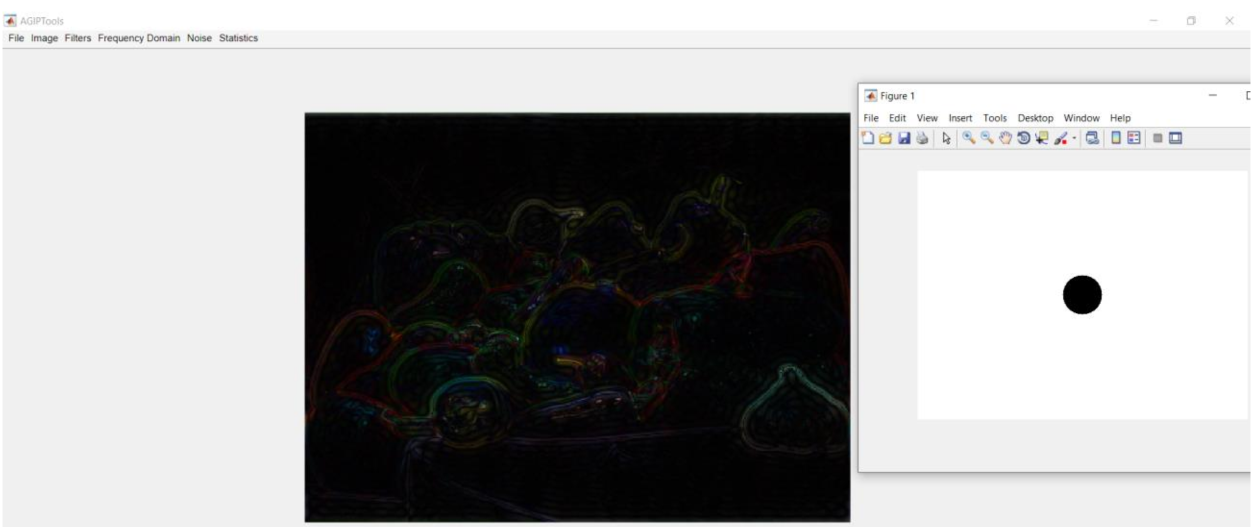
Frequency Domain → High pass Filters → Ideal

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$





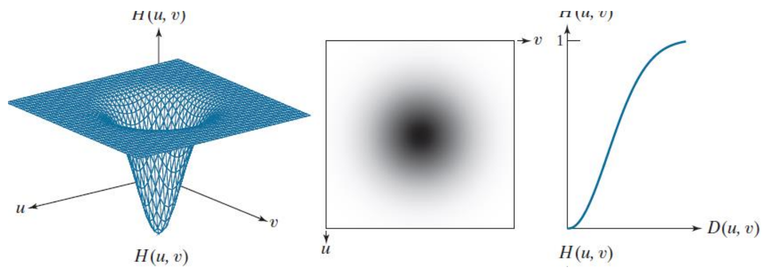
$D_0 = 60$

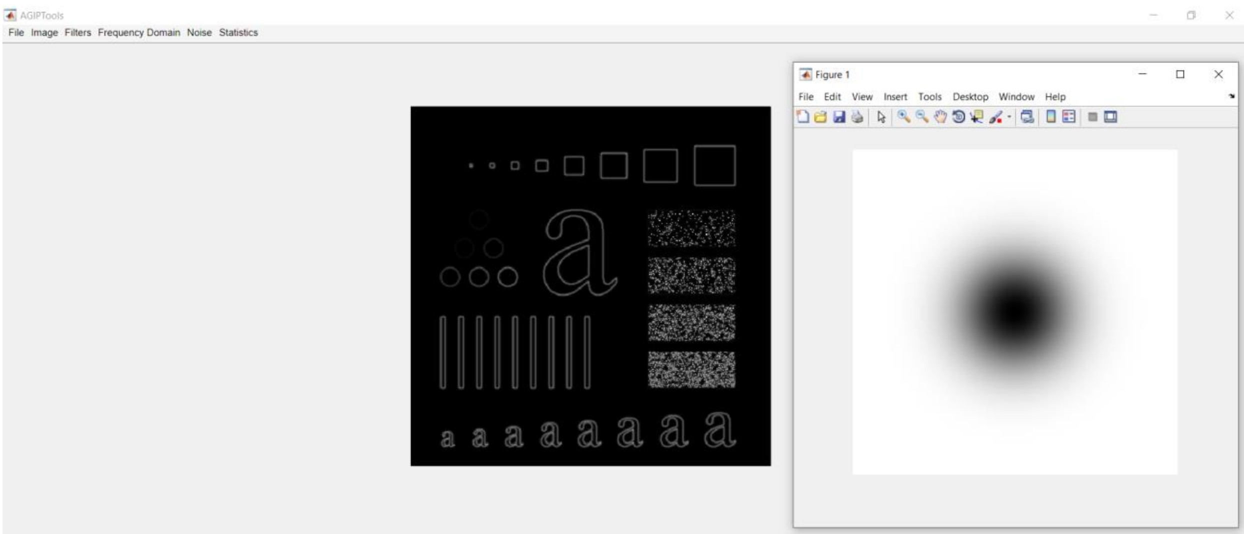


$D_0 = 30$

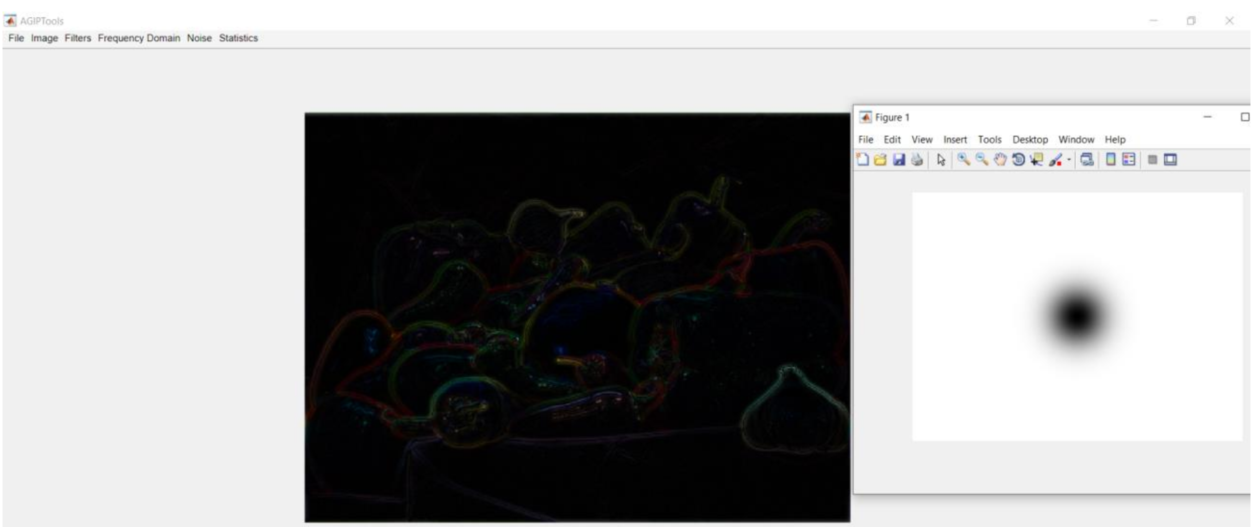
Frequency Domain → High pass Filters → Gaussian

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$





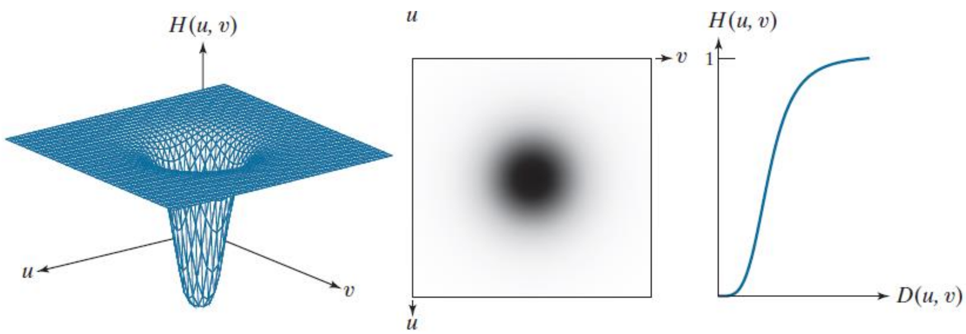
$D_0 = 60$

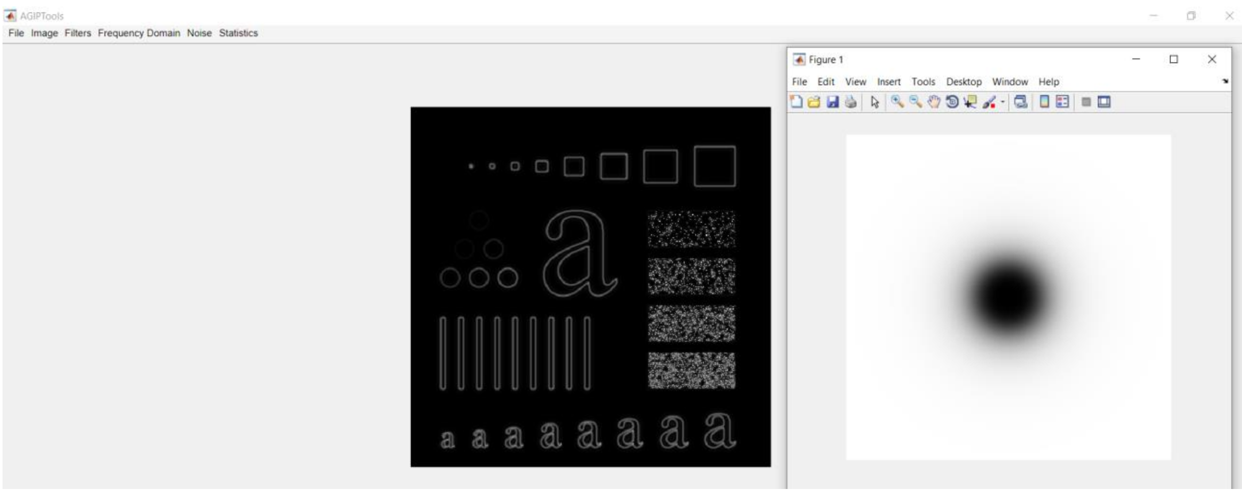


$D_0 = 30$

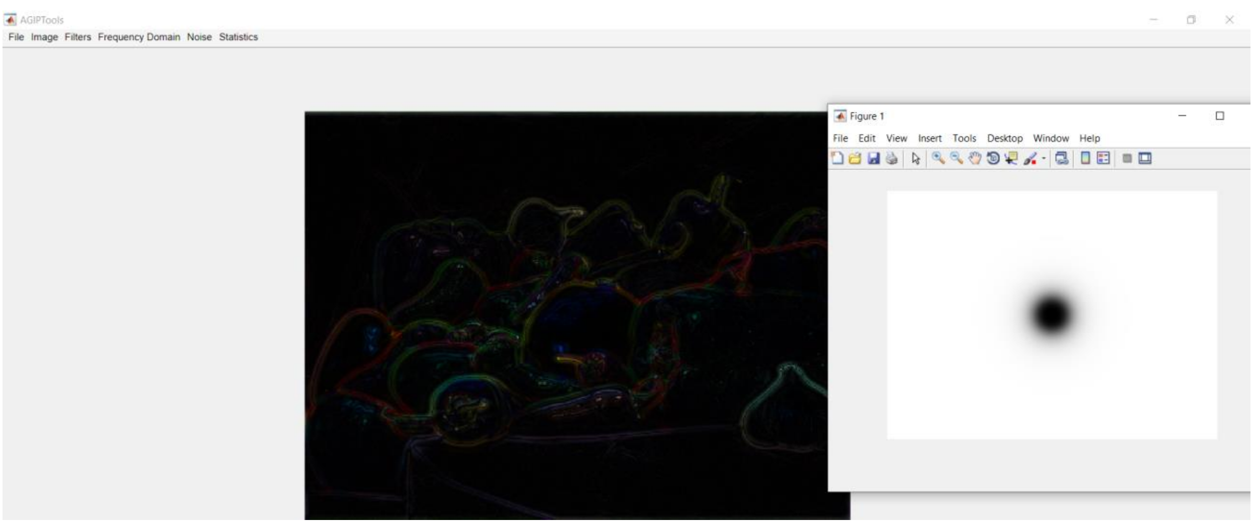
Frequency Domain → High pass Filters → Butterworth

$$H(u,v) = \frac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$$





D_0 = 60, n = 2



D_0 = 30, n = 2

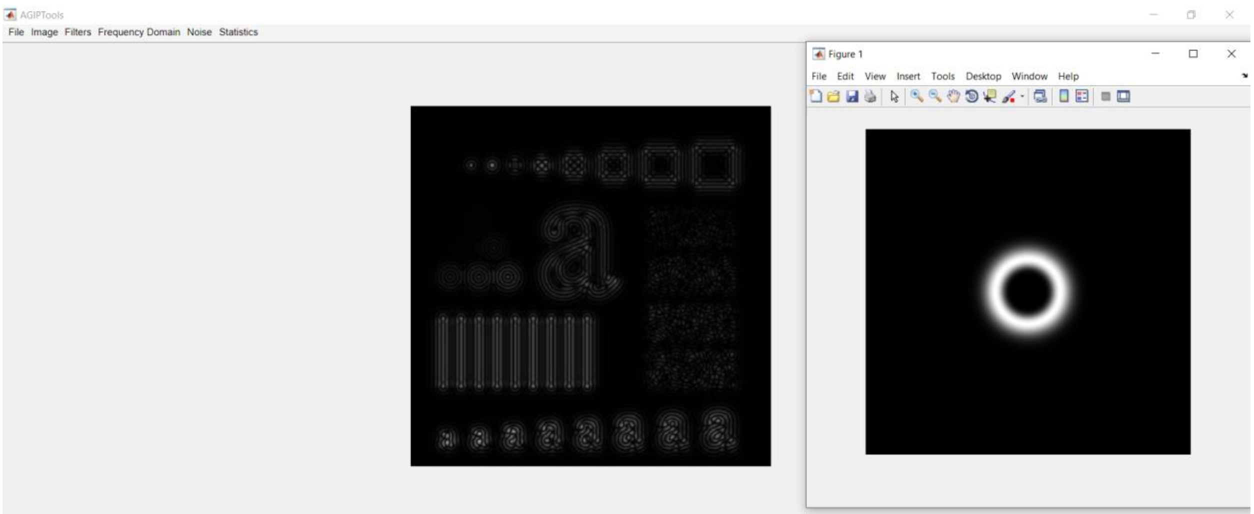
Frequency Domain → Band pass Filters

Frequency Domain → Band reject Filters

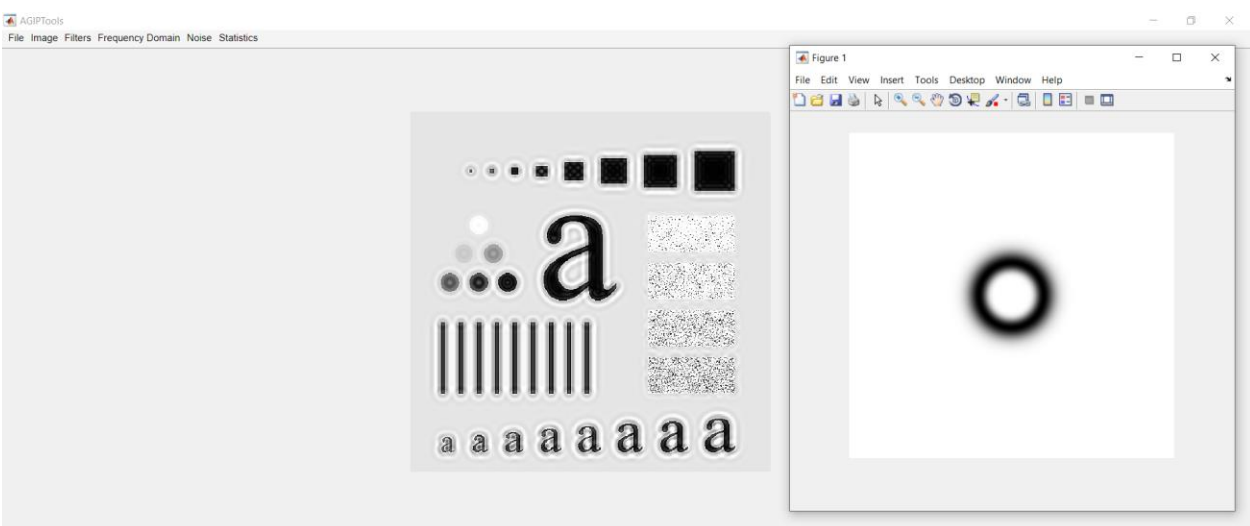
TABLE 4.6

Highpass filter transfer functions. D_0 is the cutoff frequency and n is the order of the Butterworth transfer function.

Ideal	Gaussian	Butterworth
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$



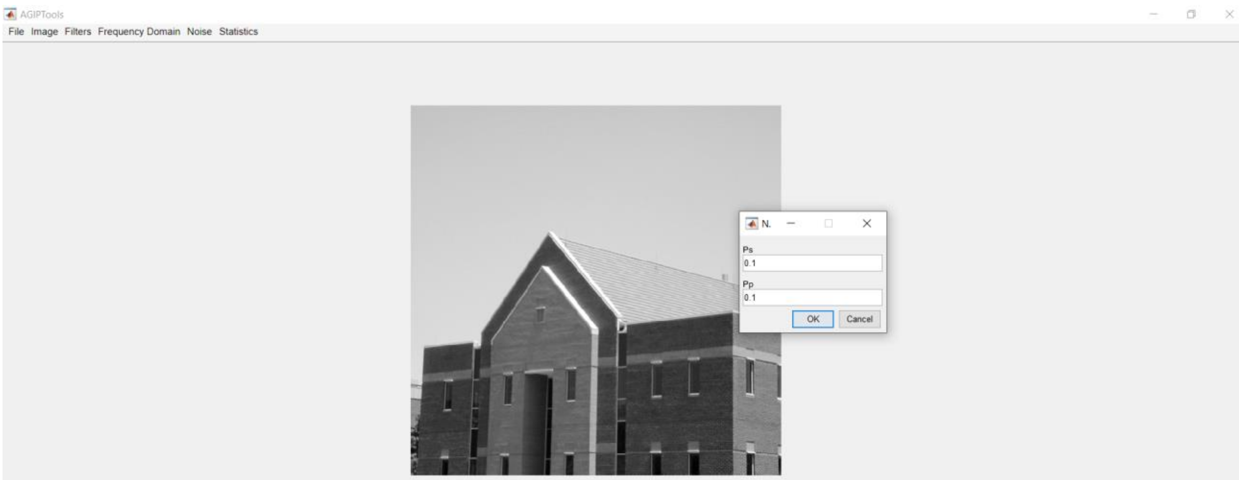
Band pass



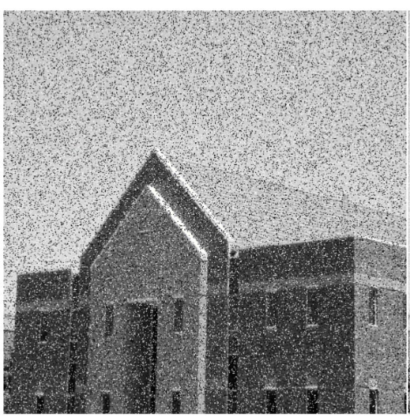
Band reject

Noise → Salt and Pepper

$$
p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1 - (P_s + P_p) & \text{for } z = V \end{cases}
$$

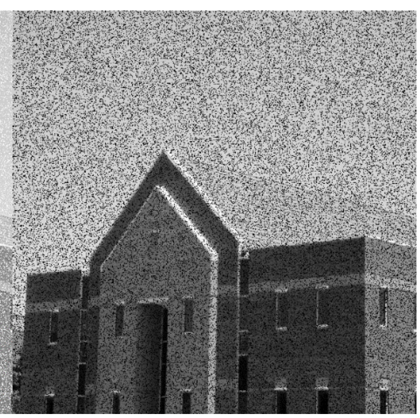


Original



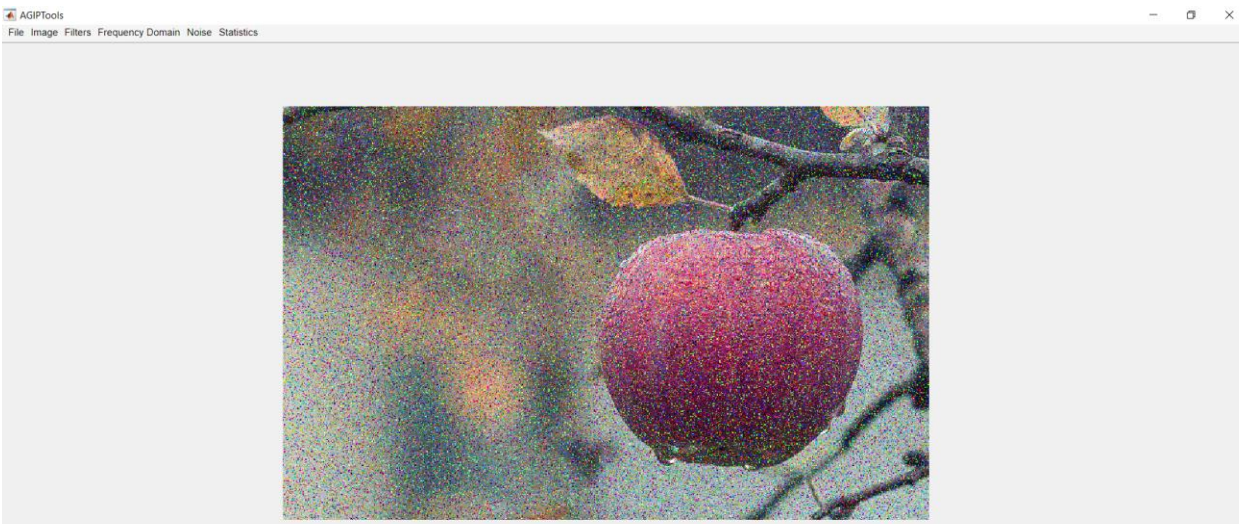
Ps=0.1, Pp=0.1

Ps=0.2

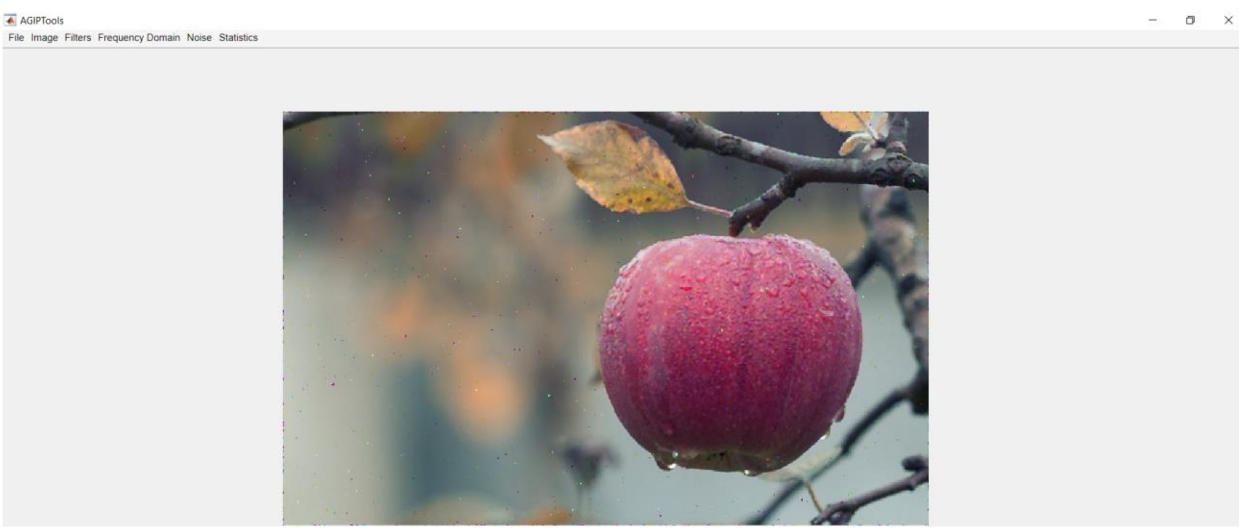


Pp=0.2

3*3 Median Filter



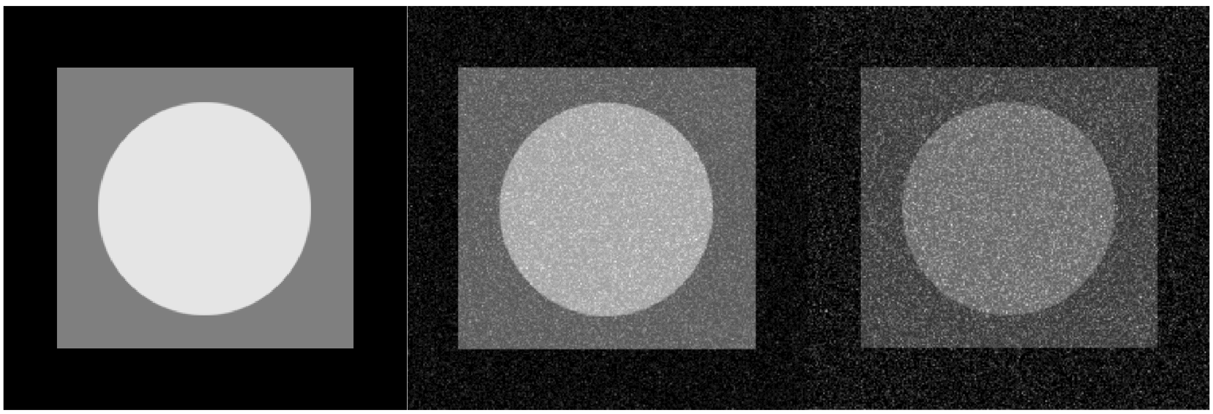
Ps=0.1, Pp=0.1



3*3 Median Filter

Noise → Gaussian

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}}$$

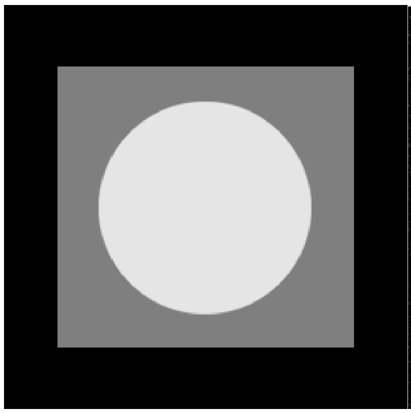


Original | Mean=20, Sigma=5 | Mean=50, Sigma=5

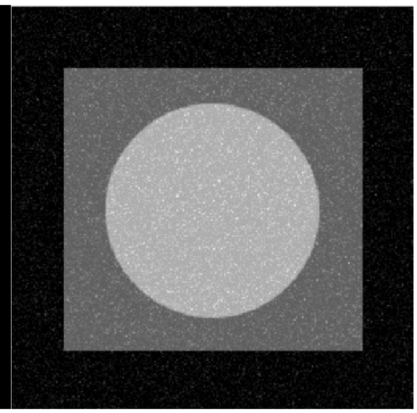
Original | Mean=20, Sigma=5

Noise → Exponential

$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$



Original



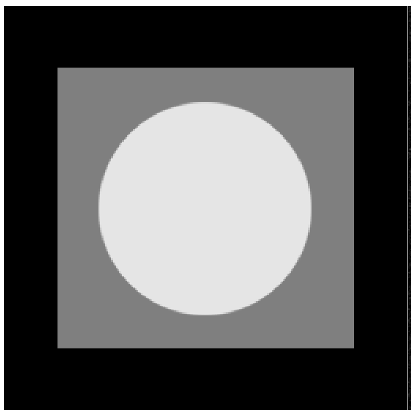
Exp Noise

Original

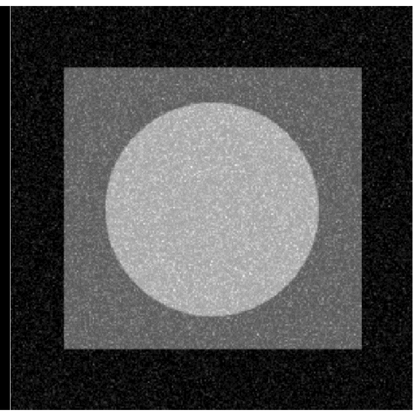
Exp Noise

Noise → Rayleigh

$$p(z) = \begin{cases} \dfrac{2}{b}(z - a)e^{-(z - a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$$



Original



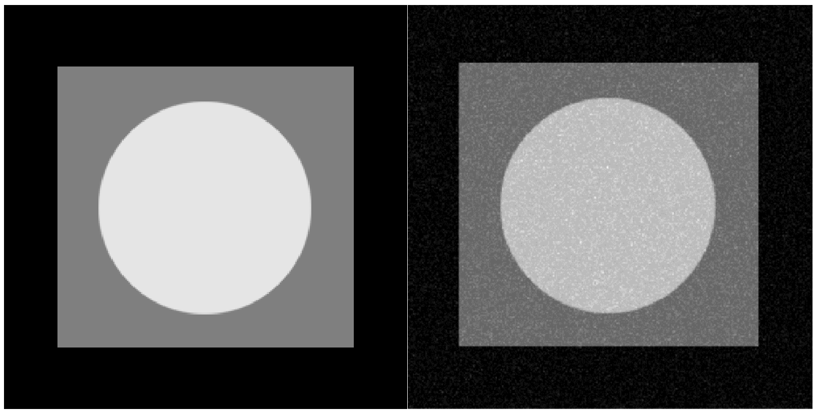
Rayleigh Noise

Original

Rayleigh Noise

Noise → Gamma

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

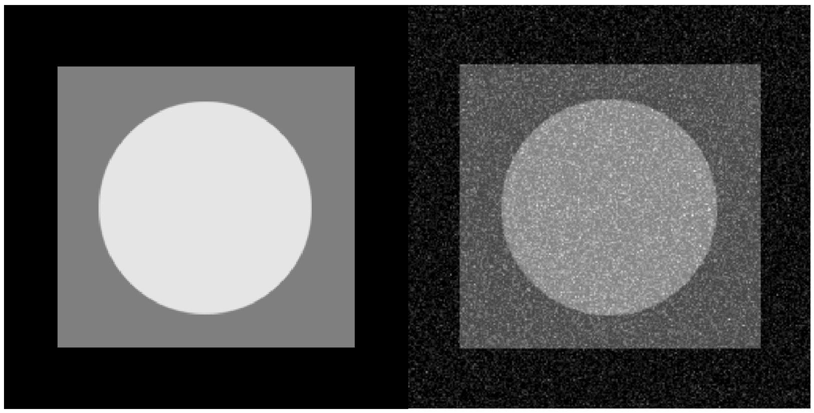


Original Gamma Noise

Original Gamma Noise

Noise → Uniform

$$p(z) = \begin{cases} \dfrac{1}{b - a} & a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$



Original Uniform Noise

Original Uniform Noise