Dr Deepak B Phatak
Subrao Nilekani Chair Professor
Department of CSE, Kanwal Rekhi Building
IIT Bombay

Lecture 17, Analysis of Midsem (Contd.)

(Slides on Image histogram from Wikipedia)

Q 4 Find peculiar array elements

Two arrays A and B have M and N elements respectively. All elements are integers. Assume that the array A is sorted in increasing order of numbers. There may exist pairs of elements of the type (x, y) such that x is an element of A and y is an element of B, and if these are swapped across the arrays, then the resultant arrays will have the property that sum of elements of new array A = sum of elements in new array B

Q 4 …

```
#include<iostream>
#include<math.h>
using namespace std;
int main(){
  int A[100], B[100], M, N, i, j sum1, sum2, diff, x, y;

  // Read numbers in two arrays
   cin >> M;
  for (i=0; i<M; i++) cin >> A[i];
  cin >> N;
  for (i=0; i<M; i++) cin >> B[i];
```

Q 4 …

```
// Find sum of all elements of each array
    sum1 = 0;
    for (i=0; i<M; i++) sum1 += A[i];
    sum2 = 0;
    for (i=0; i<M; i++) sum2 += B[i];
    diff = abs (sum1 - sum2);
    if (diff%2 !=0) {
      cout << “Difference of sums is not even”;
      cout << “Desired elements do not exist”;
      cout << endl;
      return  1;
    }
```

Q 4 …

```
// Now locate the elements meeting the criterion
// if x is element of A, and y element of B,
// we need to  find x and y such that
// sum1 +y = sum2 + x
// or x = sum1 - sum2 + y
//
// In general, for some j, if we are looking at B[j]
// then for some I, we must have
// A[i] = sum1 –sum2 + B[j]
```

Q 4 …

```
for (j=0; j < N; j++){   // start with some B[j]
   x = sum1 - sum2 + B[j]; // Find  desired value x
      for (i=0; i < M; i++){
         if (A[i] == x){    // search for x in array A
            cout >>  A[i] >> “ “ >> B[j];
         }
         else{
            break;   // abandon search,
         }
      }
   }
```

Q 4 (b) …

If M = N, the problem “size” is N

In our program,

We have an outer iteration executing N times

- every time considering one value from B[]

  For each outer iteration,

  - we have an inner iteration also executing N times checking every value in A[]

So the complexity of this algorithm is O(N)

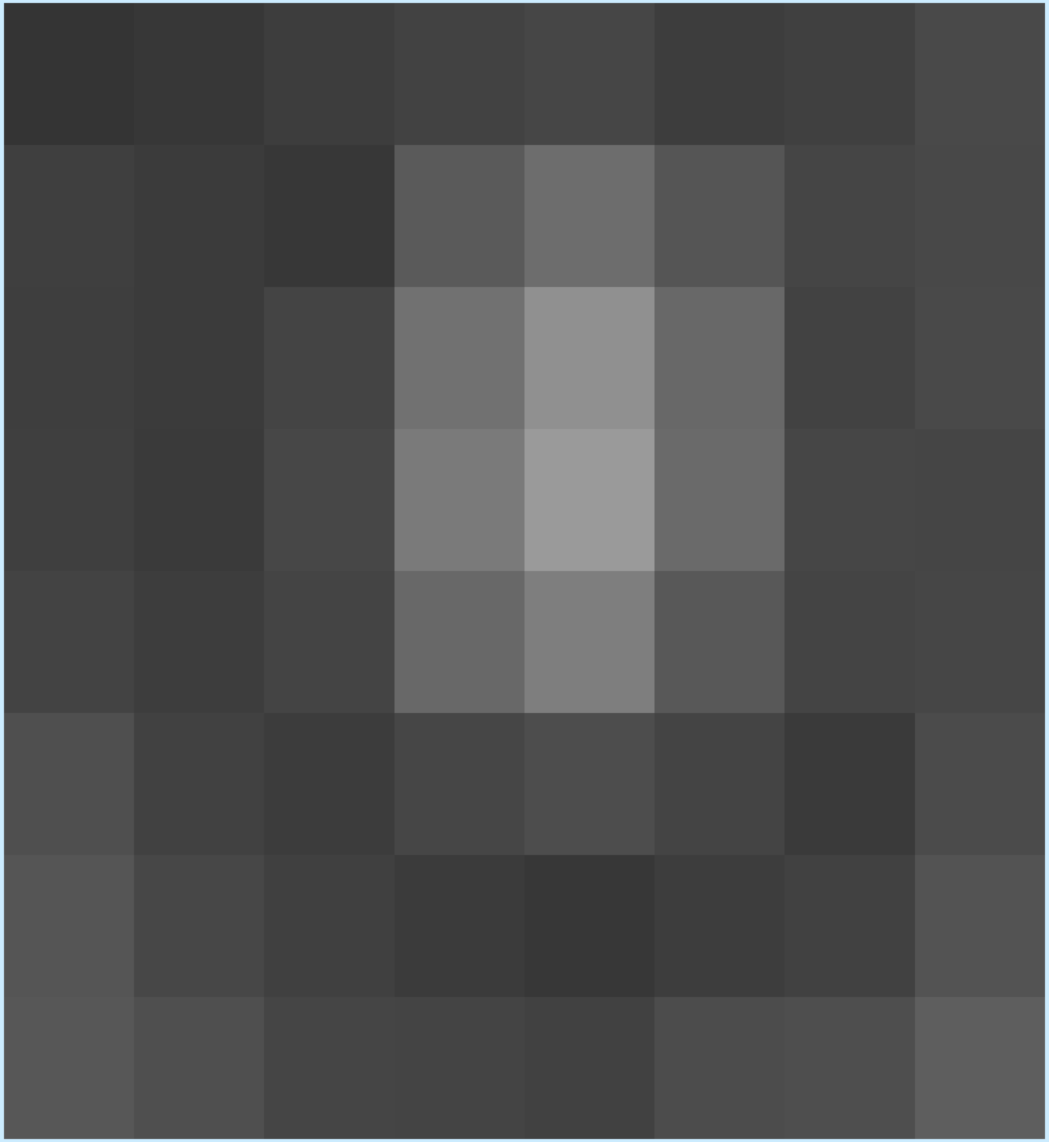
- If we search A using binary search, the inner iteration will execute only Log_2 N times

Complexity will be O(N $\log_2$N)

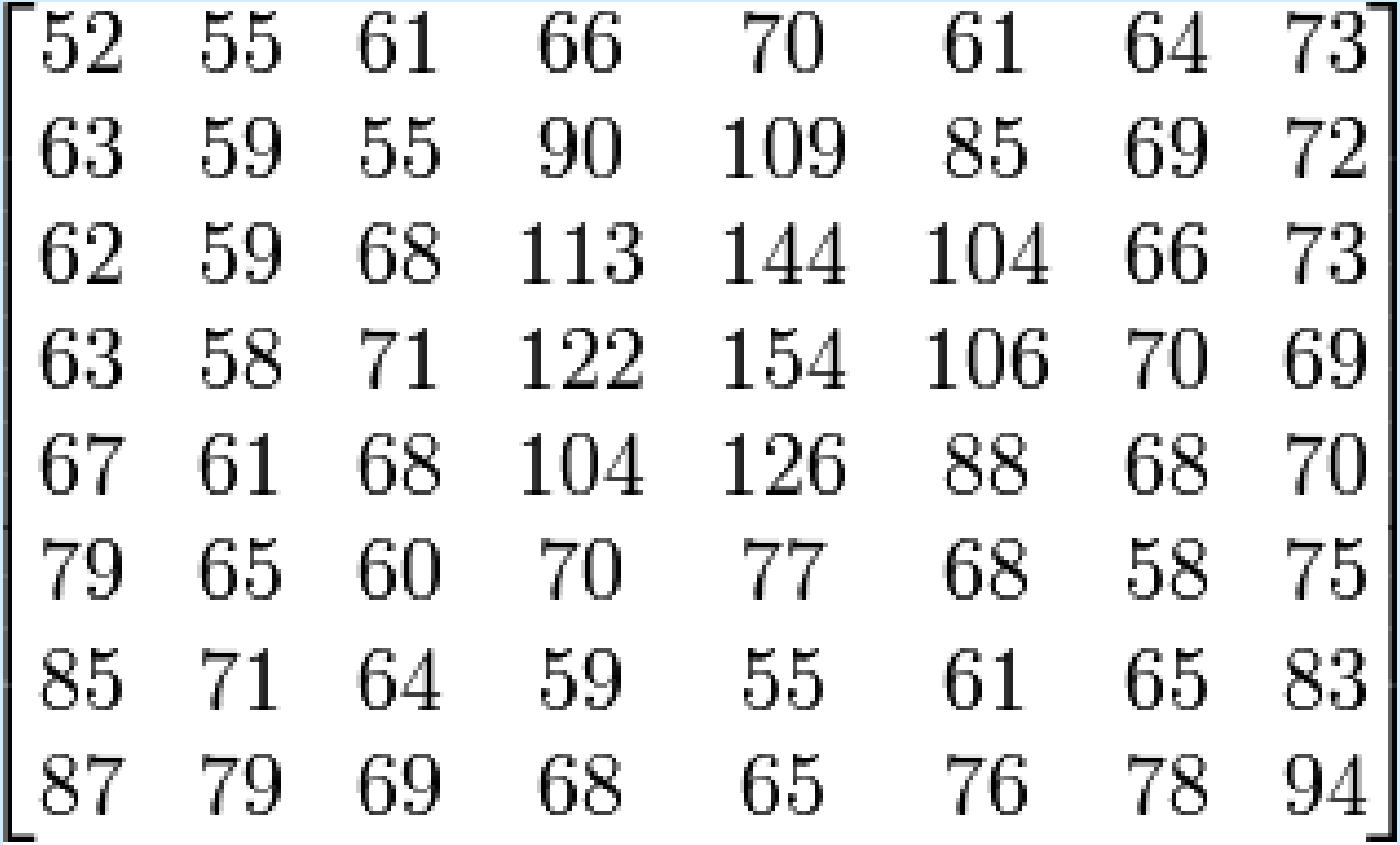
Images and histograms

- Images are stored as a large number of pixels
  - Arranged in a matrix form
- Each picture point (pixel) has an associated “tonal” value
  - For grayscale images, the value range is 0-255
  - 0: Black, 255: White
- Thus each element of an image[500][500] matrix would contain a value as above
- Histogram indicates how many pixels have the same value

A sample image 8 pixel x 8 pixel



Pixel values in the sample image



$$\begin{bmatrix}
52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\
63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\
62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\
63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\
67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\
79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\
85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\
87 & 79 & 69 & 68 & 65 & 76 & 78 & 94
\end{bmatrix}$$

Histogram values

Val	n	Val	n	Val	n	Val	n	Val	n
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Cumulative Distribution Function (cdf)

V	c	V	c	V	c	V	c	V	c
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

Histogram equalization

- The histogram equalization formula

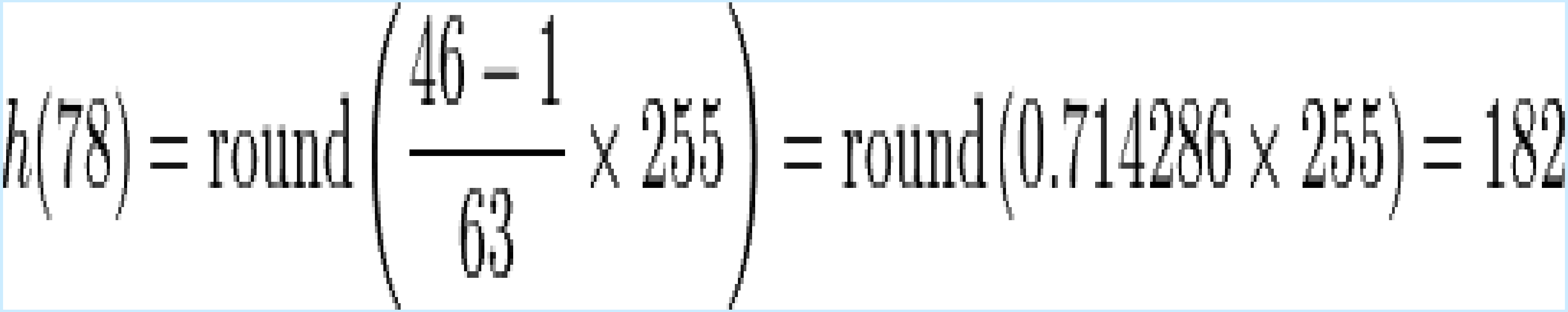
$$h(v) = \mathrm{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1)\right)$$

- “Equalization” formula for example image

$$h(v) = \mathrm{round}\left(\frac{cdf(v) - 1}{63} \times 255\right)$$

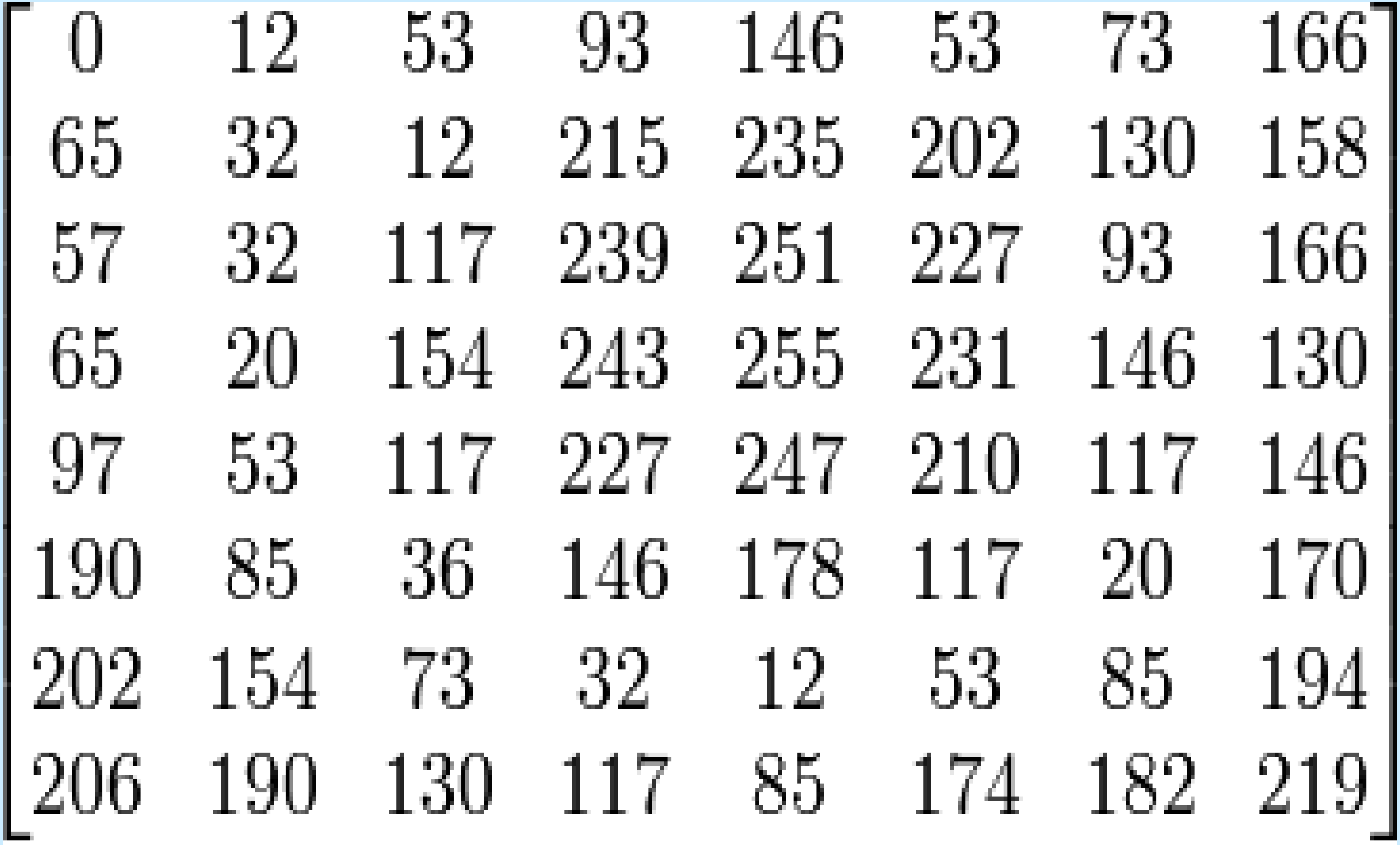
Histogram equalization …

- For example, the cdf of 78 is 46



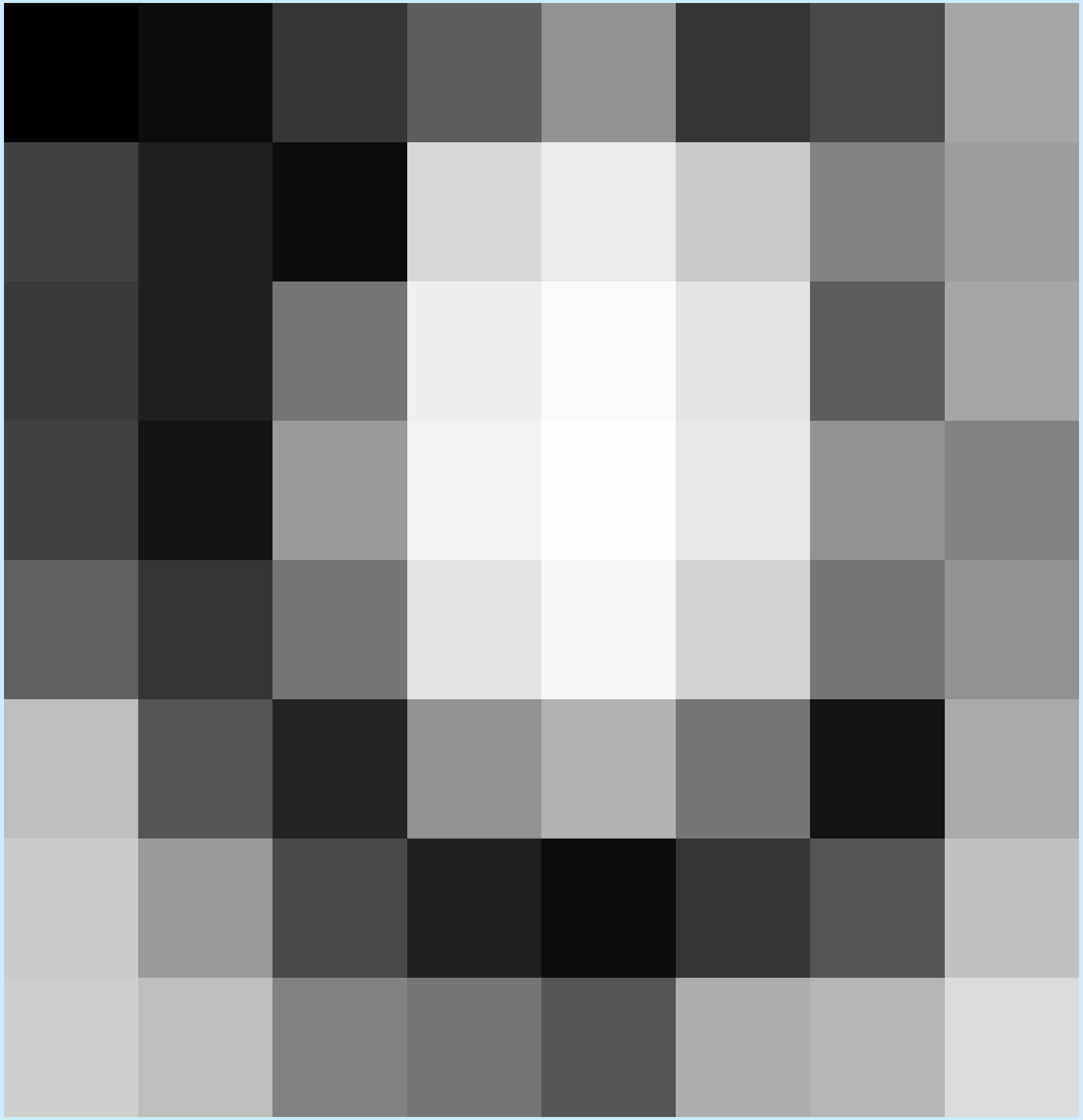
$$h(78) = \text{round}\left(\frac{46-1}{63} \times 255\right) = \text{round}(0.714286 \times 255) = 182$$

Pixel values after histogram equalization

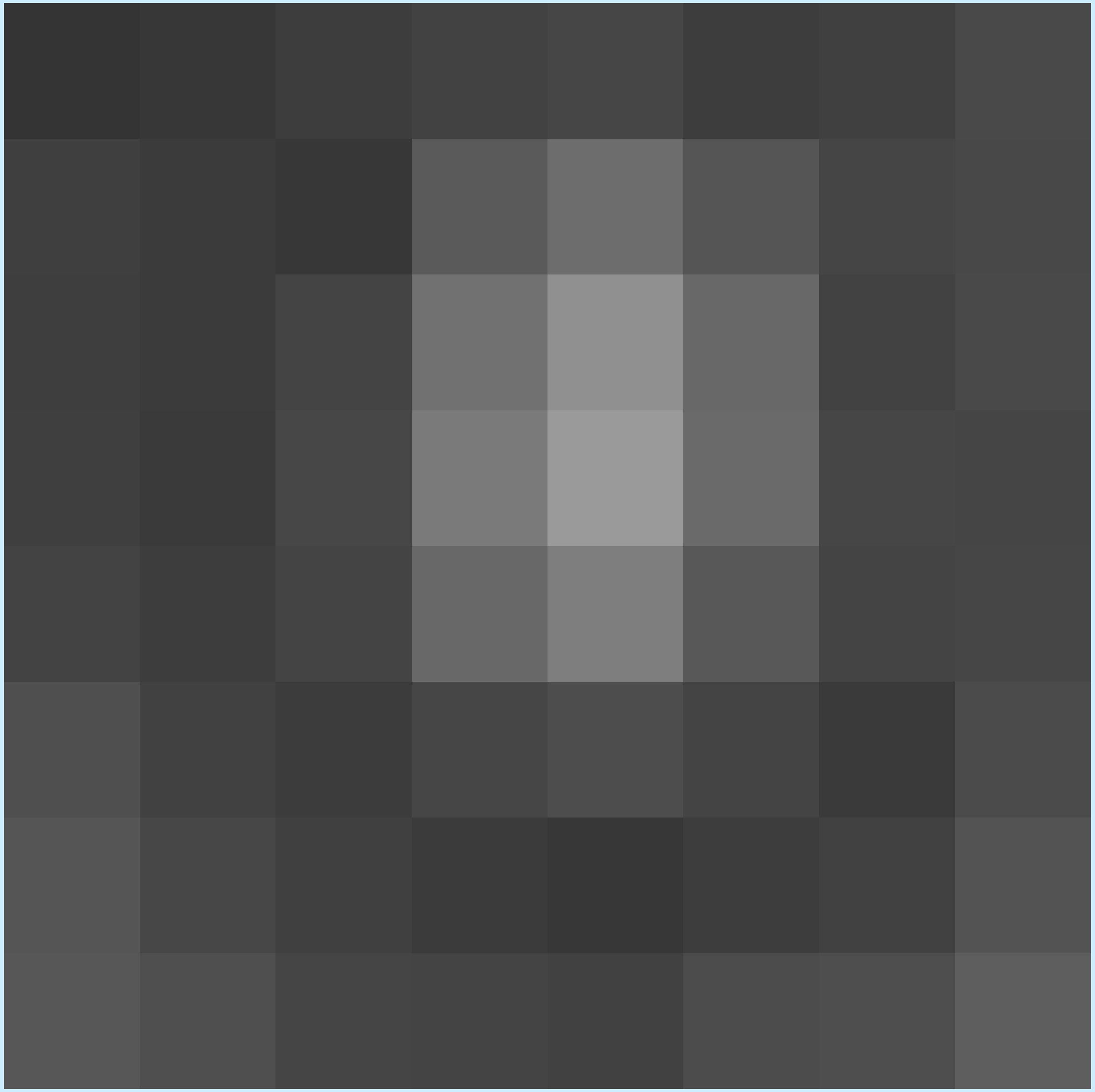


$$\begin{bmatrix}
0 & 12 & 53 & 93 & 146 & 53 & 73 & 166 \\
65 & 32 & 12 & 215 & 235 & 202 & 130 & 158 \\
57 & 32 & 117 & 239 & 251 & 227 & 93 & 166 \\
65 & 20 & 154 & 243 & 255 & 231 & 146 & 130 \\
97 & 53 & 117 & 227 & 247 & 210 & 117 & 146 \\
190 & 85 & 36 & 146 & 178 & 117 & 20 & 170 \\
202 & 154 & 73 & 32 & 12 & 53 & 85 & 194 \\
206 & 190 & 130 & 117 & 85 & 174 & 182 & 219
\end{bmatrix}$$

Enhancement of Contrast

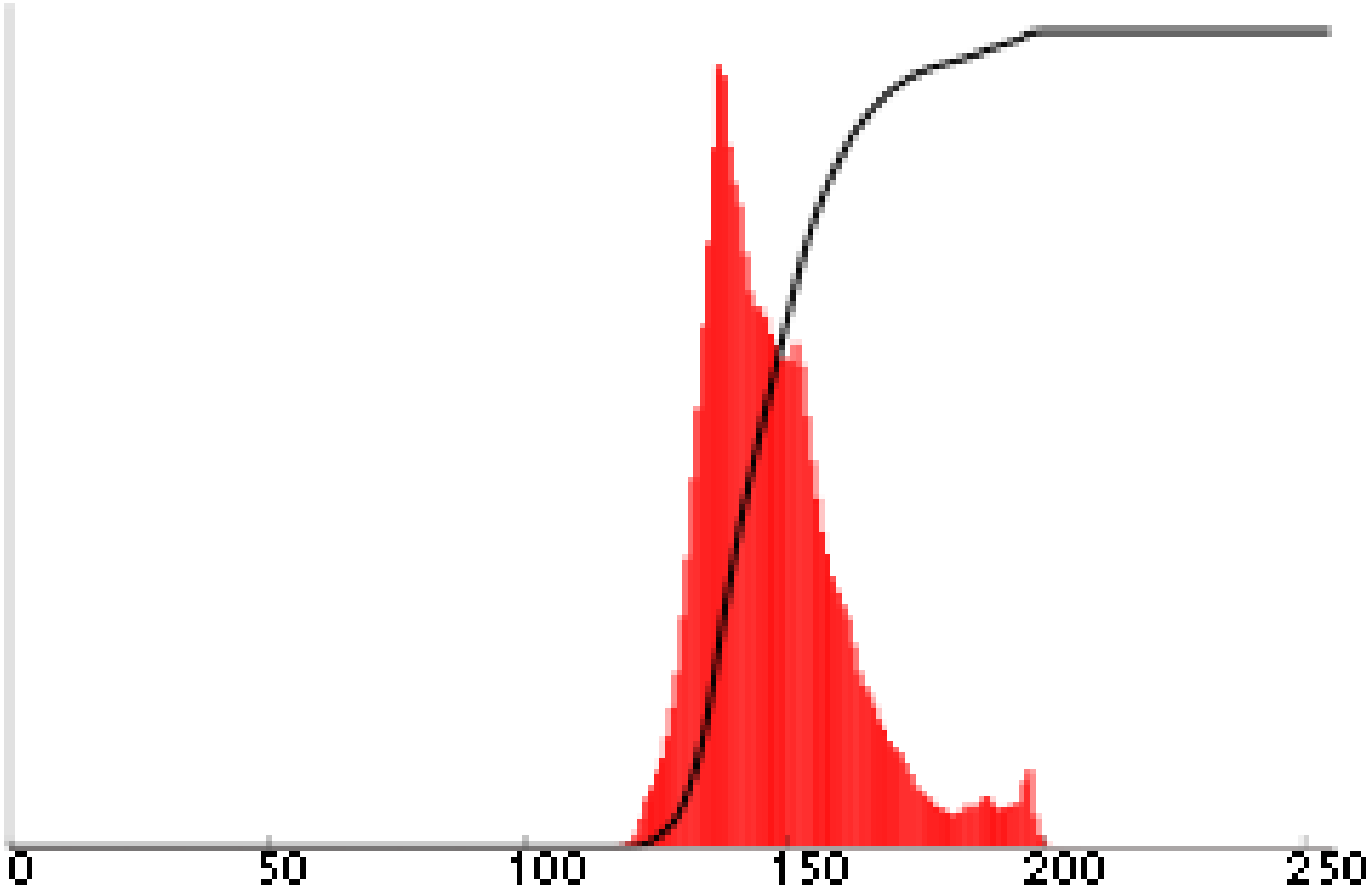


Original Picture for comparison

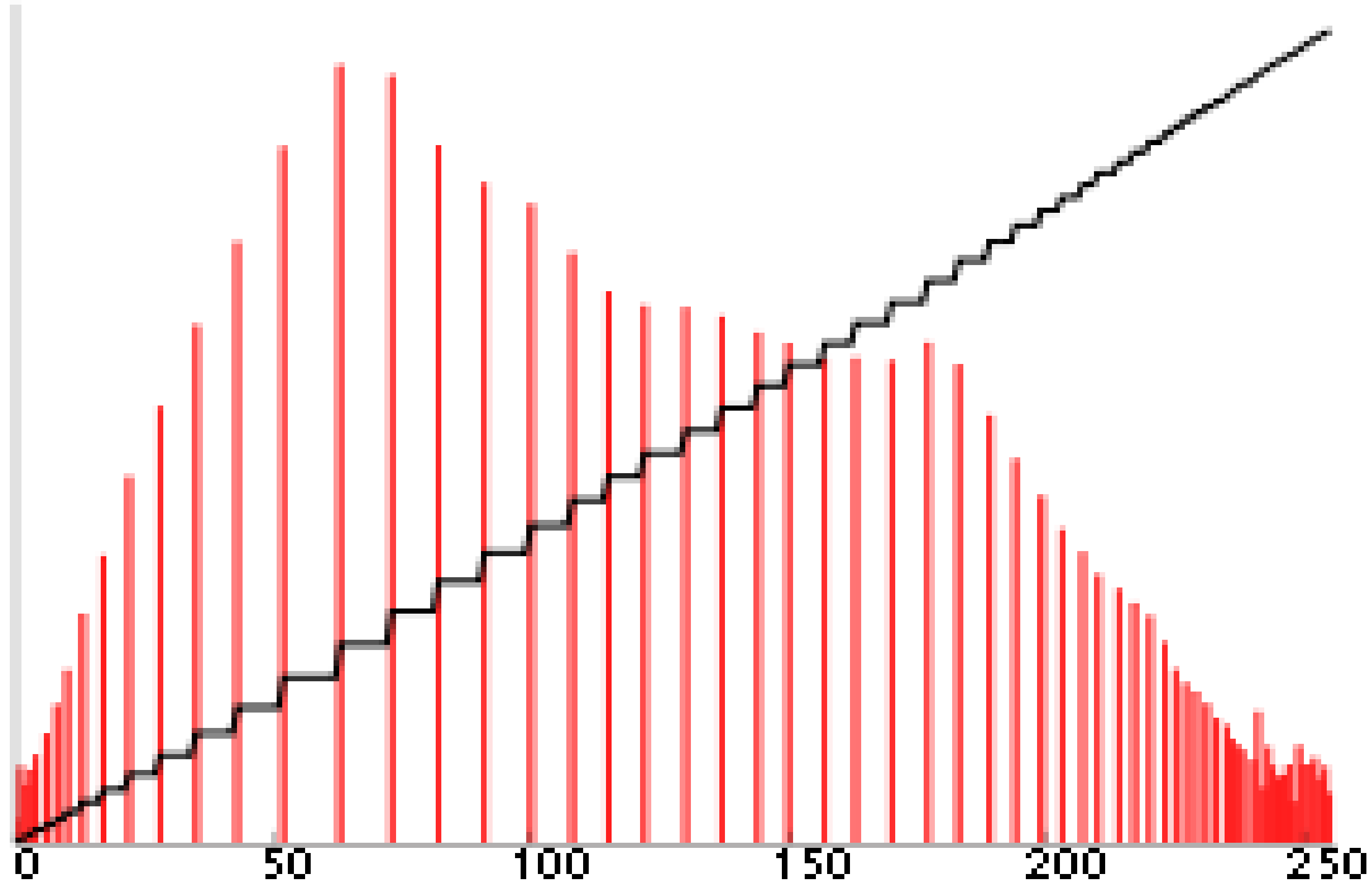


Another grayscale picture

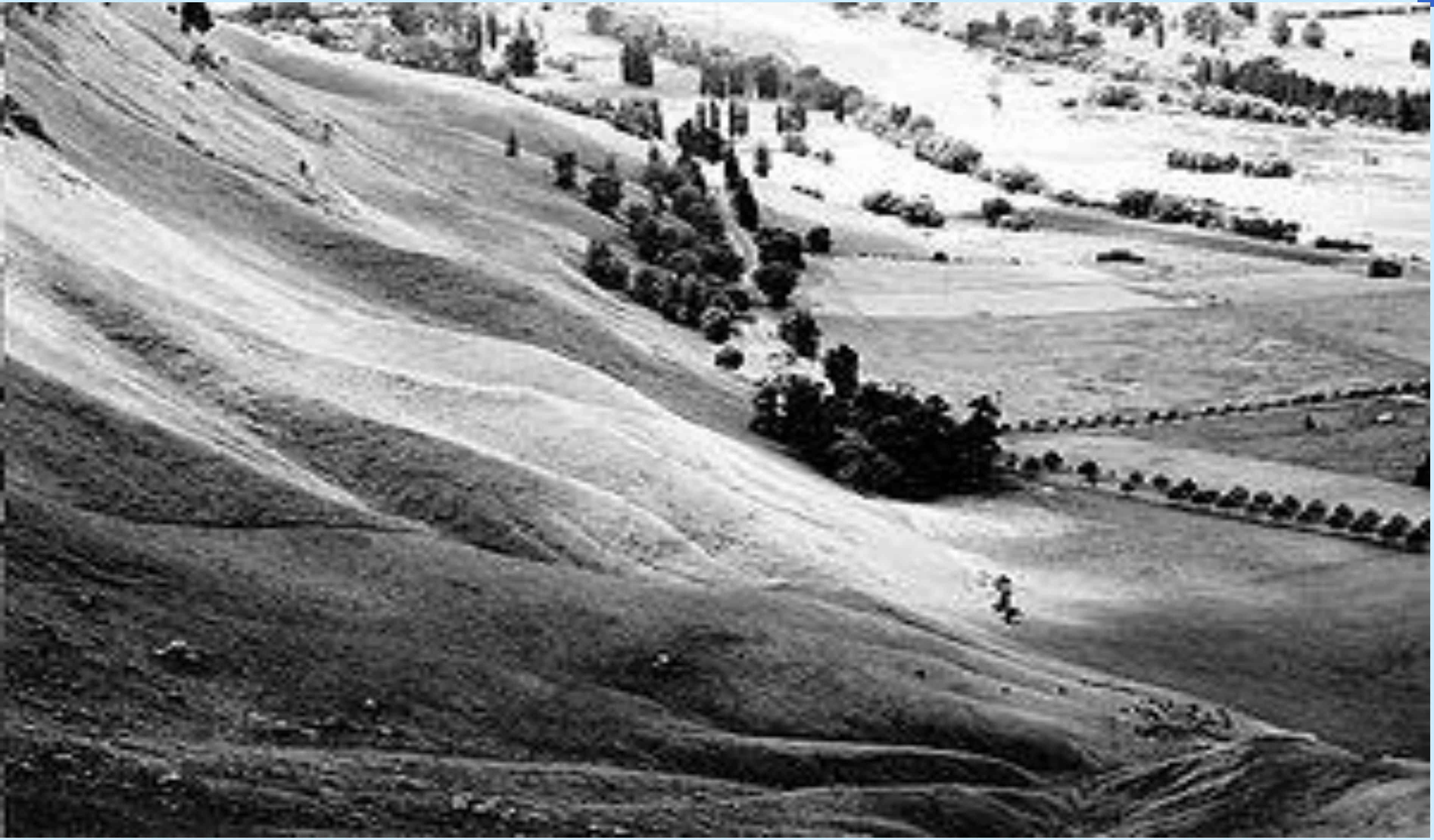
Histogram and cdf



“Equalized” histogram and cdf



Picture with enhanced contrast



Original picture for comparison

Program to calculate histogram

```
#include<iostream>
#include<fstream>
using namespace std;
int main(){
int i, j, npix;
int image[500][500], histogram[256];
ifstream ImageFile("image.txt");
```

Histogram …

```
if(ImageFile.is_open()){
   ImageFile >> npix;   // number of pixels
   //read image pixel values in the matrix
   for(i=0; i< npix; i++){
      for(j=0; j < npix; j++){
        ImageFile >> image[i][j];
        cout <<image[i][j]<< " ";
      }
   cout << endl;
   }
}
```

Histogram …

```
else {
  cout<<endl<<"unable to open file";
  return 1;
}
ImageFile.close();
```

Histogram …

```
// set histogram counts to zero
for(i=0;i< 256;i++) histogram[i] = 0;
// Calculate histogram values
for(i=0;i< npix; i++){
   for (j = 0; j < npix; j++){
   histogram[image[i][j]]++;
   }
}
```

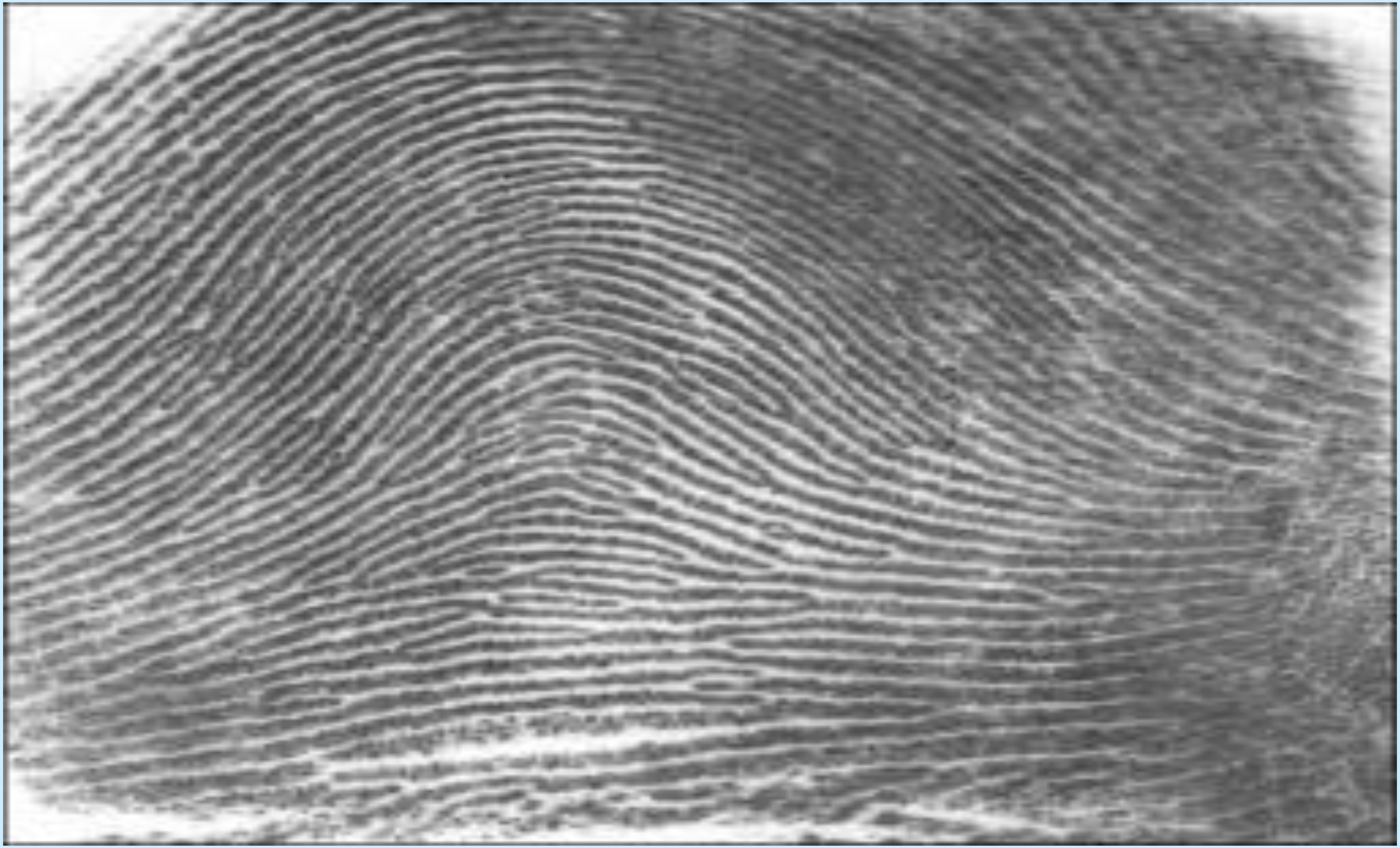
Histogram …

```
// print the histogram at non zero values
cout << "Histogram at non zero values is:" << endl;
for (i=0; i<256; i++){
   if (histogram[i] !=0) {
      cout << i << "  " << histogram[i] << endl;
   }
}
```

Histogram …

```
 // find the maximum value in the histogram
  int imax, max = 0;
  for (i=0; i<256; i++){
    if (histogram[i] > max) max = histogram[i];
  }
  cout << "Maximum histogram value " << max ;
  cout <<  “   occurs at:”  << endl;
  // Print the grey levels at which max value occurs
  for(i=0; i<256; i++){
      if (histogram[i]== max) cout << i << "   ";
  }
  return 0;
}
```

Image of a finger print



Announcements

Form teams of 4 members within your lab batch

Team leader to Submit a text file to lab TA

Format:

Roll Number	Name	Hostel	Room	Remark
- - -	- -	- - -	- - -	
- - -	- -	- - -	- - -	
-- - -	- -	- - -	- - -	Team leader
- - -	- -	- - -	- - -	

(5th person, only for some batches)

Announcements

Many students have scored over 80% marks in mid-semester exam (>=36)

Congratulations!

The names and scores will be announced on the course home page