



image  

---

processing

# IMAGE PROCESSING LAB-2

Mounika Reddy PATIMIDHI

Master's In International Biometrics - UPEC

## Table of Contents

|  |   |
|--|---|
| 1.HISTOGRAM .....  | 2 |
| 1.1 Mean and Median standard deviation values of an image using mean2 and std2 ..... | 2 |
| 1.2 Visualization .....  | 2 |
| 1.3 HISTOGRAM PROPERTIES.....  | 3 |
| 1.4 Histogram Equalization .....   | 4 |
| 2.THRESHOLDING.....  | 5 |
| 2.1 Converting the image into binary image .....                                     | 5 |

## 1.HISTOGRAM

### 1.1 Mean and Median standard deviation values of an image using mean2 and std2

Mean is linear filtering and nonlinear filtering is Median.

```
I = imread('chro.bmp');
```

```
A = mean2(I);
```

```
B = std2(I);
```

```
C = median(I);
```

%this median function will give us a matrices in order to get a number we

%should use (:)



The image shows a MATLAB Workspace window with a table of variables. The table has two columns: 'Name' and 'Value'. The variables listed are A, B, C, and I. A is a scalar value 112.9408, B is a scalar value 32.5270, C is a 1x256 uint8 array, and I is a 256x256 uint8 array.

| Name | Value         |
|------|---------------|
| A    | 112.9408      |
| B    | 32.5270       |
| C    | 1x256 uint8   |
| I    | 256x256 uint8 |

### 1.2 Visualization

A visualization of the default matplotlib colormaps is available here. As matplotlib does not directly support colormaps for line-based plots, the colors are selected based on an even spacing determined by the number of columns in the Data Frame.

```
I = imread('chro.bmp');
```

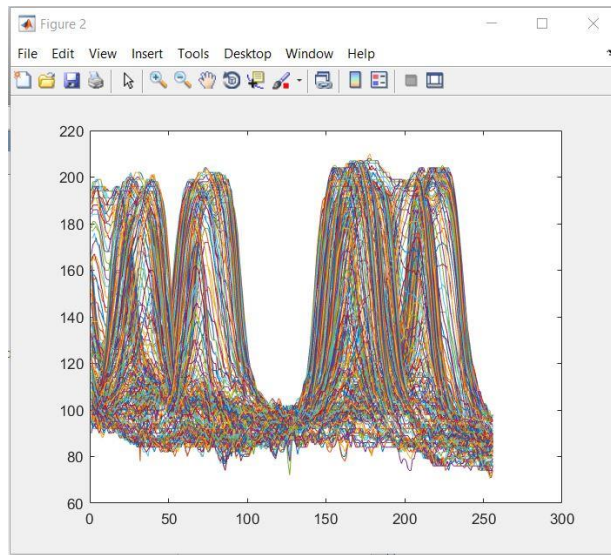
```
subplot(1,3,1),imhist(I);
```

```
subplot(1,3,2),imhist(I,64);
```

```
subplot(1,3,3),imhist(I,16);
```

```
B = cumsum(I);
```

```
figure;
```



### 1.3 HISTOGRAM PROPERTIES

Histogram properties control the appearance and behavior of the histogram. By changing property values, you can modify aspects of the histogram. Use dot notation to refer to a particular object and property:

```
h = histogram(randn(10,1));
```

```
c = h.BinWidth;
```

```
h.BinWidth = 2;
```

%%%%%%%%Histogram with properties:

```
nbins = 37;
```

```
NumBins = 23;
```

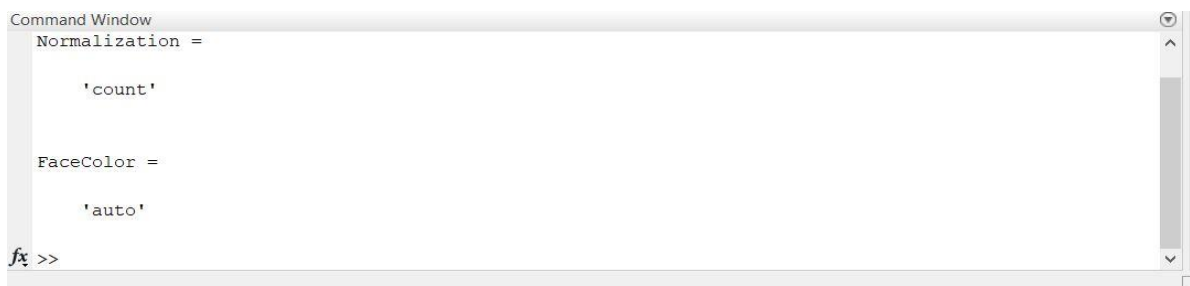
```
BinWidth = 0.3000;
```

```
BinLimits = [-3.3000 3.6000];
```

```
Normalization = 'count'
```

```
FaceColor = 'auto'
```

```
EdgeColor = [0 0 0];
```



## 1.4 Histogram Equalization

```
%%%1.4
```

```
[m,n] = size('inputIm');
```

```
enhancedIm=uint8(zeros(m,n));
```

```
freq=zeros(1,6);
```

```
HistEq=zeros(1,6);
```

```
Map = zeros(m,n);
```

```
i=1:m
```

```
j=1:n
```

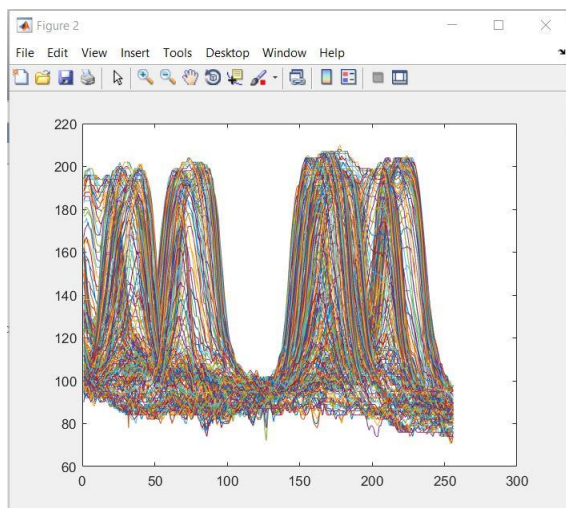
```
i=1:length(HistEq)
```

```
x = randn(10,1);
```

```
h = histogram(x);
```

```
BinWidth: (0.20);
```

```
BinLimits: [-3.28 0.36]
```



| Workspace     |                      |
|---------------|----------------------|
| Name          | Value                |
| A             | 112.9408             |
| ans           | 0                    |
| B             | 256x256 uint8        |
| BinLimits     | [-3.3000,3.6000]     |
| BinWidth      | 0.3000               |
| C             | 1x256 uint8          |
| EdgeColor     | [0,0,0]              |
| enhancedIm    | [0,0,0,0,0,0]        |
| FaceColor     | 'auto'               |
| freq          | [0,0,0,0,0,0]        |
| h             | 1x1 Histogram        |
| HistEq        | [0,0,0,0,0,0]        |
| i             | [1,2,3,4,5,6]        |
| I             | 256x256 uint8        |
| j             | [1,2,3,4,5,6,7]      |
| m             | 1                    |
| Map           | [0,0,0,0,0,0]        |
| n             | 7                    |
| nbins         | 37                   |
| Normalization | 'count'              |
| NumBins       | 23                   |
| x             | [-1.3499;3.0349;0... |

## 2.THRESHOLDING

### 2.1 Converting the image into binary image

%%%%%binarize image using im2bw

```
I = imread('gdr.bmp');
```

```
a = im2bw(I);
```

```
imshow(a)
```

