Image processing
- Point Operators

Objectives

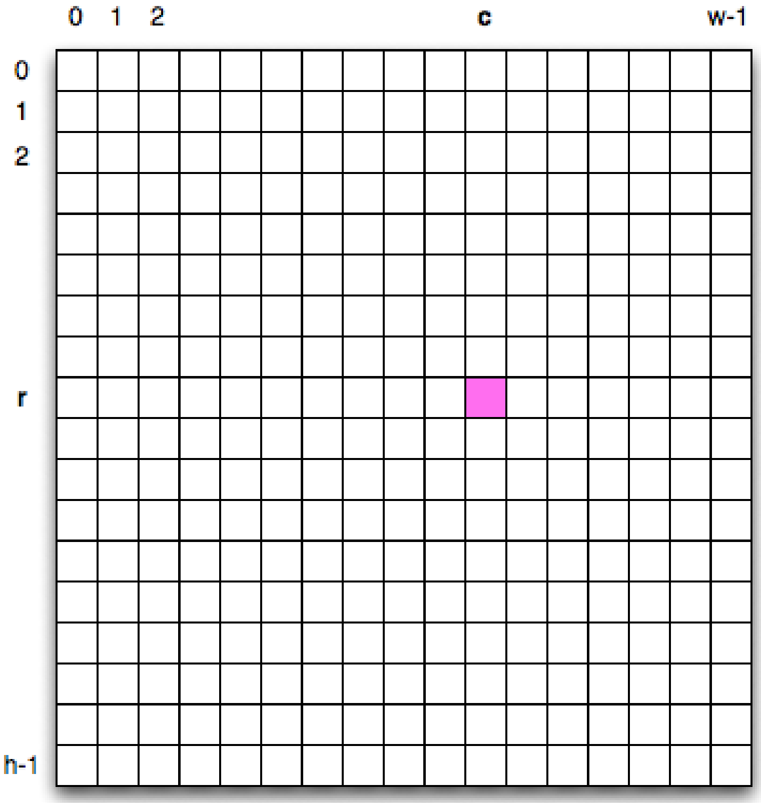
- Learn Image processing
- Image Arithmetic
- Understand Pixel transforms
- Color transforms
- Histogram based image operations

Image processing

- Digital image processing is the use of a digital computer to process digital images through an algorithm. The generation and development of digital image processing are mainly affected by three factors:
  - The development of computers.
  - The development of mathematics.
  - The demand for a wide range of applications in environment, agriculture, military, industry and medical science has increased.
- Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.
- It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image.

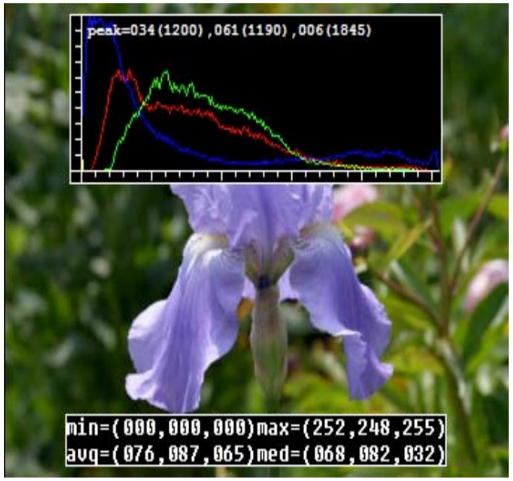
Image processing

- Image processing to preprocess the image and convert it into a form suitable for further analysis

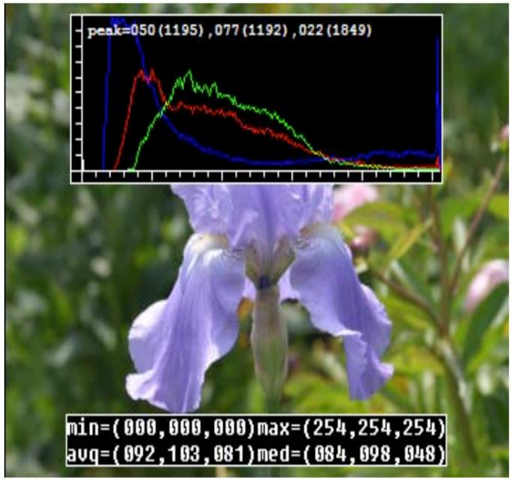


- Digital image is a two dimensional array of pixel
- Each pixel:
  - Intensity value- represented by a digital number
  - Position address- its row and column numbers
- How to calculate intensity value of pixel in RGB space?

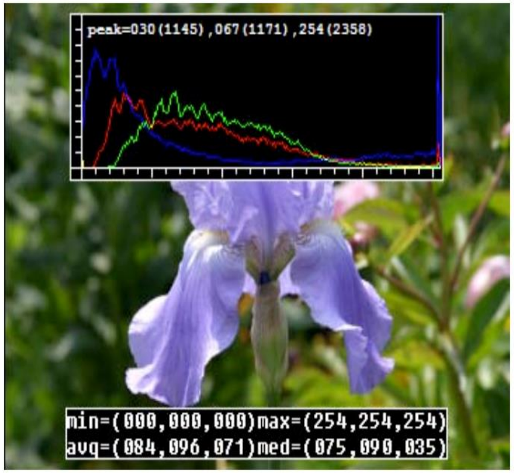
Image processing



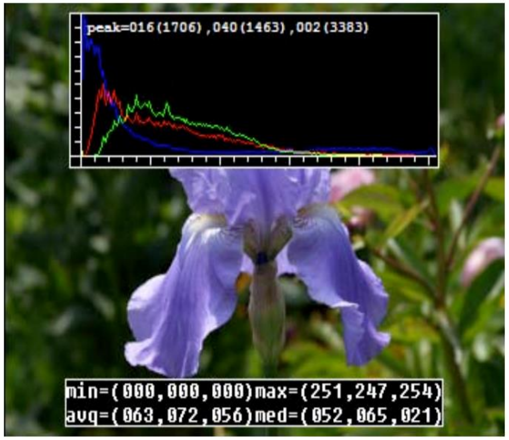
(a)



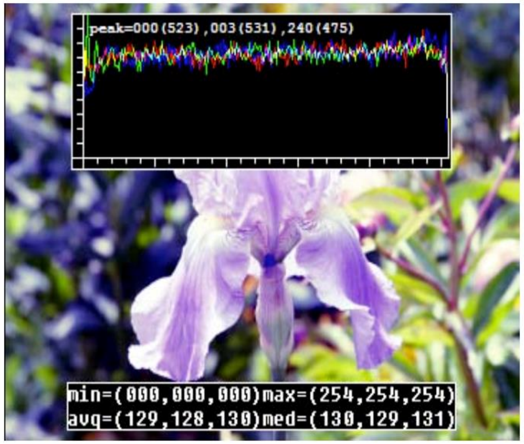
(b)



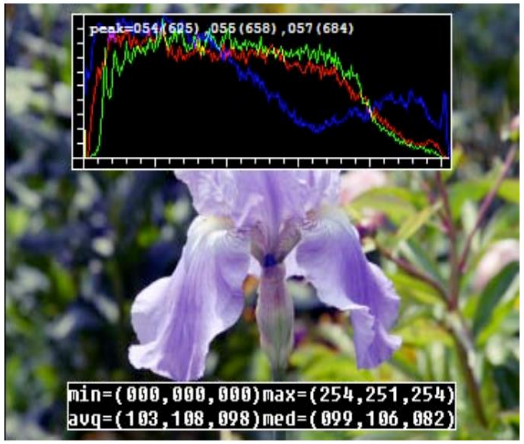
(c)



(d)



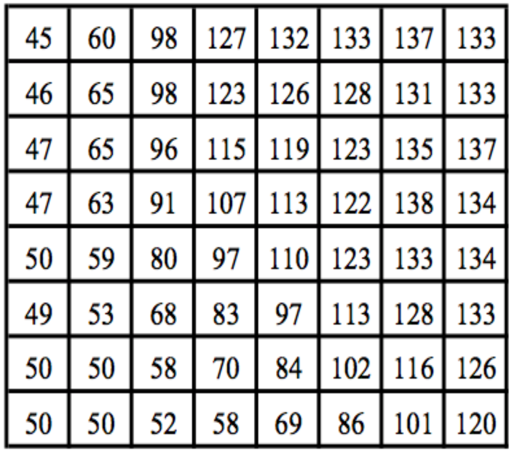
(e)



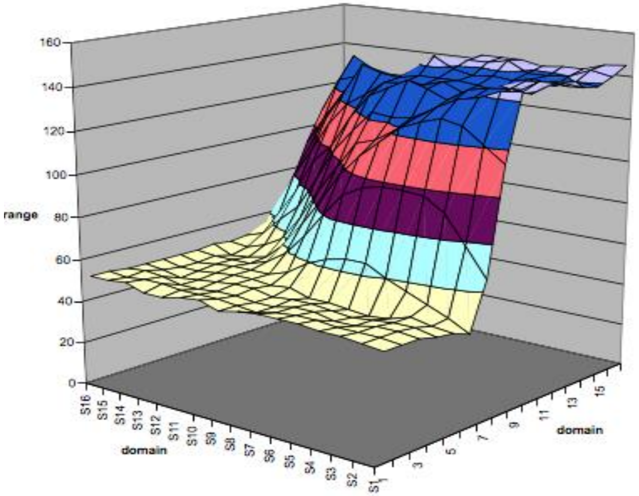
(f)

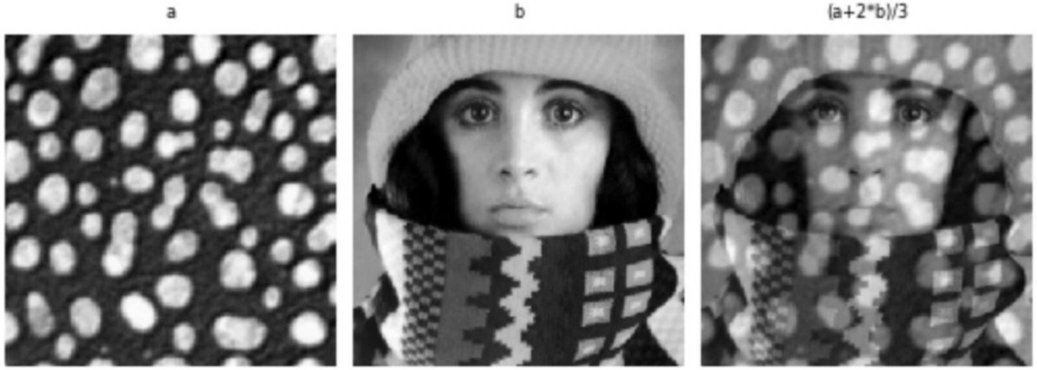
Pixel transforms

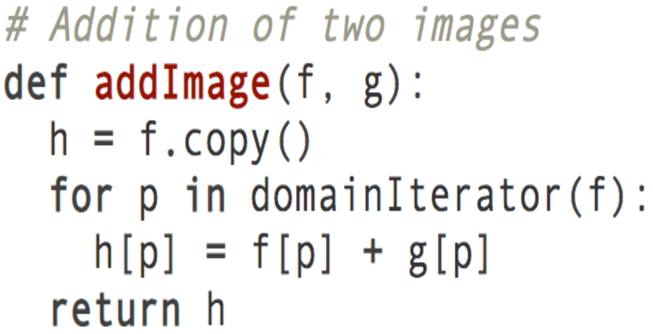
- Pixel-by-pixel processing



45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120







```
# Addition of two images
def addImage(f, g):
  h = f.copy()
  for p in domainIterator(f):
    h[p] = f[p] + g[p]
  return h
```

Pixel Operators

- Operator is a function that takes one or more input images and produces an output image:

$$g(\boldsymbol{x}) = h(f(\boldsymbol{x})) \text{ or } g(\boldsymbol{x}) = h(f_0(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$$

- Discrete (sampled) images, pixel locations, x = (i, j)

$$g(i,j) = h(f(i,j))$$

- Multiplication and addition with a constant

$$g(\boldsymbol{x}) = af(\boldsymbol{x}) + b$$

  – The parameters a > 0 and b are often called the gain and bias parameters

  – Control contrast and brightness

Pixel Operators

- Multiplicative gain -Superposition principle:

$$h(f_0 + f_1) = h(f_0) + h(f_1)$$

- Dyadic (two-input) operator is the linear blend operator

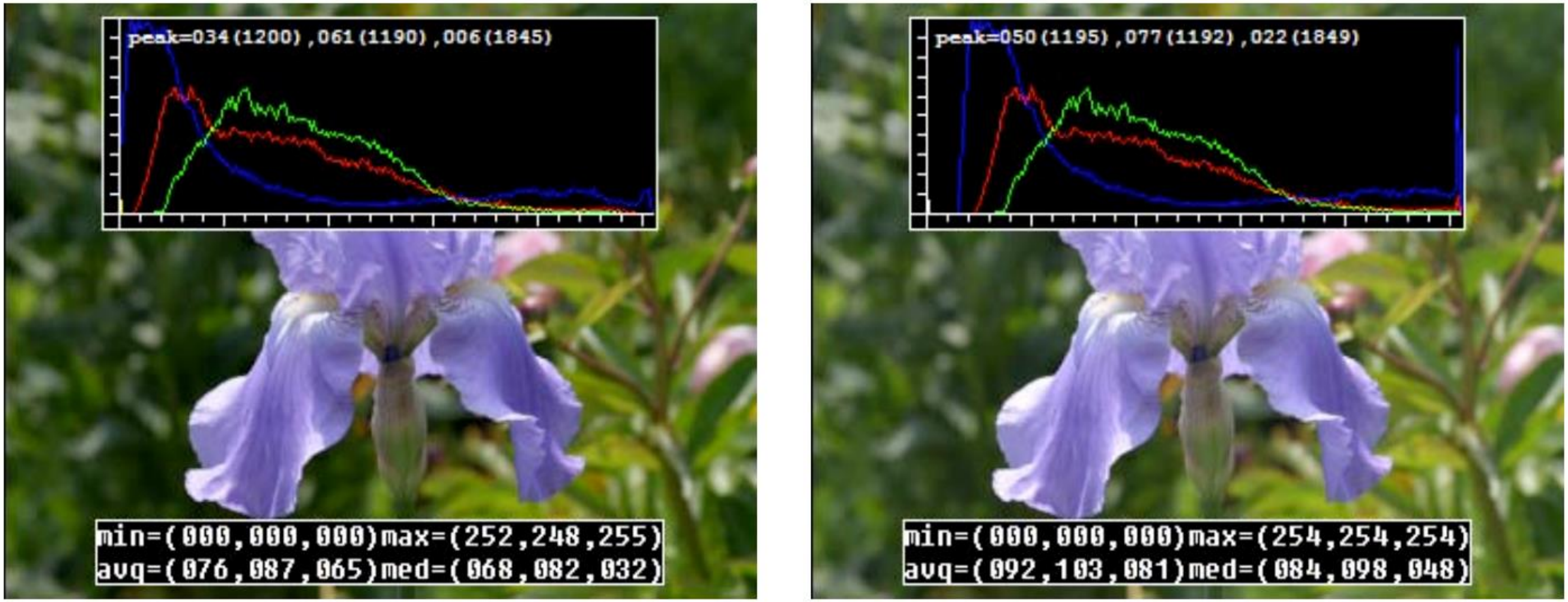
$$g(\boldsymbol{x}) = (1 - \alpha) f_0(\boldsymbol{x}) + \alpha f_1(\boldsymbol{x})$$

$$g(\boldsymbol{x}) = [f(\boldsymbol{x})]^{1/\gamma}$$

- Invert the gamma mapping applied by the sensor
-

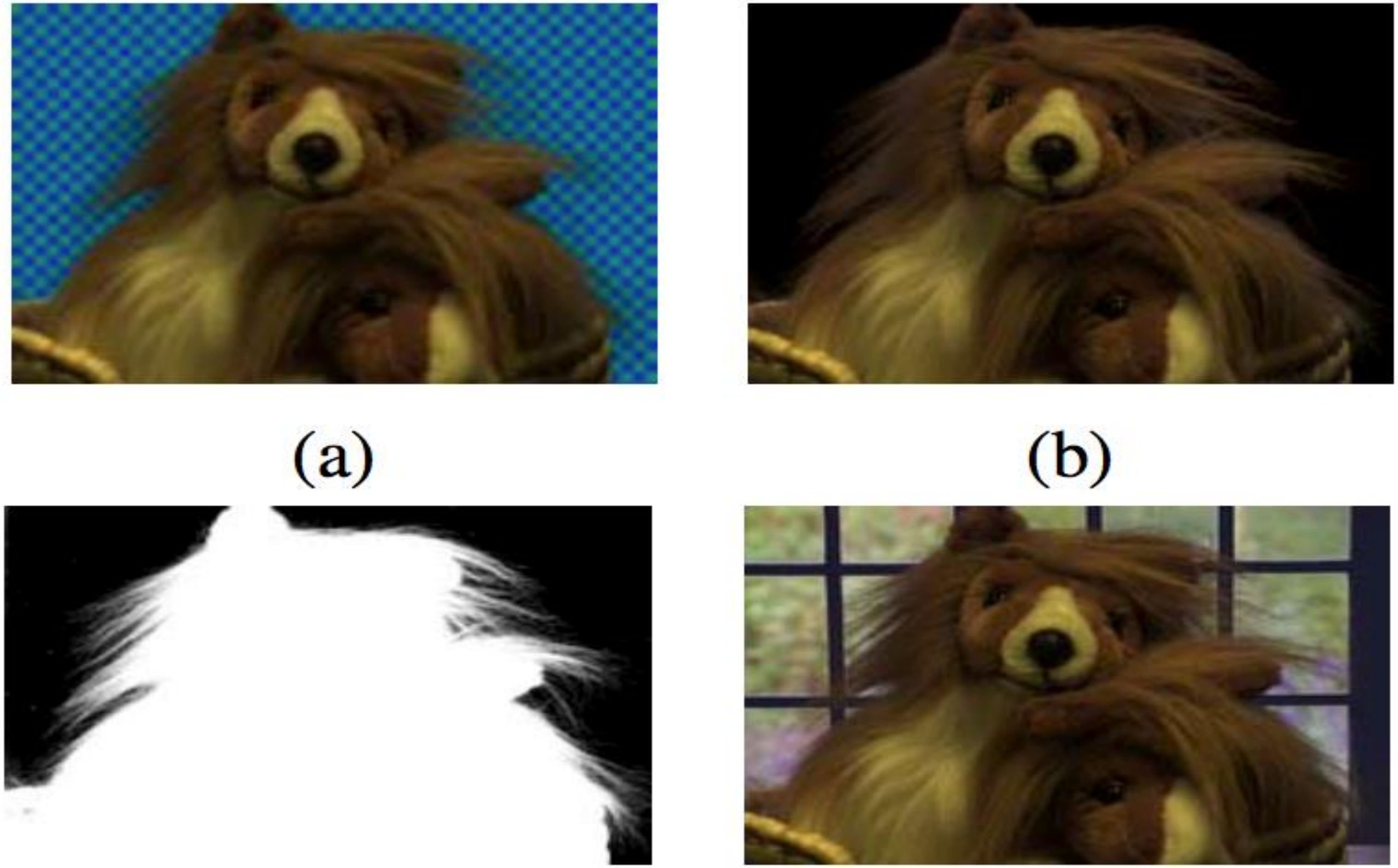
Color transforms

- brightness increased (additive offset, b = 16)



(a) (b)

Compositing and matting



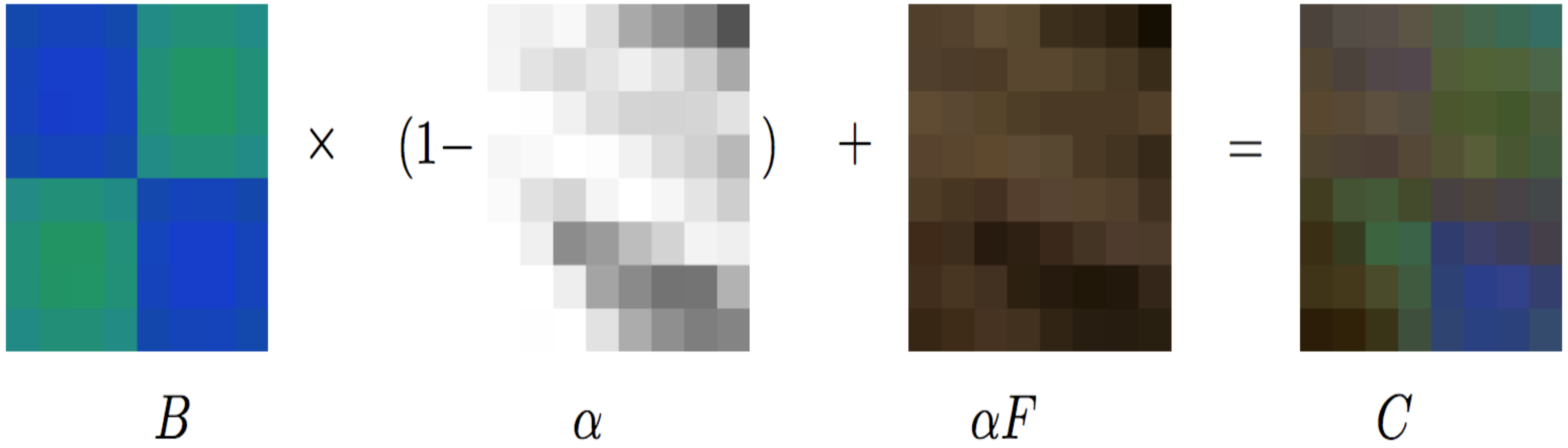
(a)

(b)

(c)

(d)

Compositing Equation



$$C = (1 - \alpha)B + \alpha F$$

Histogram equalization

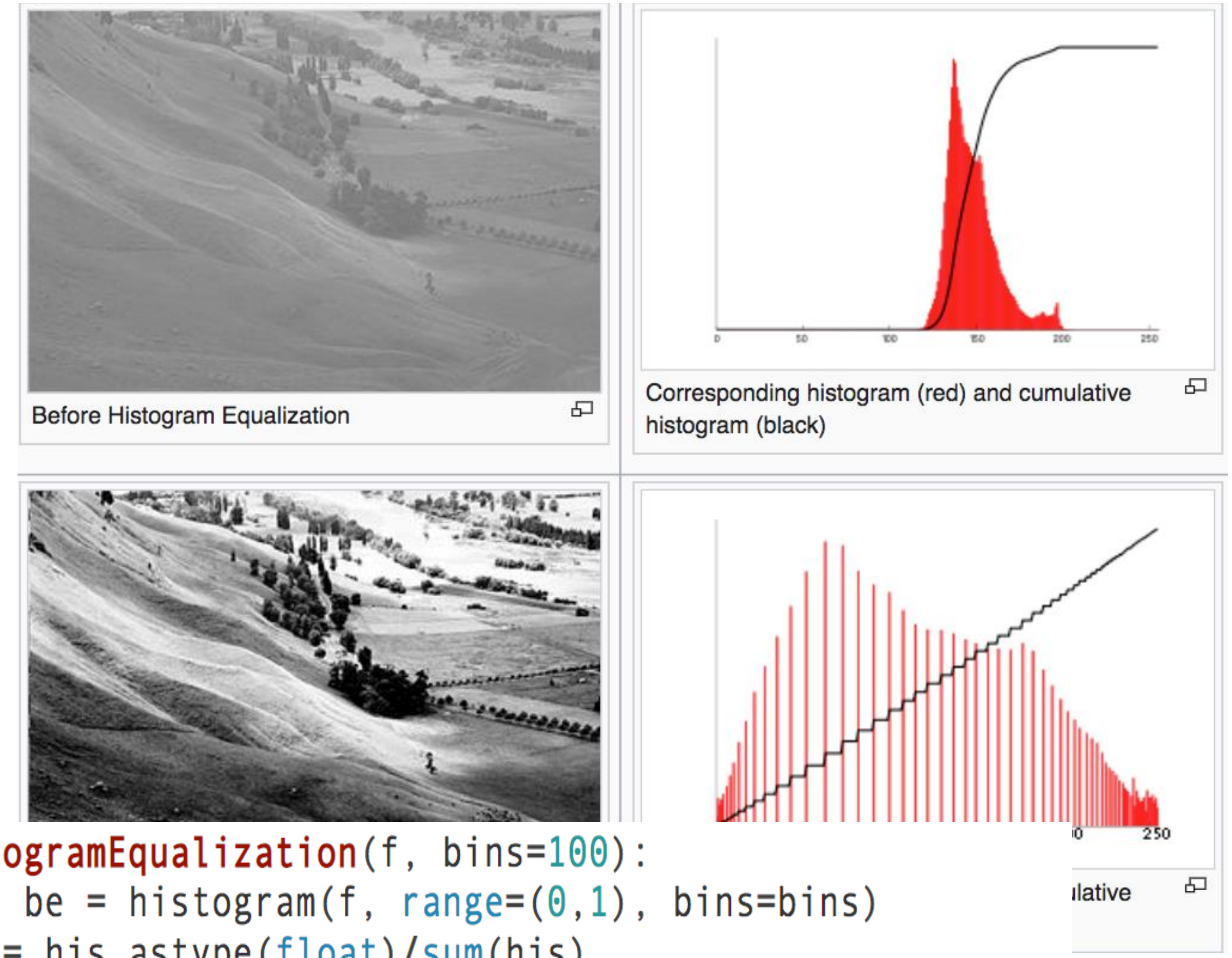
- Increases the global contrast of many image (especially when the image is represented by a narrow range of intensity values)

- Equalization is often used to normalize the gray values (or luminance in color images) to be invariant to changes in illumination conditions.

- The cumulative distribution c(I)

$$c(I) = \frac{1}{N}\sum_{i=0}^{I} h(i) = c(I-1) + \frac{1}{N}h(I)$$

Histogram equalization



Before Histogram Equalization

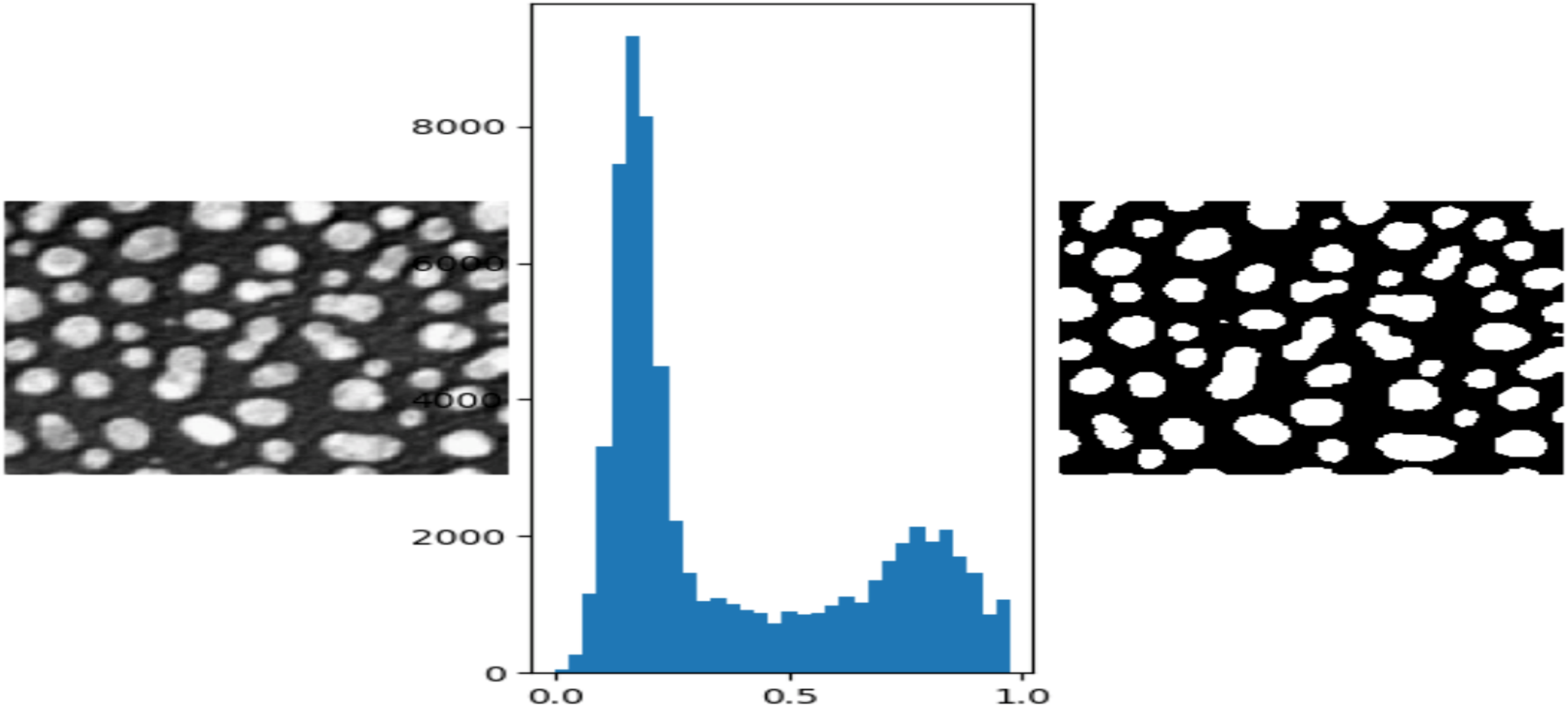
Corresponding histogram (red) and cumulative histogram (black)

ılative

```
def histogramEqualization(f, bins=100):
    his, be = histogram(f, range=(0,1), bins=bins)
    his = his.astype(float)/sum(his)
    return interp(f, be, hstack((zeros((1)), cumsum(his))))
```

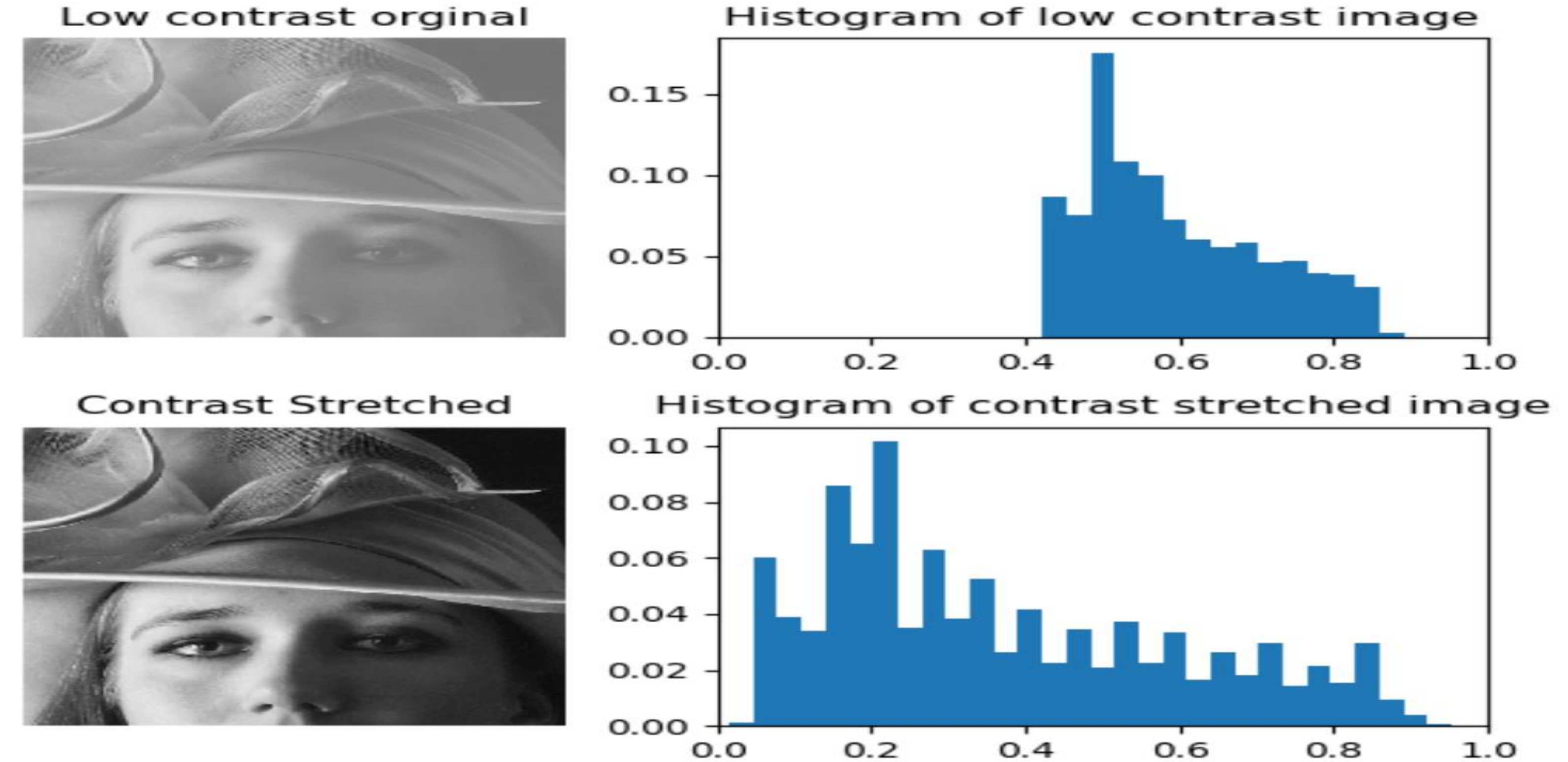
Thresholding

- Thresholding methods replace each pixel in an image with a black pixel if the image intensity I(i,j) is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.



Contrast stretching

- Contrast stretching (also called Normalization) attempts to improve an image by stretching the range of intensity values it contains to make full use of possible values



Summary

- Learn Image processing
- Image Arithmetic
- Understand Pixel transforms
- Color transforms
- Histogram based image operations