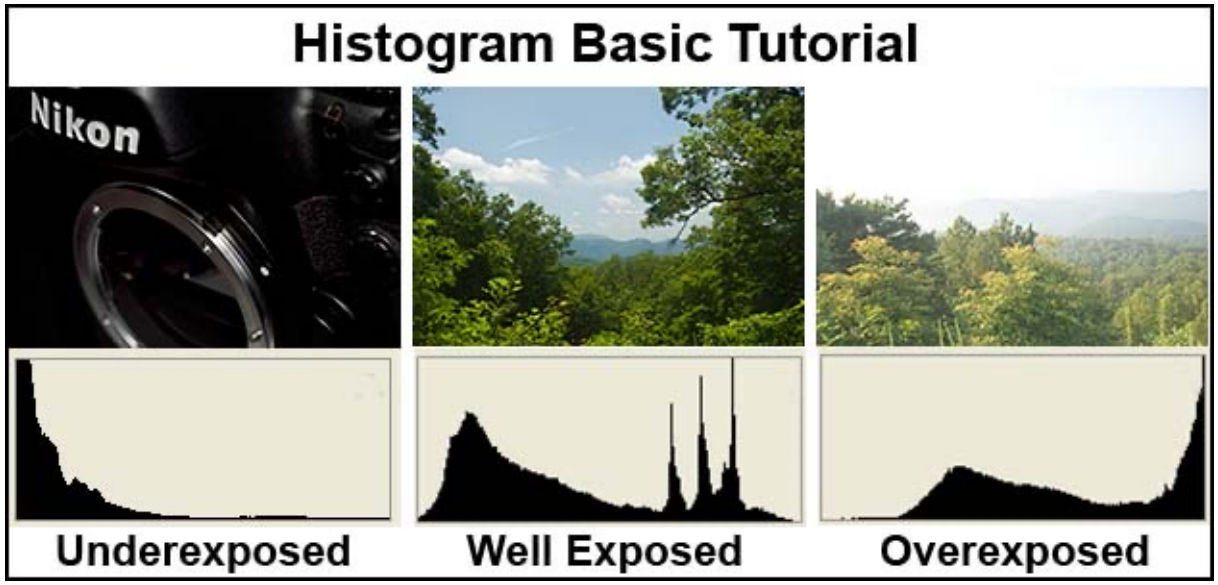
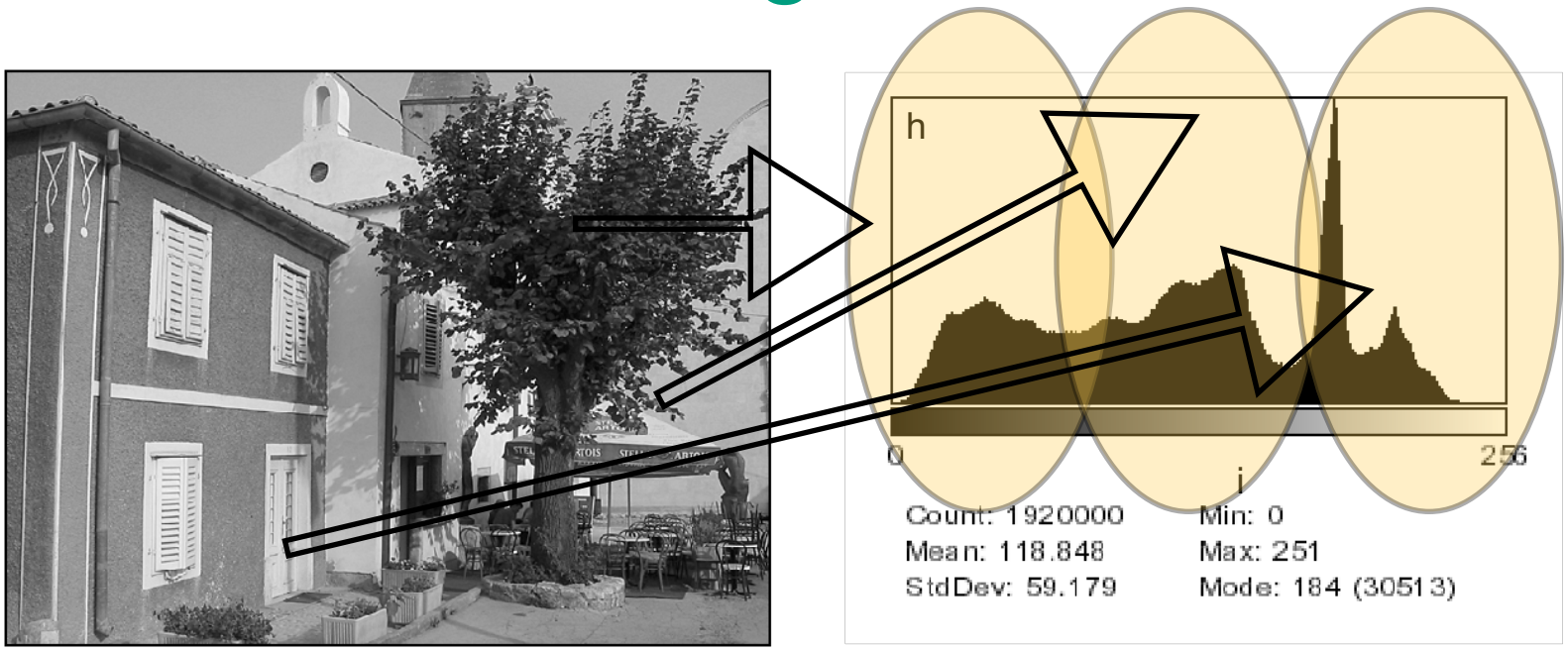
INF250

2: «Histograms and Image Statistics»

What is a histogram?



What is a histogram?



$$\mathsf{h}(i) = \text{ the } \textit{number} \text{ of pixels in } I \text{ with the intensity value } i$$

What is a histogram?

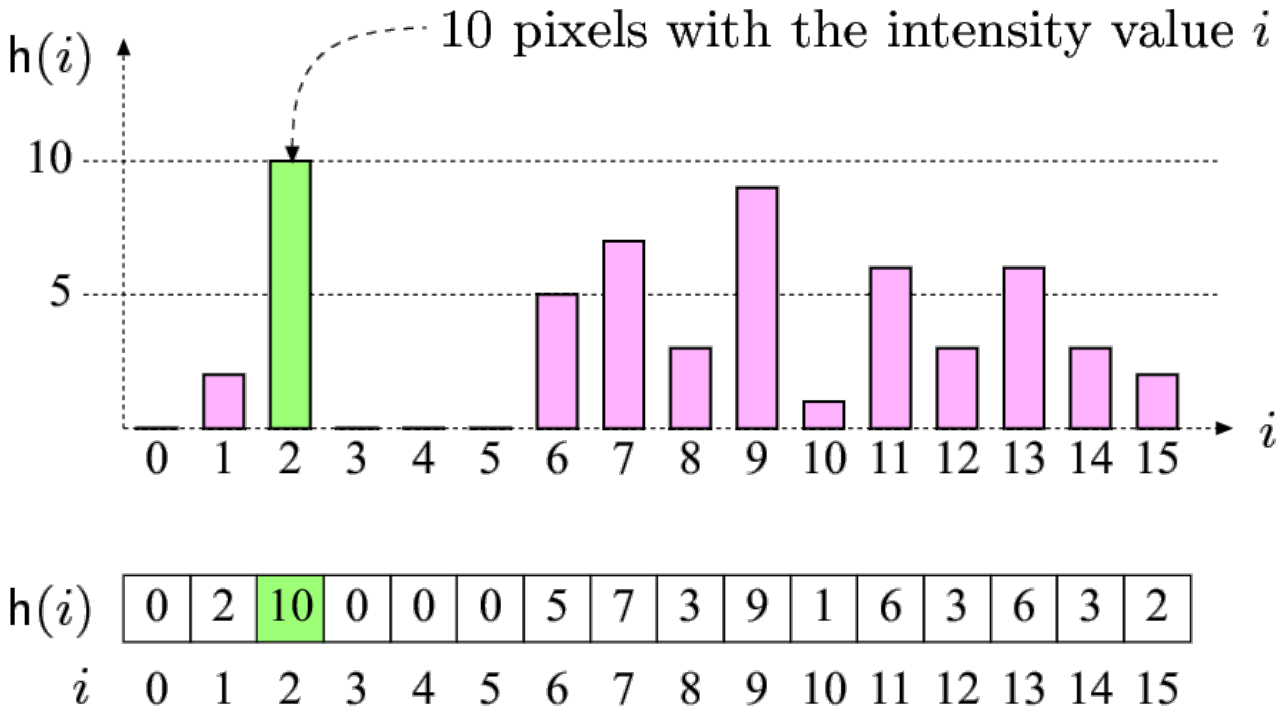
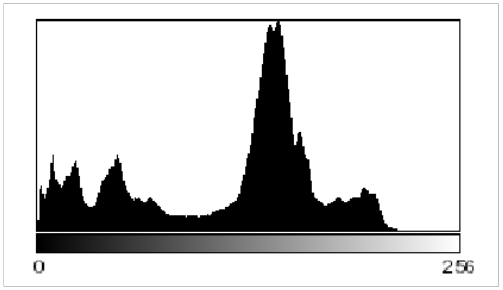
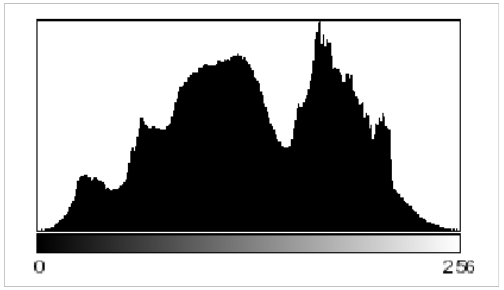


Figure 3.3 Histogram vector for an image with $K = 16$ possible intensity values. The indices of the vector element $i = 0 \ldots 15$ represent intensity values. The value of 10 at index 2 means that the image contains 10 pixels of intensity value 2.

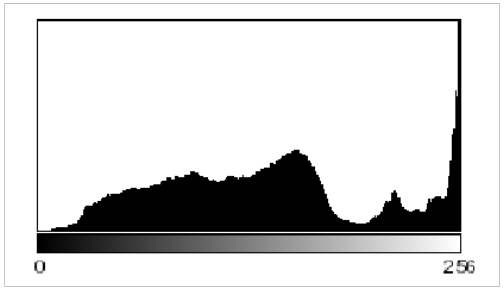
Correct exposure?



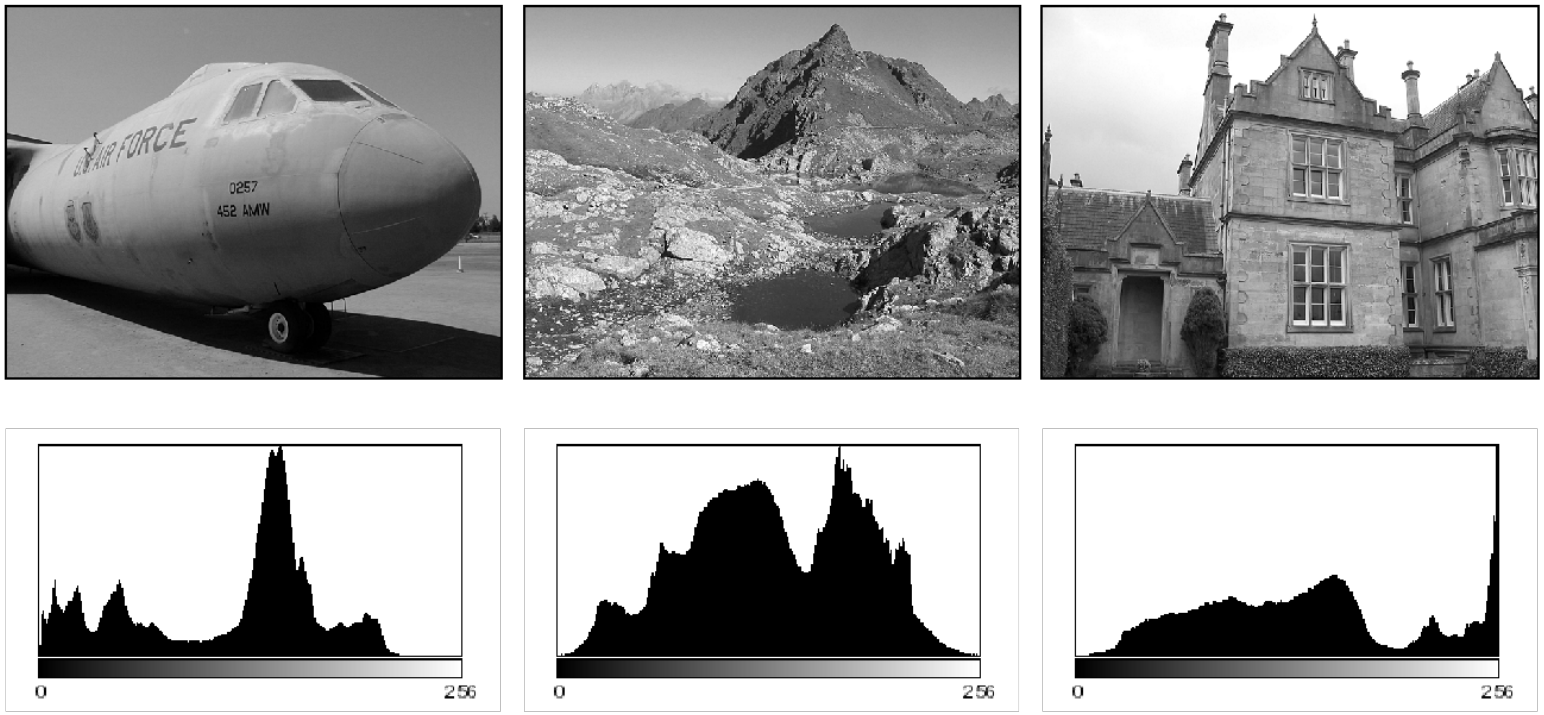
(a)



(b)



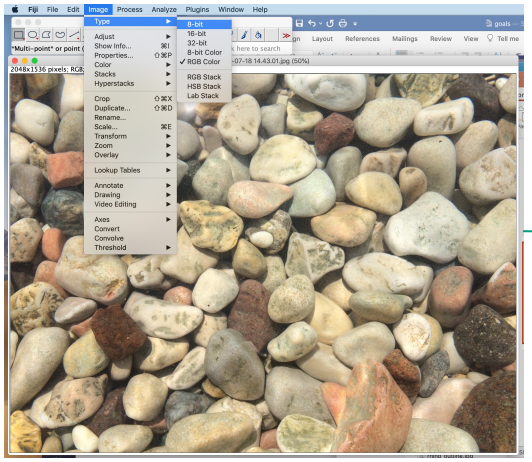
(c)

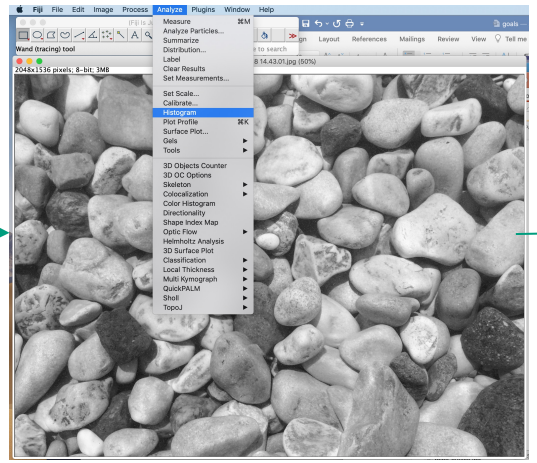


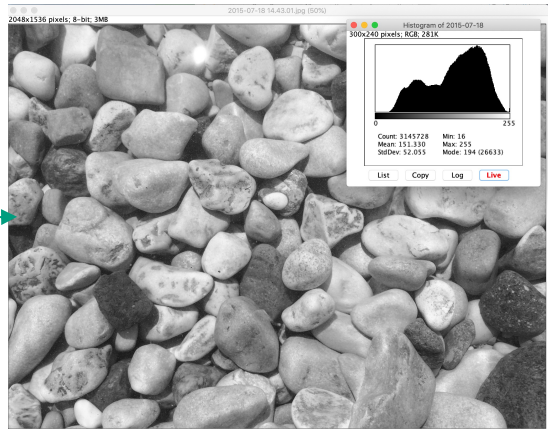
(a) (b) (c)

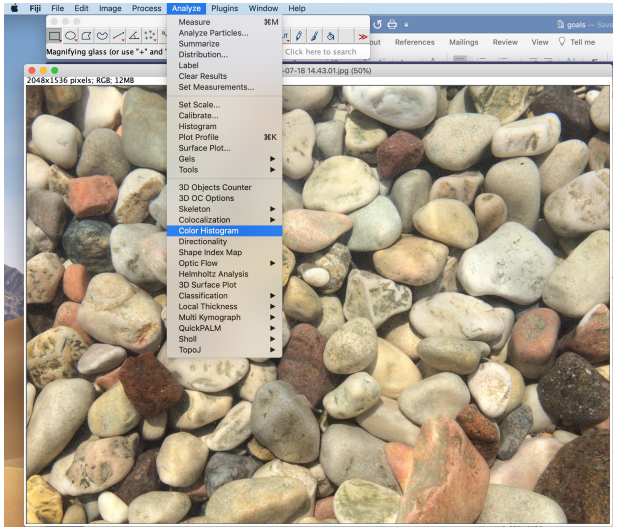
Under exposed correct Over exposed

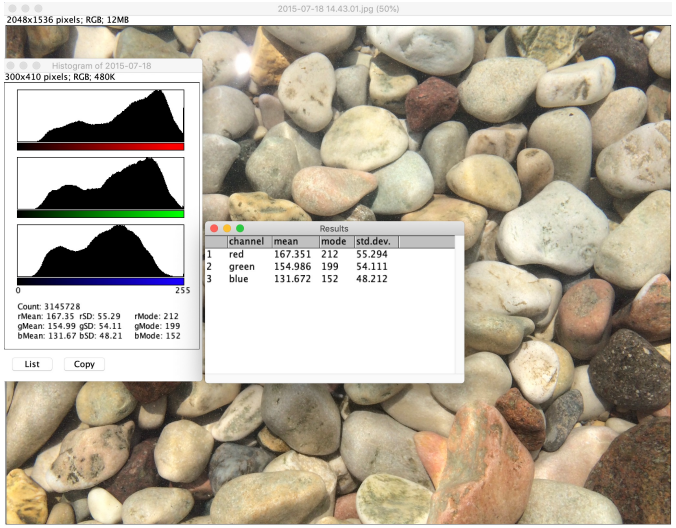
ImageJ/Fiji - Histograms





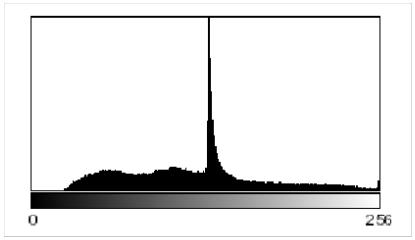






Histogram for colour image

(a)

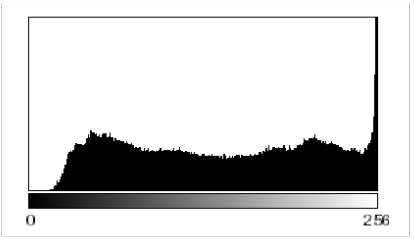


(b) h_{Lum}

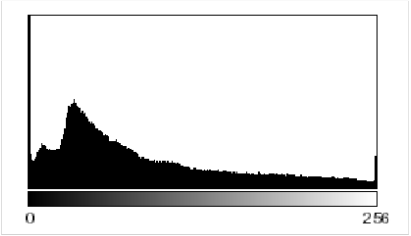
(c) R

(d) G

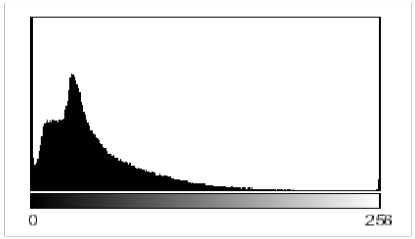
(e) B



(f) h_{R}



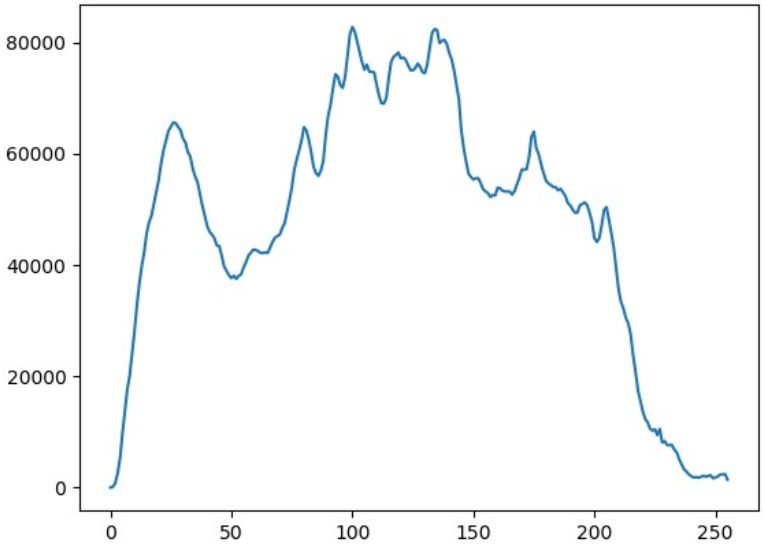
(g) h_{G}



(h) h_{B}

Python: Histogram 8-bit image

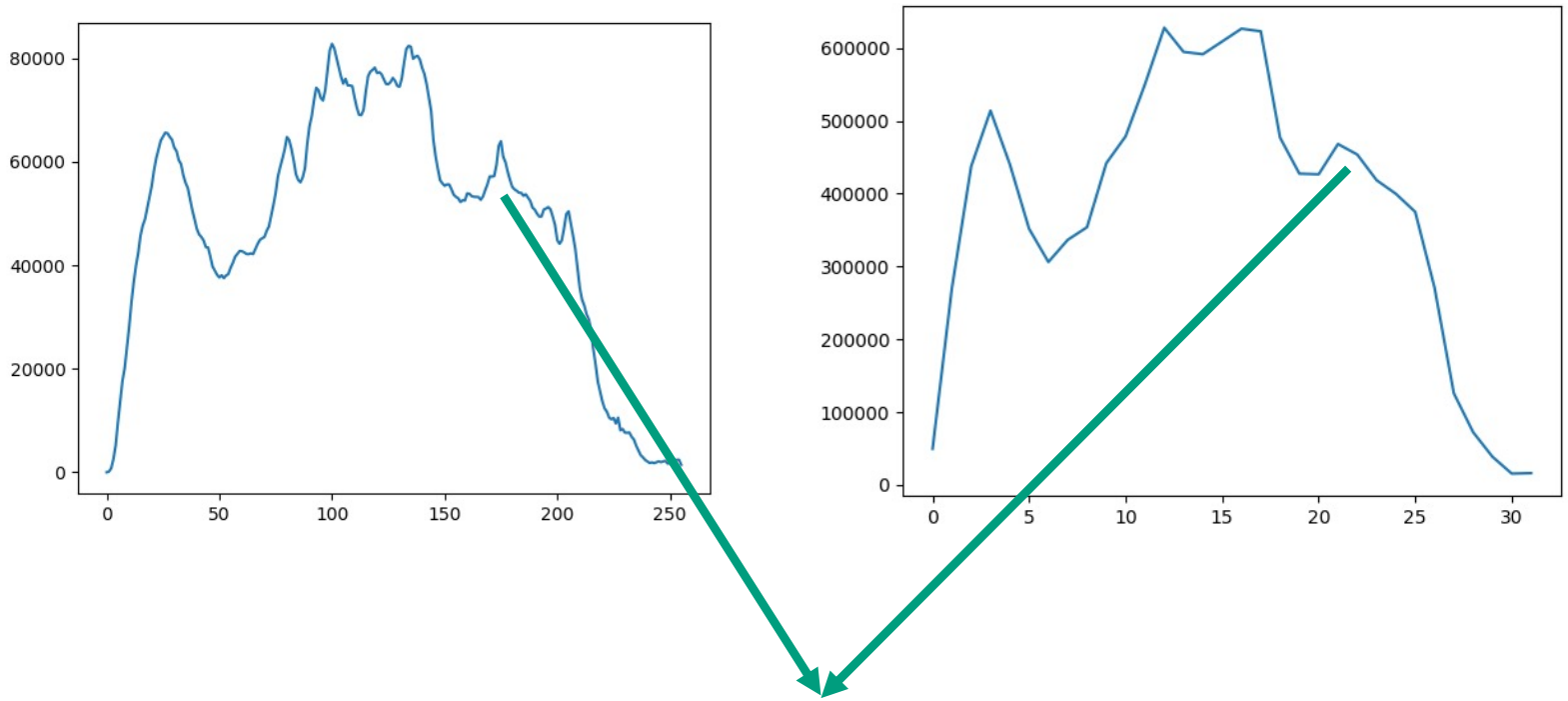
```
import matplotlib.pyplot as plt
import numpy as np

#reading image
filename = 'fall.tiff'
from skimage import io
fall = io.imread(filename)

# display image
plt.imshow(fall)

#mean av 3 RGB bilder
imagemean = fall.mean(axis=2)

# histogram
shape = np.shape(imagemean)
K = 256
M = shape[0]*shape[1]
#M = shape[0]
histogram = np.zeros(256)
for i in range(shape[0]):
    for j in range(shape[1]):
        pixval = int(imagemean[i,j])
        histogram[pixval] += 1
plt.plot(histogram)
```


80000
60000
40000
20000
0
0
50
100
150
200
250

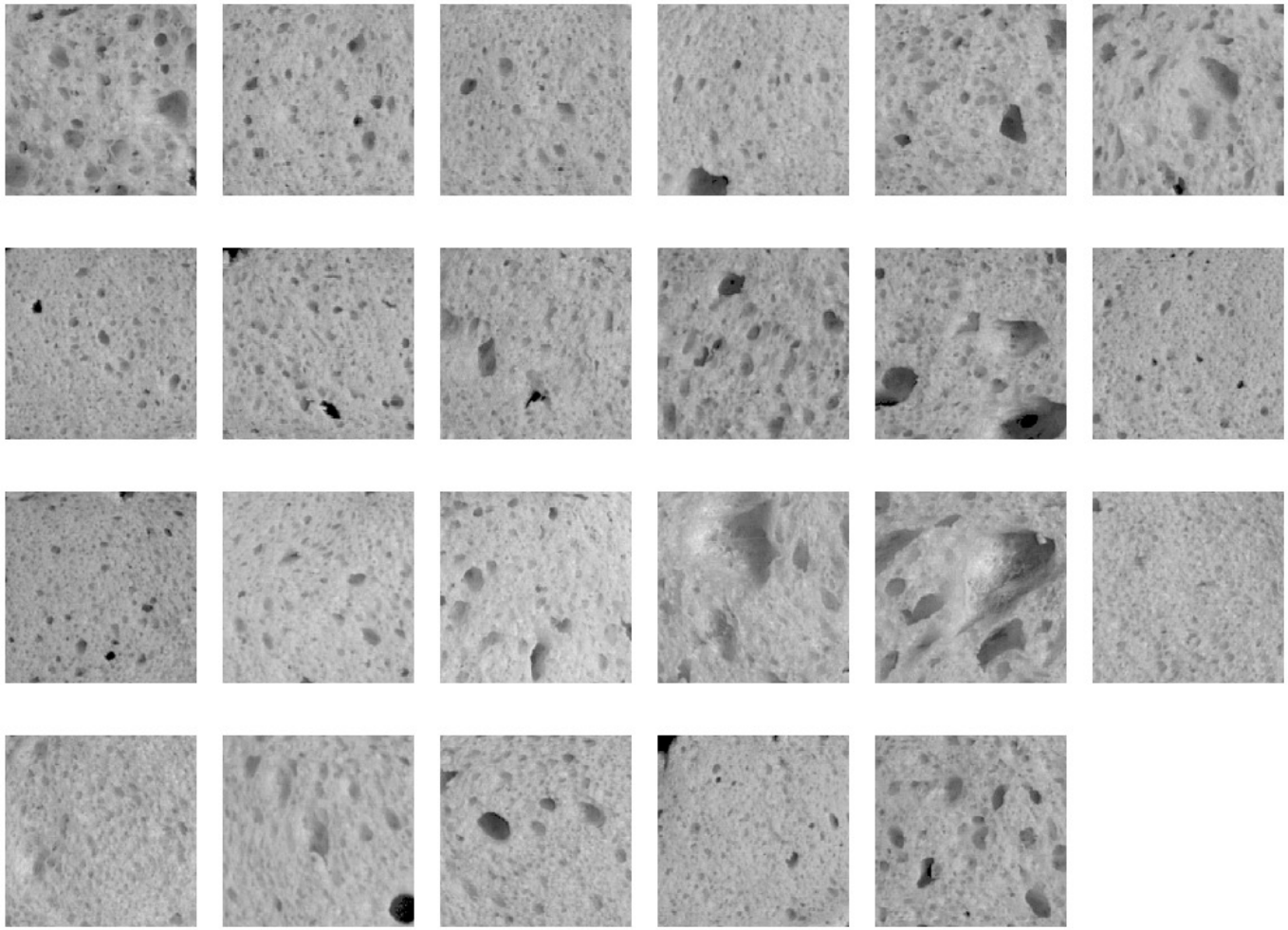
Histogram-binning, Ex 32 bins

```
# Binned histogram
K = 256
b = 32
histogram = np.zeros(b)
shape = np.shape(imagemean)
for i in range(shape[0]):
    for j in range(shape[1]):
        pixval = int(imagemean[i,j])
        histval = int(pixval*(b/K))
        histogram[histval] += 1

plt.figure()
plt.plot(histogram)
```

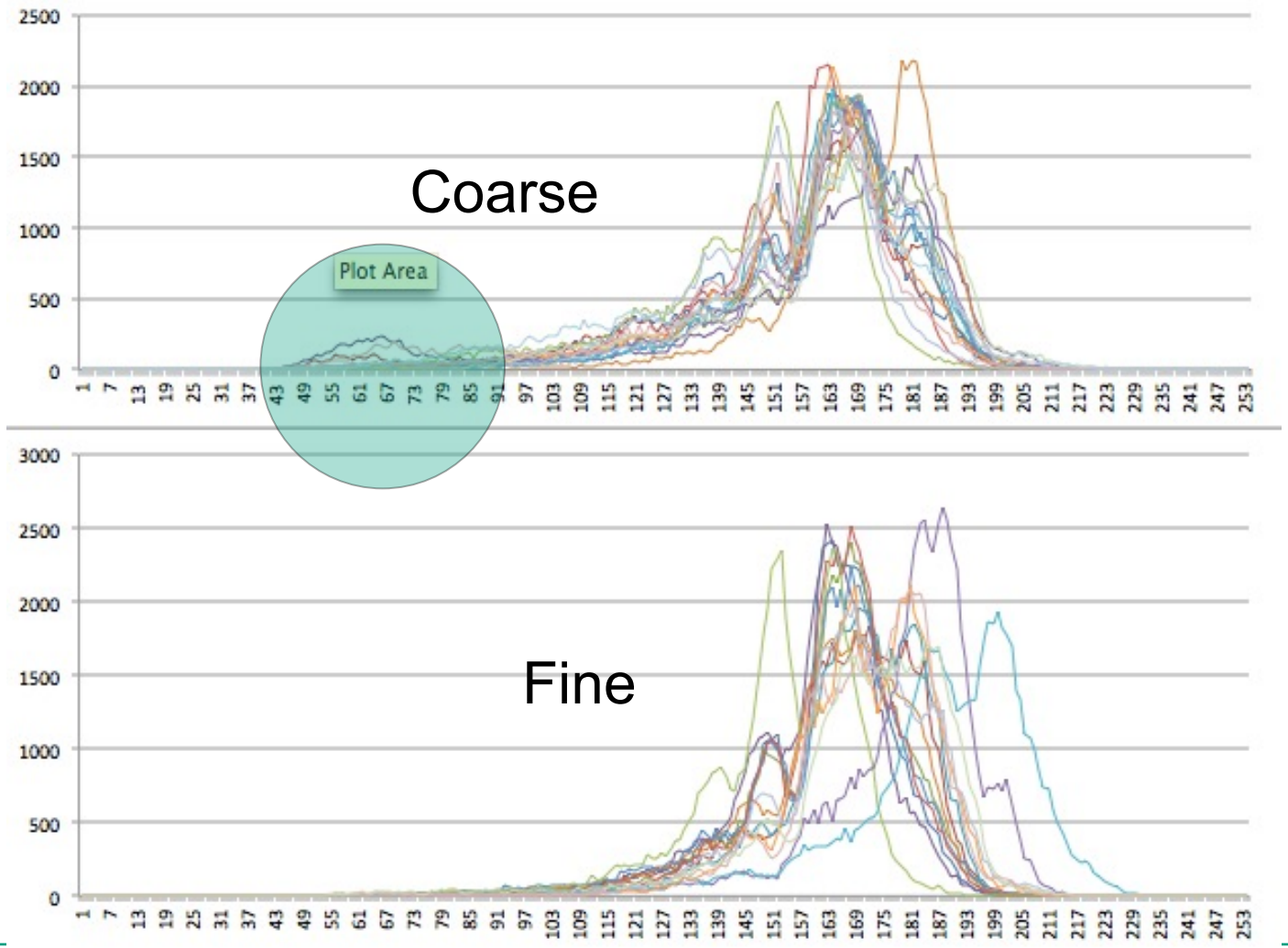
Comparison no-bin/bin



Example of histogram texture analysis



Baguette textures and histograms



Statistics on the histogram

- Often used to describe and image
  - Max/min
  - Standard deviation
  - Variance

    - Kurtosis
    - Skewness

http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm

First order statistics (functions of pixel values)

- Mean

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i$$

- Variance

$$\sigma^2 = \frac{1}{N-1}\sum_{i=0}^{N-1}(x_i - \mu)^2$$

- Coefficient of variation

$$cv = \frac{\sigma}{\mu}$$

- Skewness

$$\gamma_1 = \frac{1}{(N-1)\sigma^3}\sum_{i=0}^{N-1}(x_i - \mu)^3$$

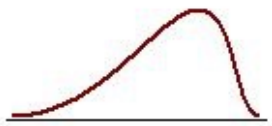
- Kurtosis

$$\gamma_2 = \frac{1}{(N-1)\sigma^4}\sum_{i=0}^{N-1}(x_i - \mu)^3 - 3$$

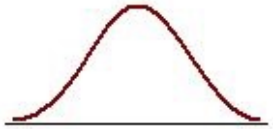
Skewness / Kurtosis

Skewness

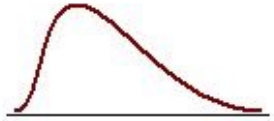
The coefficient of Skewness is a measure for the degree of symmetry in the variable distribution.



Negatively skewed distribution
or Skewed to the left
Skewness <0



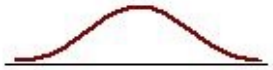
Normal distribution
Symmetrical
Skewness = 0



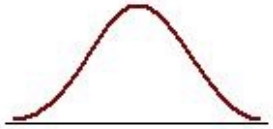
Positively skewed distribution
or Skewed to the right
Skewness > 0

Kurtosis

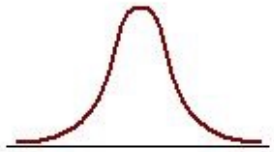
The coefficient of Kurtosis is a measure for the degree of peakedness/flatness in the variable distribution.



Platykurtic distribution
Low degree of peakedness
Kurtosis <0



Normal distribution
Mesokurtic distribution
Kurtosis = 0



Leptokurtic distribution
High degree of peakedness
Kurtosis > 0

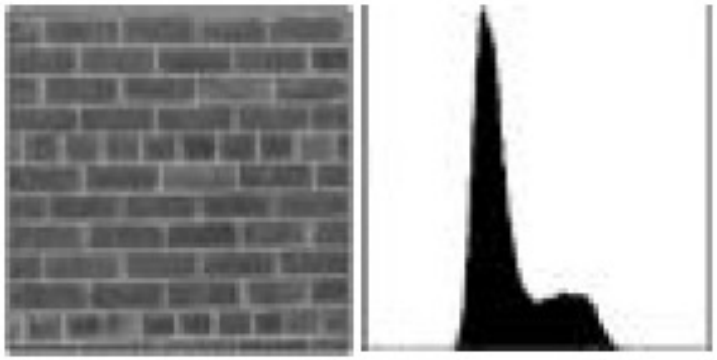
http://www.janzengroup.net/stats/lessons/descriptive.html

Skewness / Kurtosis

- Skewness is a measure of symmetry (or lack of symmetry)
- Kurtosis is a measure of flatness/peakness of the histogram
- Used as variables in texture analysis

Example of first order statistics

Mean	167.9
Variance	669.0
CV	0.15
Skewness	-0.82
Curtosis	0.01

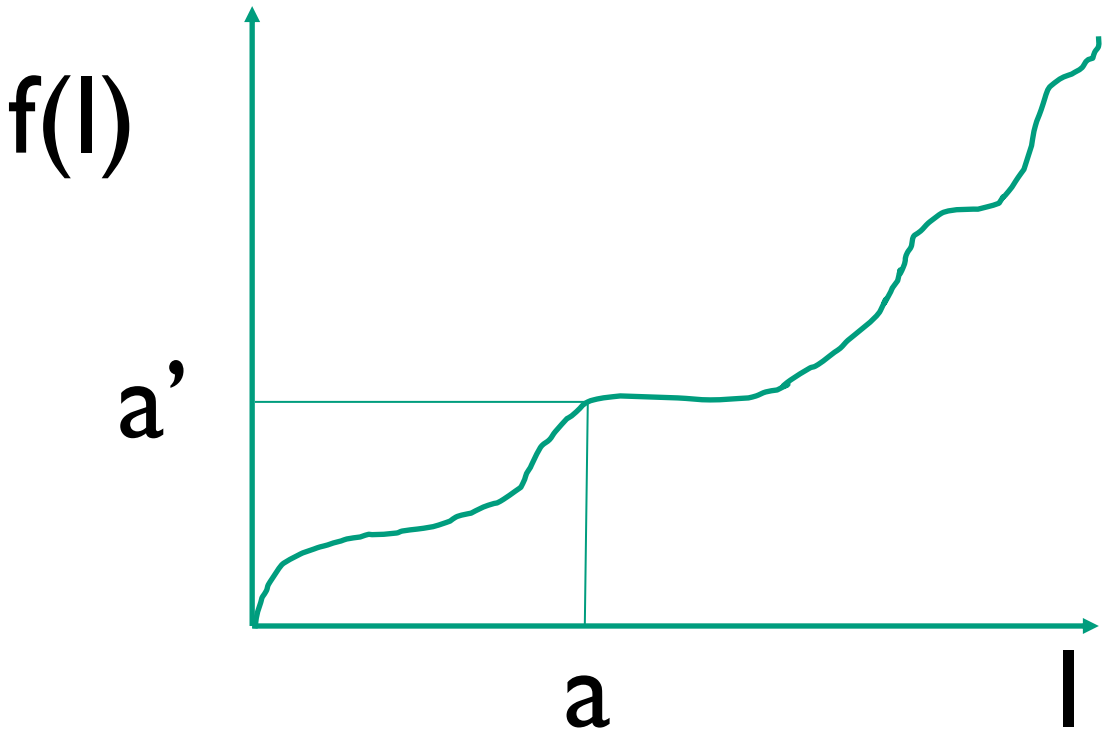


Mean	105.1
Variance	720
CV	0.26
Skewness	1.15
Curtosis	0.30

Point operations
Modify with a function

- Pixel values can be modified by a ”transfer function”
  - New pixel value from a function output
  - Brightness/Contrast
  - Random transformations (curves)
  - Quantification of images (posterizing)
  - Global tresholding
  - Gamma corrections
  - Colour transformations

Transfer-function



a'=f(a)

Modification of intensity

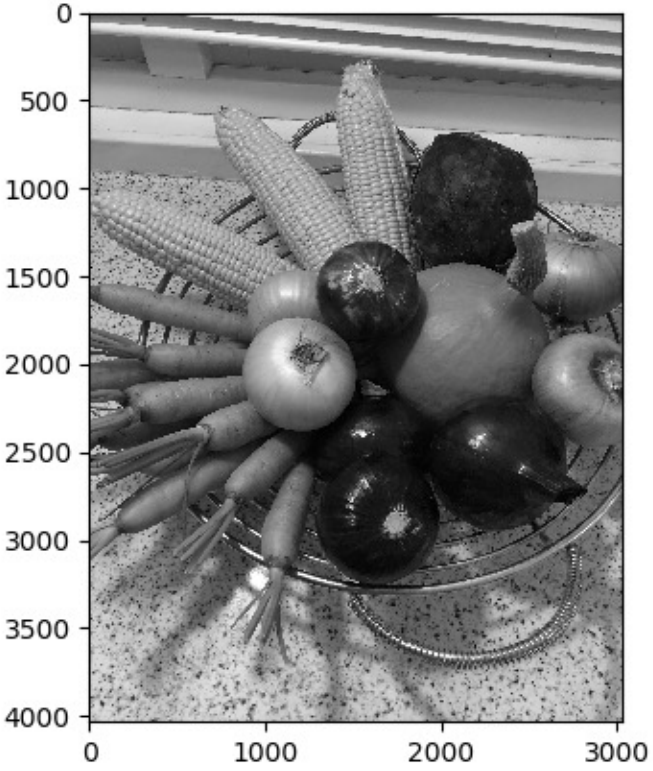
- Contrast
  - Increase contrast med 50%: f(a) = a*1.5
- Brightness
  - Increase brightness with 10 units: f(a) = a + 10

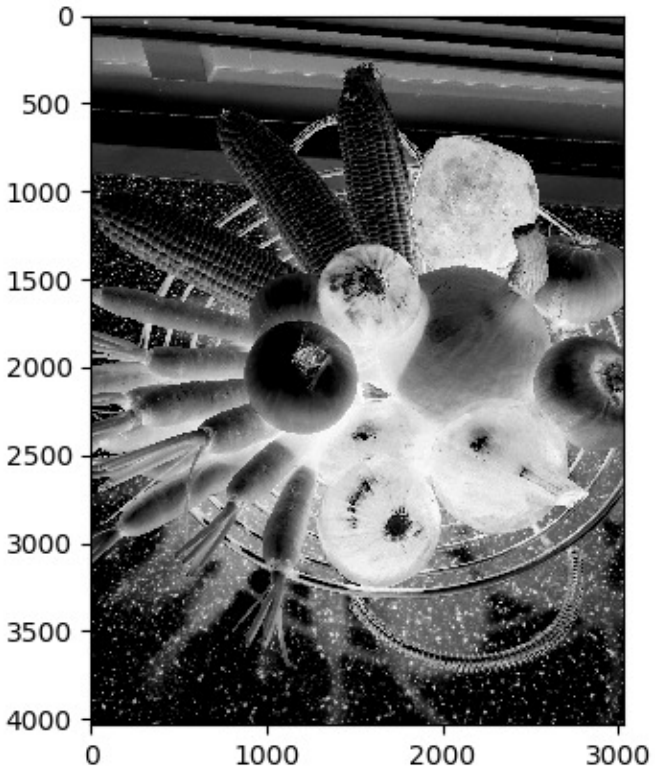
Program: Point operation

```
imagemean = lift.mean(axis=2)

shape = np.shape(imagemean)
for i in range(shape[0]):
    for j in range(shape[1]):
        imagemean[i,j] = int((imagemean[i,j])*1.5 + 0.5)
        if imagemean[i,j] > 255:
            imagemean[i,j] = 255
```

Increasing contrast with 50%, we add 0.5 to round up to an integer value

Inversion


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000

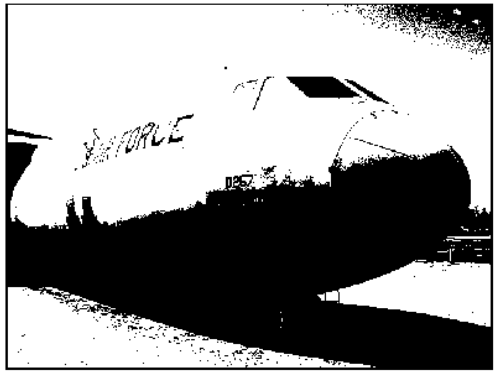
Thresholding

$$f_{threshold}(a) = \begin{cases} a_0 & \text{for} \quad a < a_{th} \\ a_1 & \text{for} \quad a \geq a_{th} \end{cases}$$

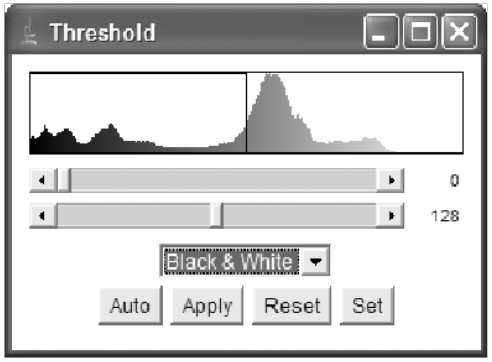
- Thresholding separates the the pixel values in two classes.
- A common application is binarizing an intensity image with pixel values values 1 and 0.

Thresholding

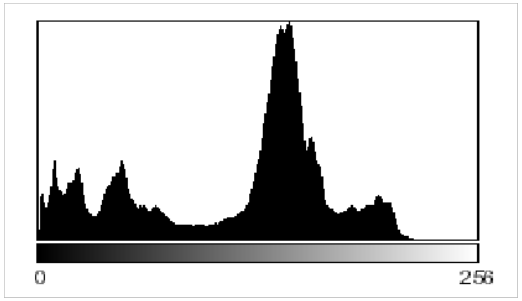
(a)



(b)



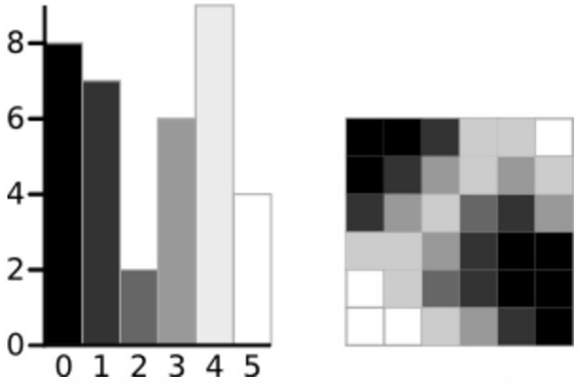
(e)



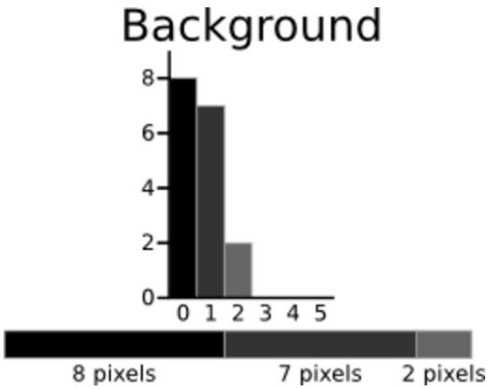
(c)

(d)

Otsu thresholding



A 6-level greyscale image and its histogram



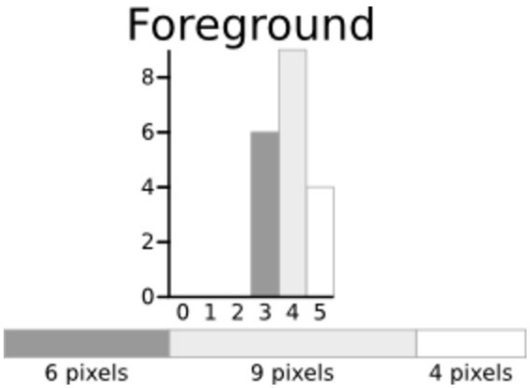
$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}$$

$$= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}$$

$$= 0.4637$$



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

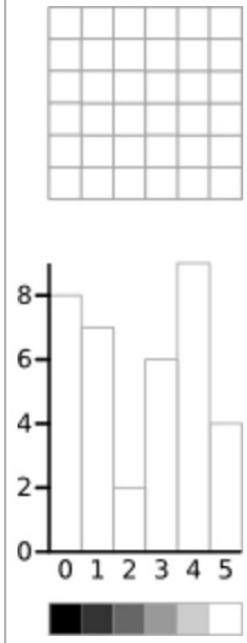
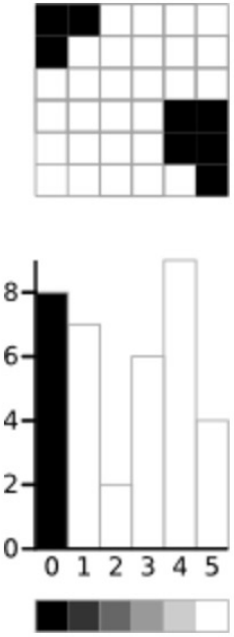
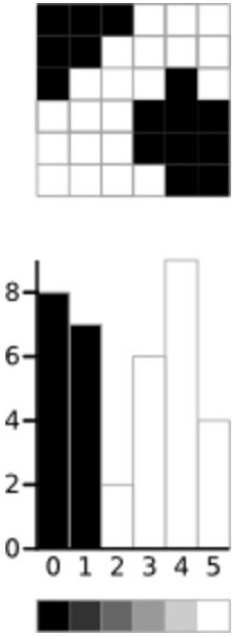
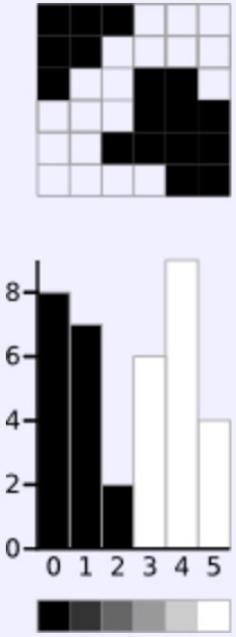
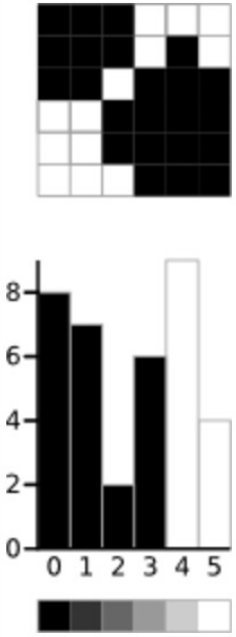
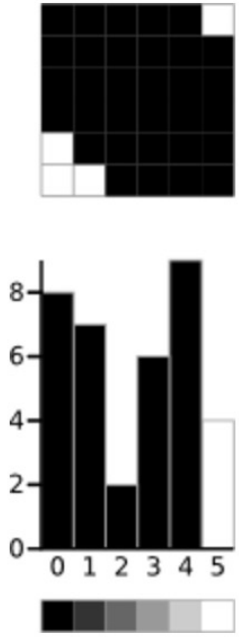
$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

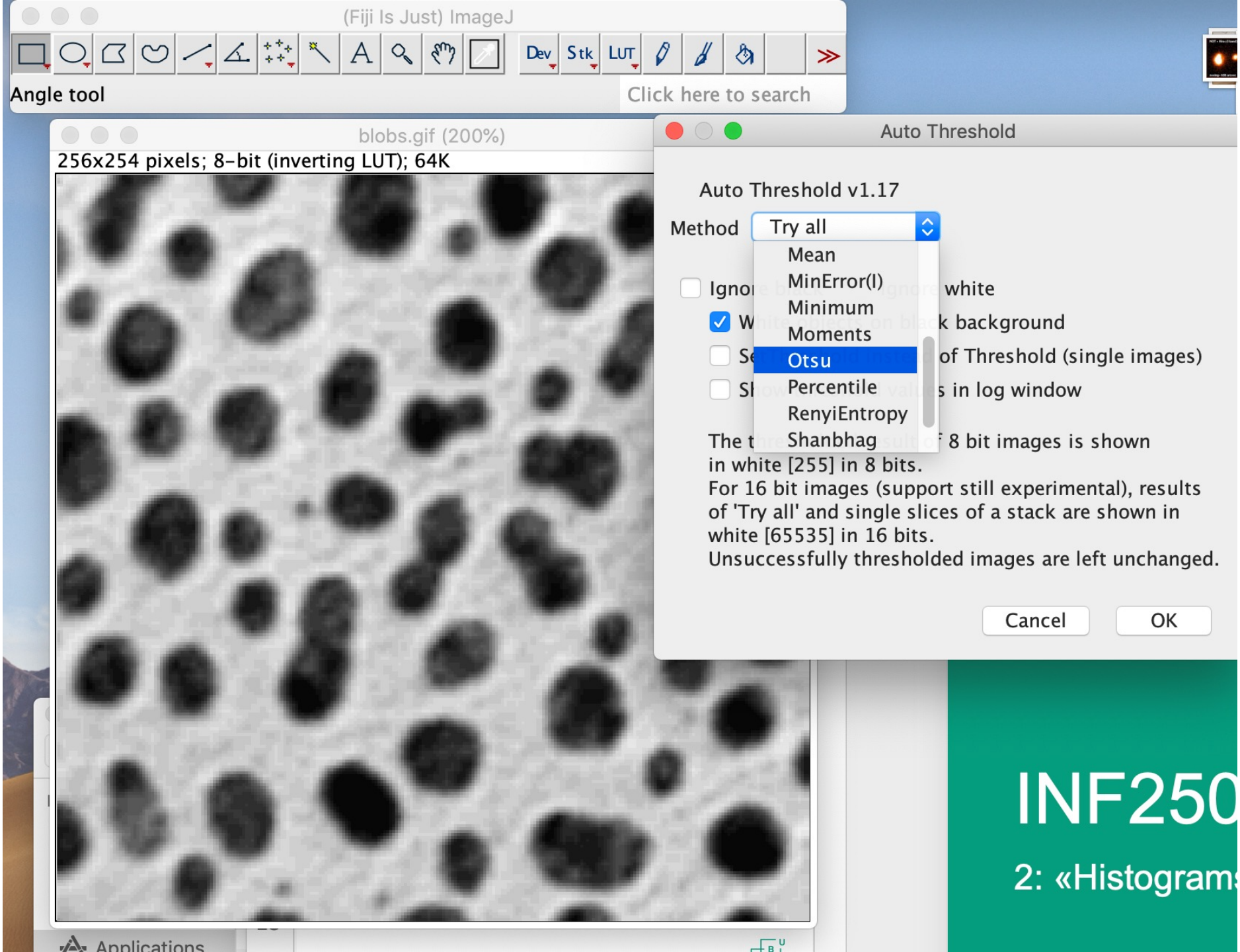
$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

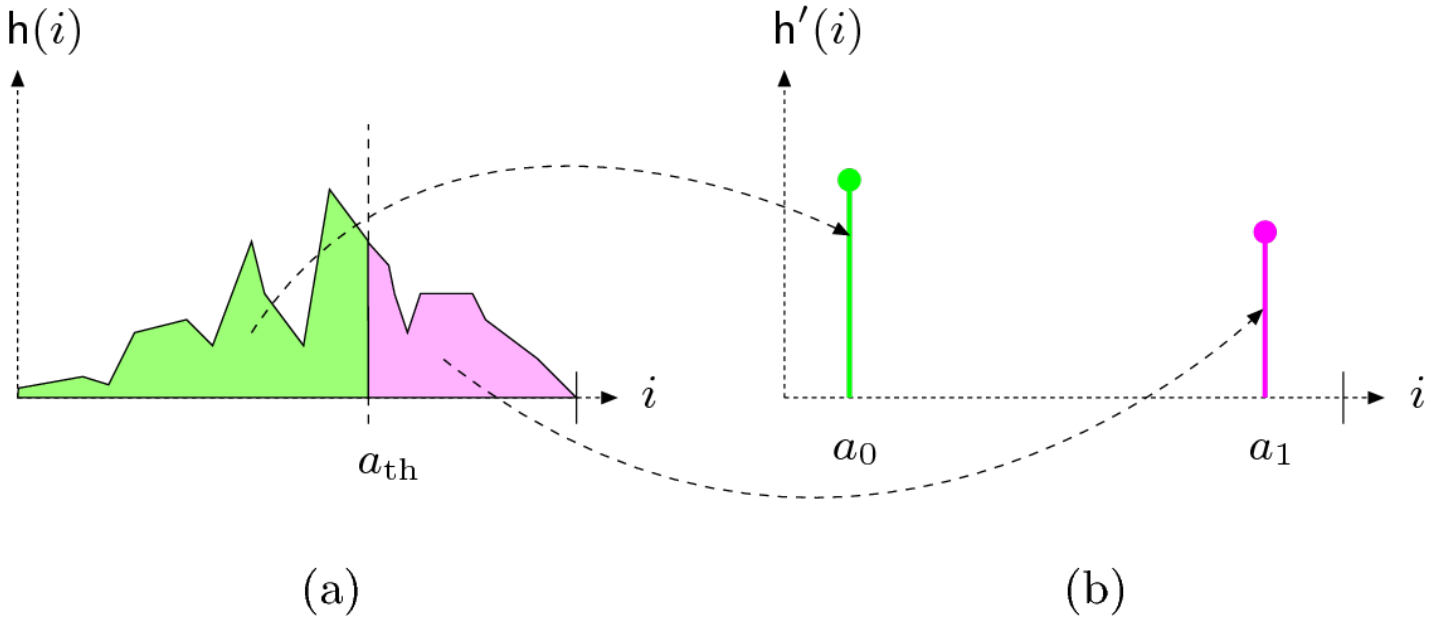
$$= 0.5152$$

http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

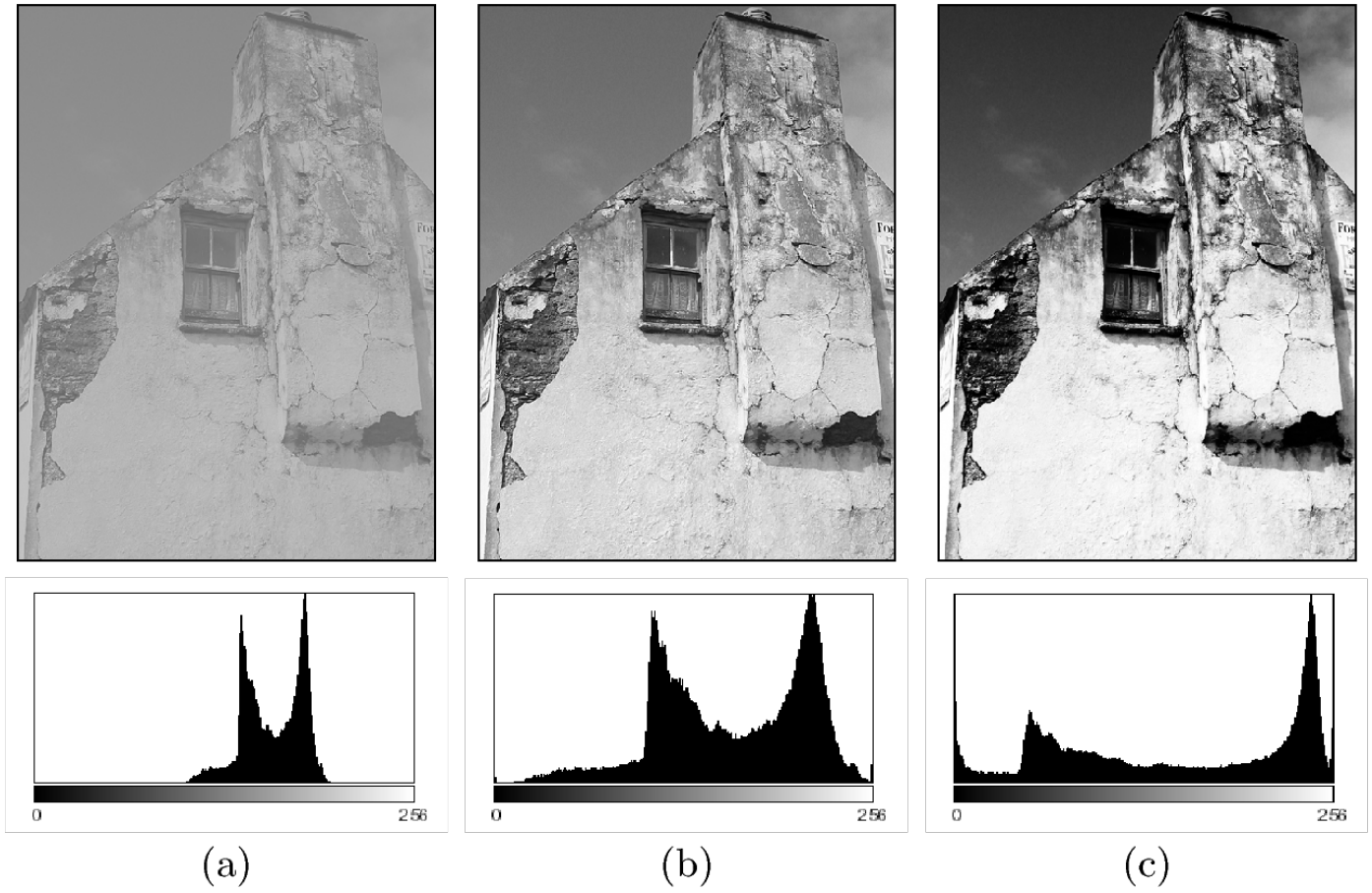
Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\mathbf{\sigma^2_W = 0.4909}$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$


(Fiji Is Just) ImageJ
Dev
Stk
LUT
Angle tool
Click here to search
blobs.gif (200%)
256x254 pixels; 8-bit (inverting LUT); 64K
Auto Threshold
Auto Threshold v1.17
Method
Try all
Mean
MinError(I)
Minimum
Moments
Otsu
Percentile
RenyiEntropy
Shanbhag
white
background
of Threshold (single images)
in log window
8 bit images is shown
in white [255] in 8 bits.
For 16 bit images (support still experimental), results
of 'Try all' and single slices of a stack are shown in
white [65535] in 16 bits.
Unsuccessfully thresholded images are left unchanged.
Cancel
OK
INF250
2: «Histograms
Applications

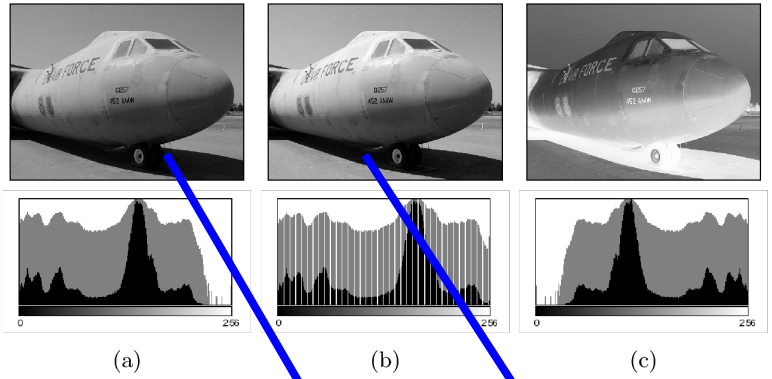
Histogram after thresholding



Contrast

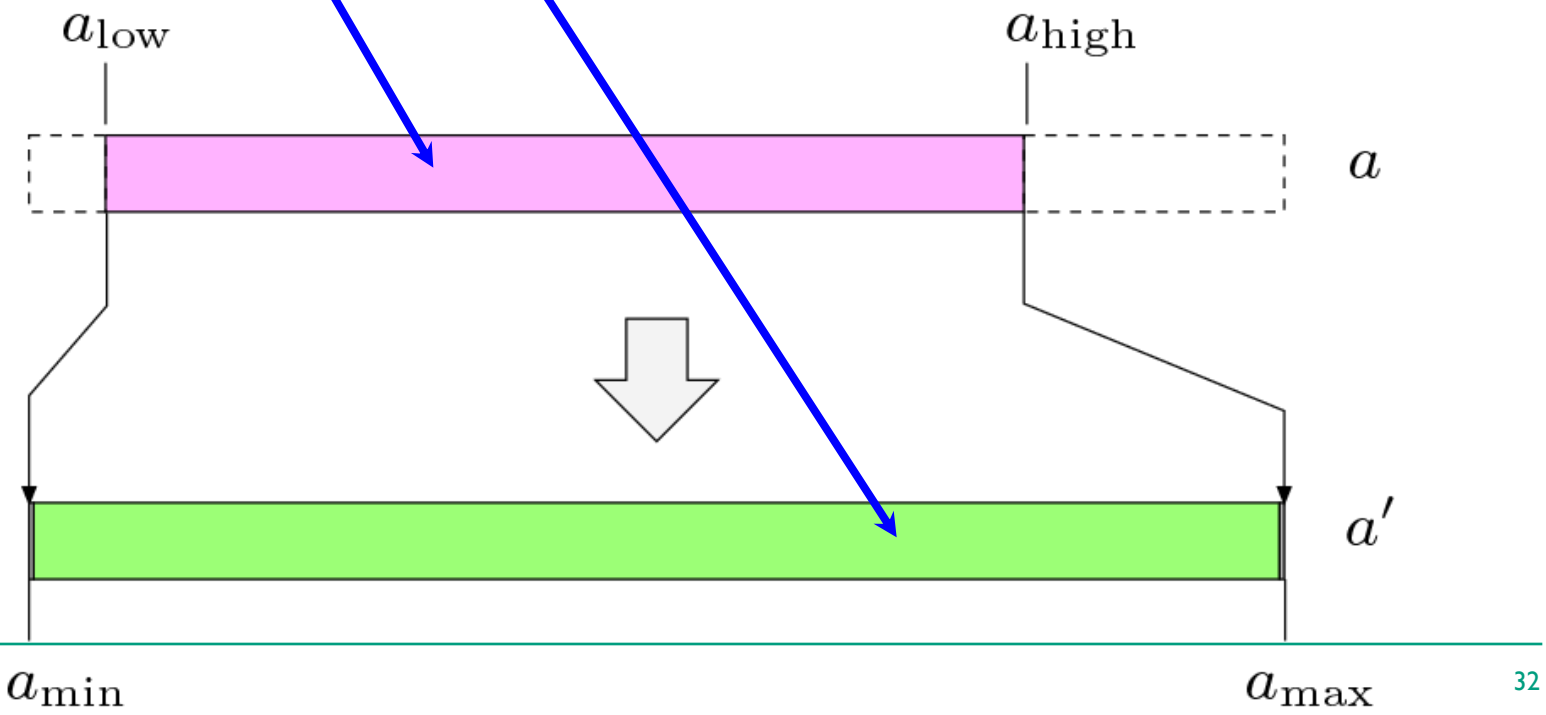


Automatic contrast adjustment



$$f_{\mathrm{ac}}(a) = a_{\min} + (a - a_{\mathrm{low}}) \cdot \frac{a_{\max} - a_{\min}}{a_{\mathrm{high}} - a_{\mathrm{low}}}$$

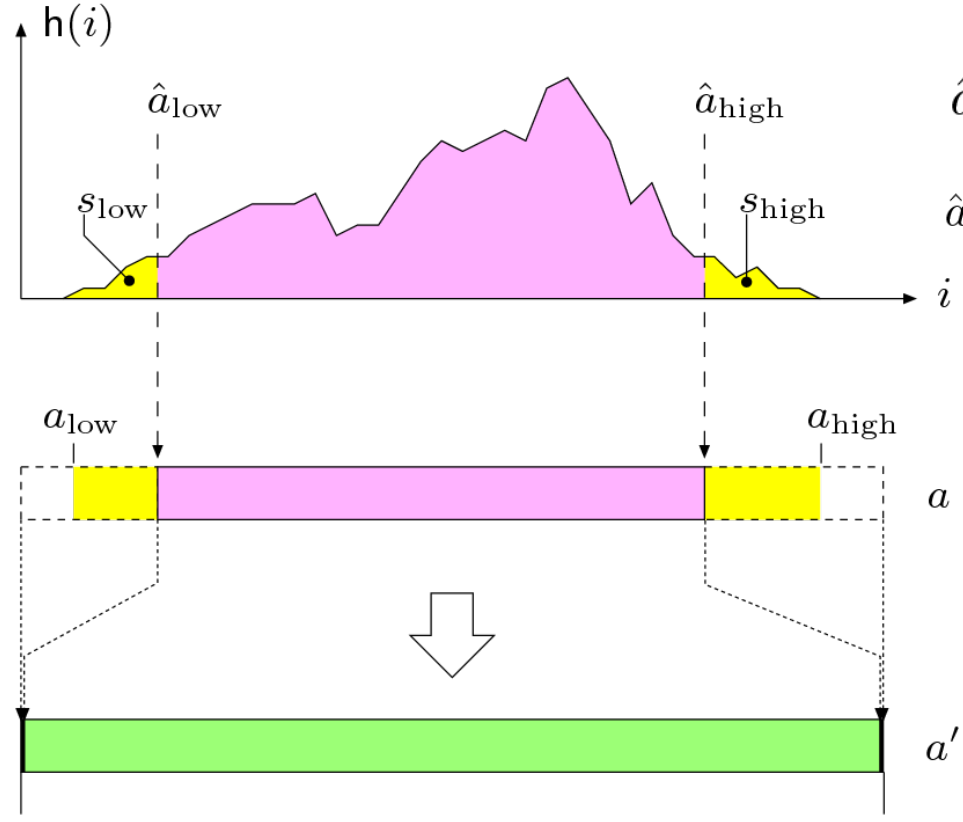
$$f_{\mathrm{ac}}(a) = (a - a_{\mathrm{low}}) \cdot \frac{255}{a_{\mathrm{high}} - a_{\mathrm{low}}}$$



Automatic contrast adjustment

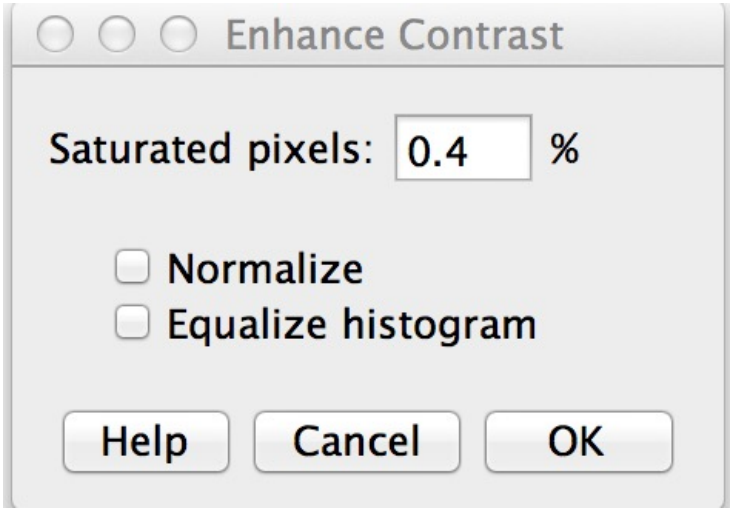
```
shape = np.shape(ireland)
ima = ireland
amin = 0
amax = 255
alow = int(min(ima.flatten()))
ahigh = int(max(ima.flatten()))
for i in range(shape[0]):
    for j in range(shape[1]):
        ima[i,j] = amin+(ima[i,j]-alow)*(amax-amin)/(ahigh-alow)

plt.imshow(ima, 'gray')
plt.hist(ima.ravel(),256,[0,256])
plt.show()
```

Modified automatic contrast adjustment



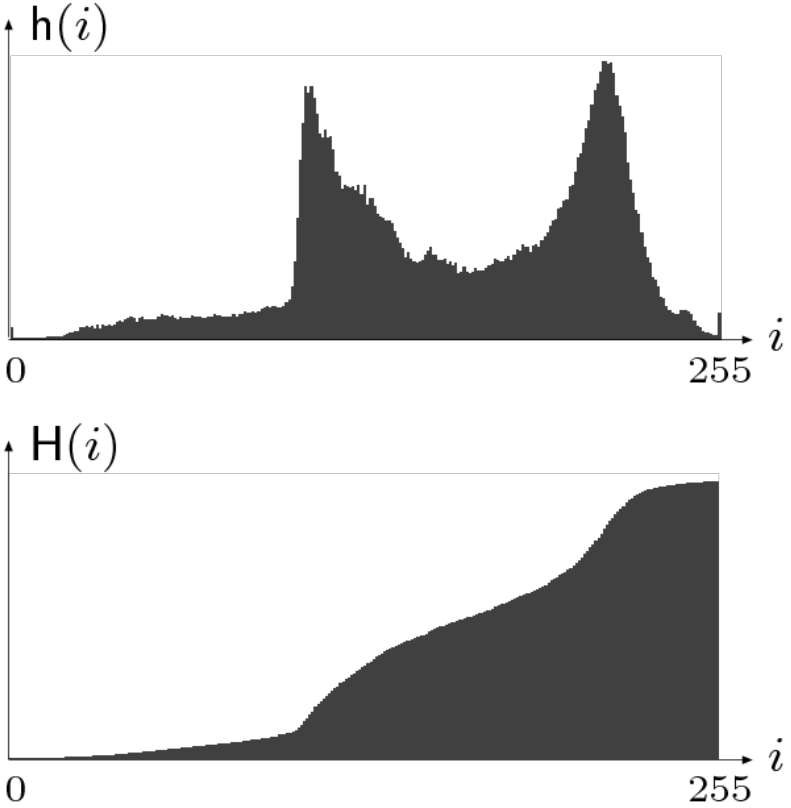
$$\hat{a}_{\text{low}} = \min\{\, i \mid \mathrm{H}(i) \geq M \cdot N \cdot s_{\text{low}} \}$$

$$\hat{a}_{\text{high}} = \max\{\, i \mid \mathrm{H}(i) \leq M \cdot N \cdot (1 - s_{\text{high}}) \}$$



$$f_{\text{mac}}(a) = \begin{cases} a_{\text{min}} & \text{for } a \leq \hat{a}_{\text{low}} \\ a_{\text{min}} + (a - \hat{a}_{\text{low}}) \cdot \dfrac{a_{\text{max}} - a_{\text{min}}}{\hat{a}_{\text{high}} - \hat{a}_{\text{low}}} & \text{for } \hat{a}_{\text{low}} < a < \hat{a}_{\text{high}} \\ a_{\text{max}} & \text{for } a \geq \hat{a}_{\text{high}} \end{cases}$$

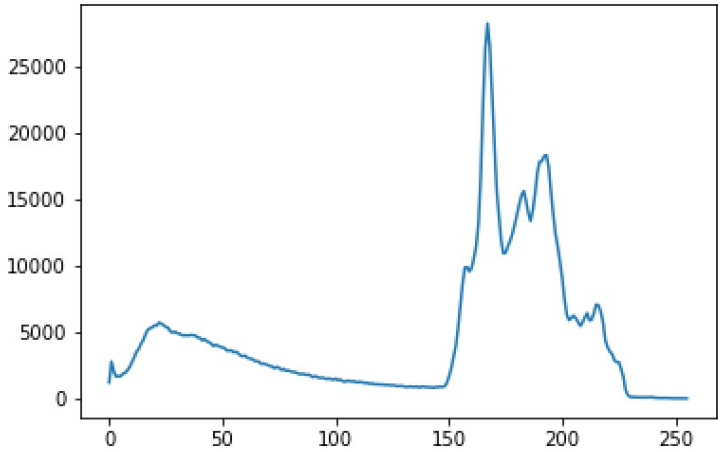
Cumulative histogram

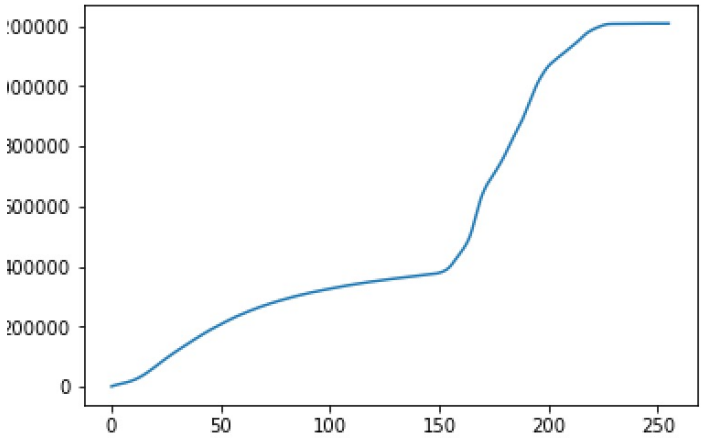


$$\mathsf{H}(i) = \sum_{j=0}^{i} \mathsf{h}(j) \quad \text{for} \quad 0 \leq i < K$$

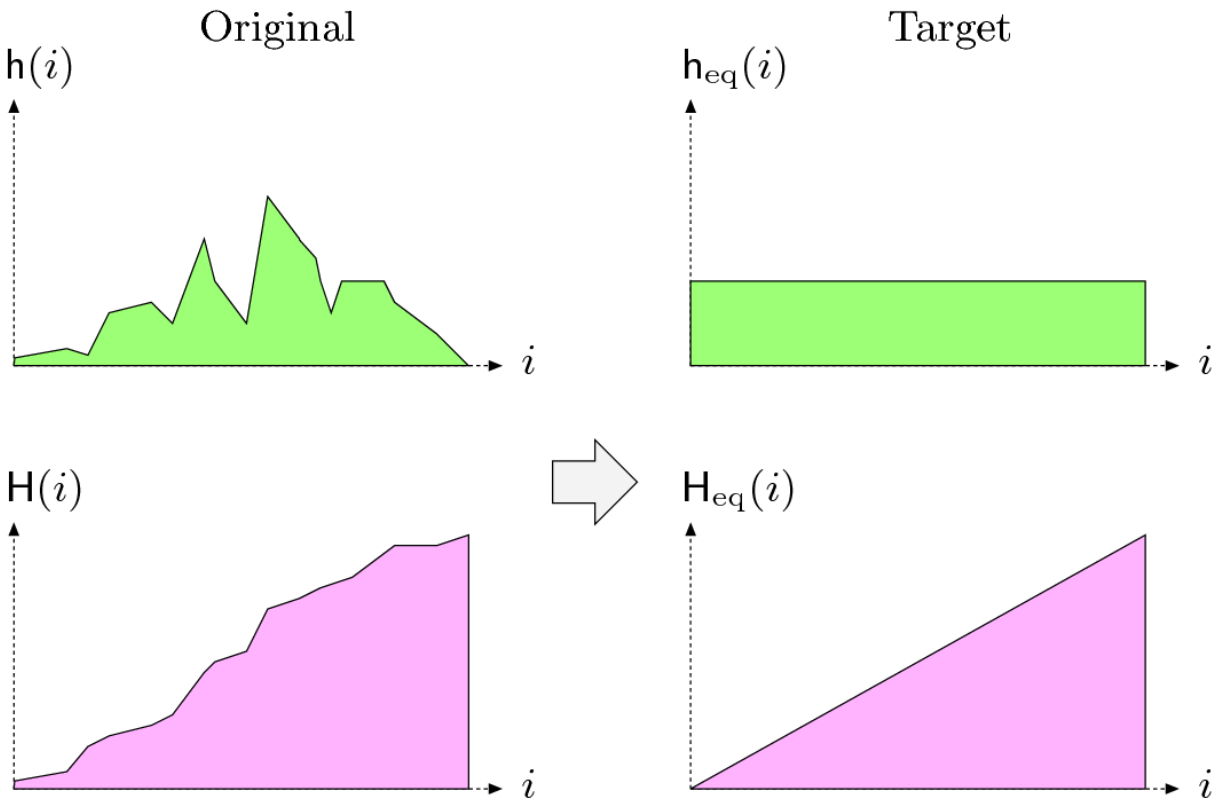
Cumulative histogram

```
# cumulative hist
cumhist = np.zeros(256)
cumhist[0] = histogram[0]
for i in range(255):
    cumhist[i+1] = cumhist[i]+histogram[i+1]

plt.plot(cumhist)
```





Histogram equalisation



Histogram equalisation

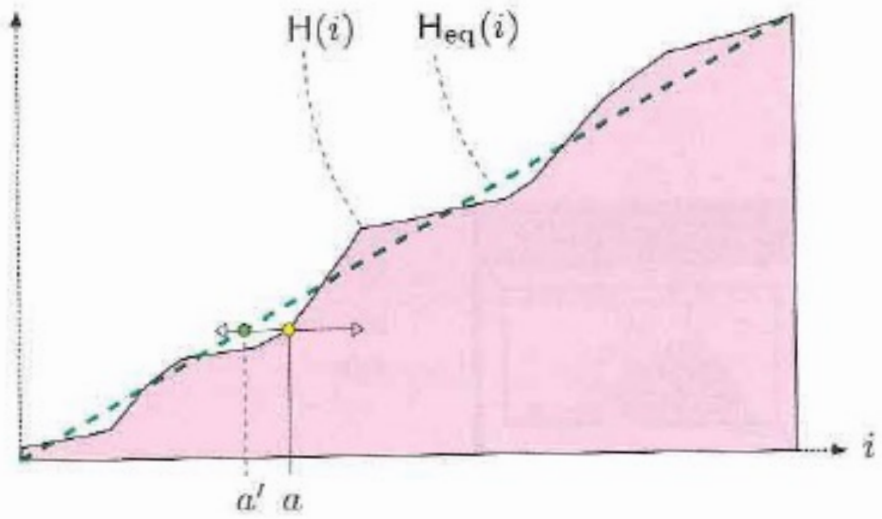
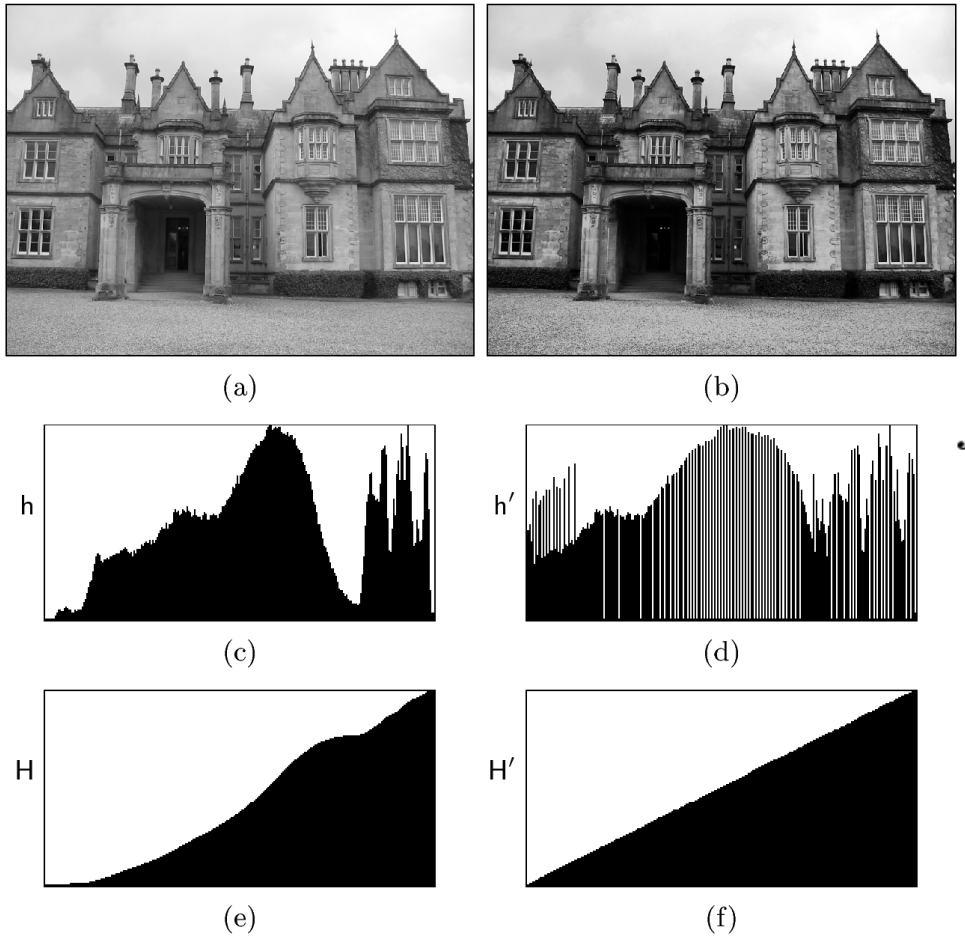


Figure 4.9 Histogram equalization on the cumulative histogram. A suitable point operation $a' \leftarrow f_{eq}(a)$ shifts each histogram line from its original position a to a' (left or right) such that the resulting cumulative histogram H_{eq} is approximately linear.

$$f_{eq}(a) = \left\lfloor \mathrm{H}(a) \cdot \frac{K-1}{MN} \right\rfloor$$

Range=[0,K-1]
M=height
N=width

Histogram equalisation

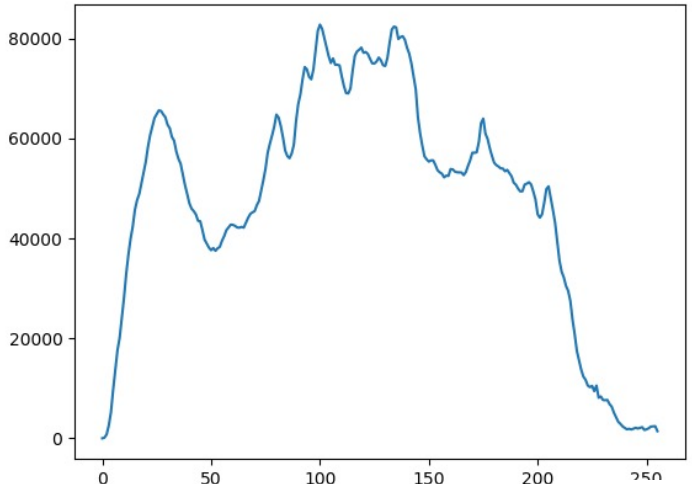


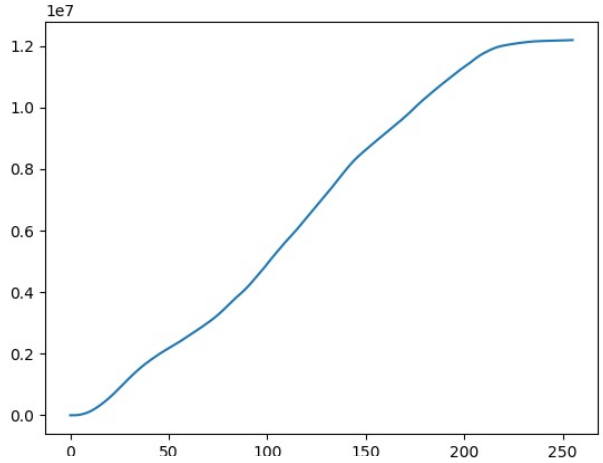
$$\tilde{\mathsf{H}}(i) = \sum_{j=0}^{i} \sqrt{\mathsf{H}(i)}$$

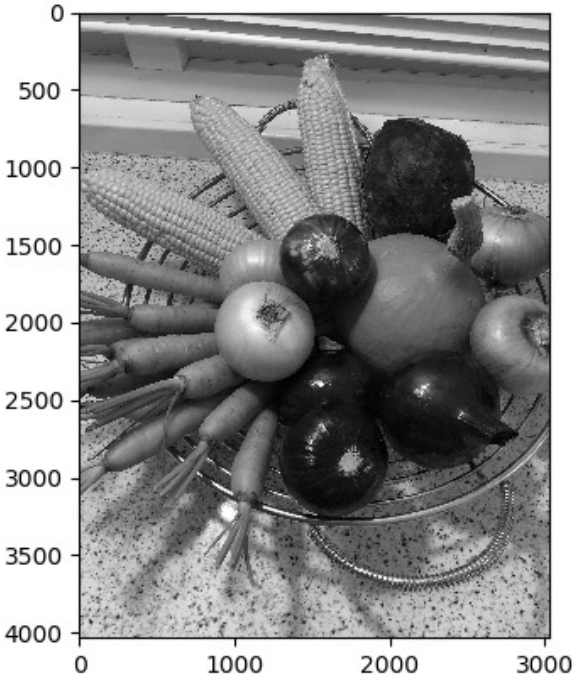
$$f_{\mathrm{eq}}(a) = \left\lfloor \mathsf{H}(a) \cdot \frac{K-1}{MN} \right\rfloor$$

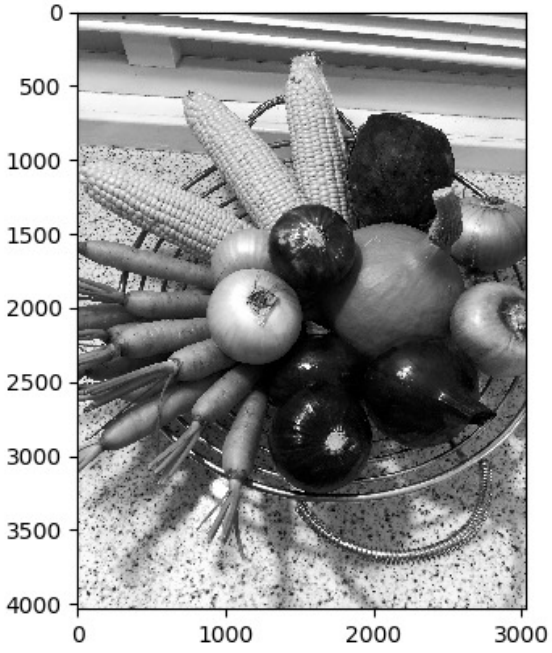
Program: Histogram equalisation

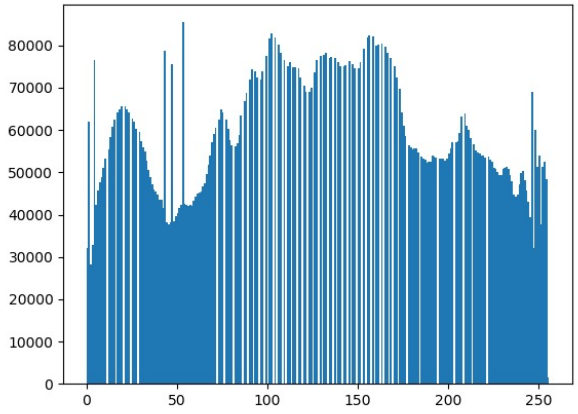
```
# cumulative hist
cumhist = np.zeros(256)
cumhist[0] = histogram[0]
for i in range(255):
    cumhist[i+1] = cumhist[i]+histogram[i+1]

# equalisation
for i in range(shape[0]):
    for j in range(shape[1]):
        a = int(ima[i,j])
        b = cumhist[a]*(K-1)/M
        ima[i,j] = b
```


80000
60000
40000
20000
0
0
50
100
150
200
250


1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000


80000
70000
60000
50000
40000
30000
20000
10000
0
0
50
100
150
200
250

Histogram specification

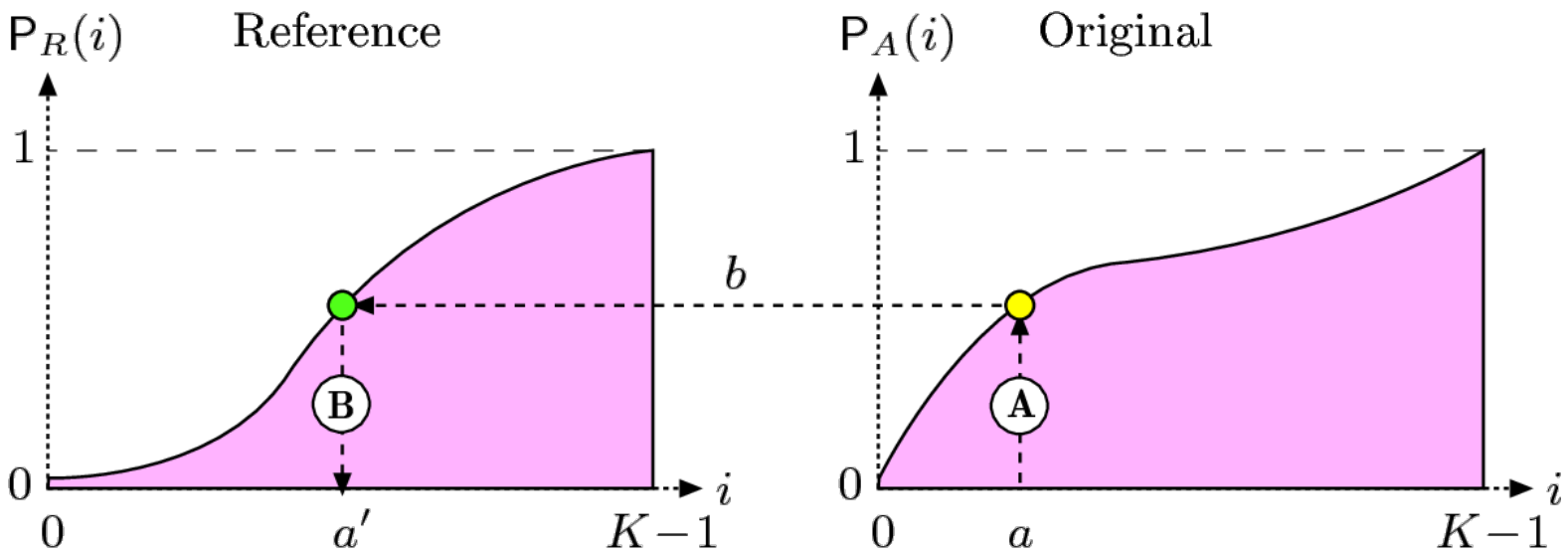
- Histogram specification can be approached through the cumulated histogram
- We have the original image A with histogram distribution P_A
- We have a reference image R with histogram distribution P_R
- We transform A so that the histrogram P_A is as equal as possible to the histogram P_R

$$P_{A'}(i) \approx P_R(i) \qquad \text{for} \;\; 0 \leq i < K$$

The pixel value a is transformed to a'

$$a' = P_R^{-1}\big(P_A(a)\big)$$

Histogram specification



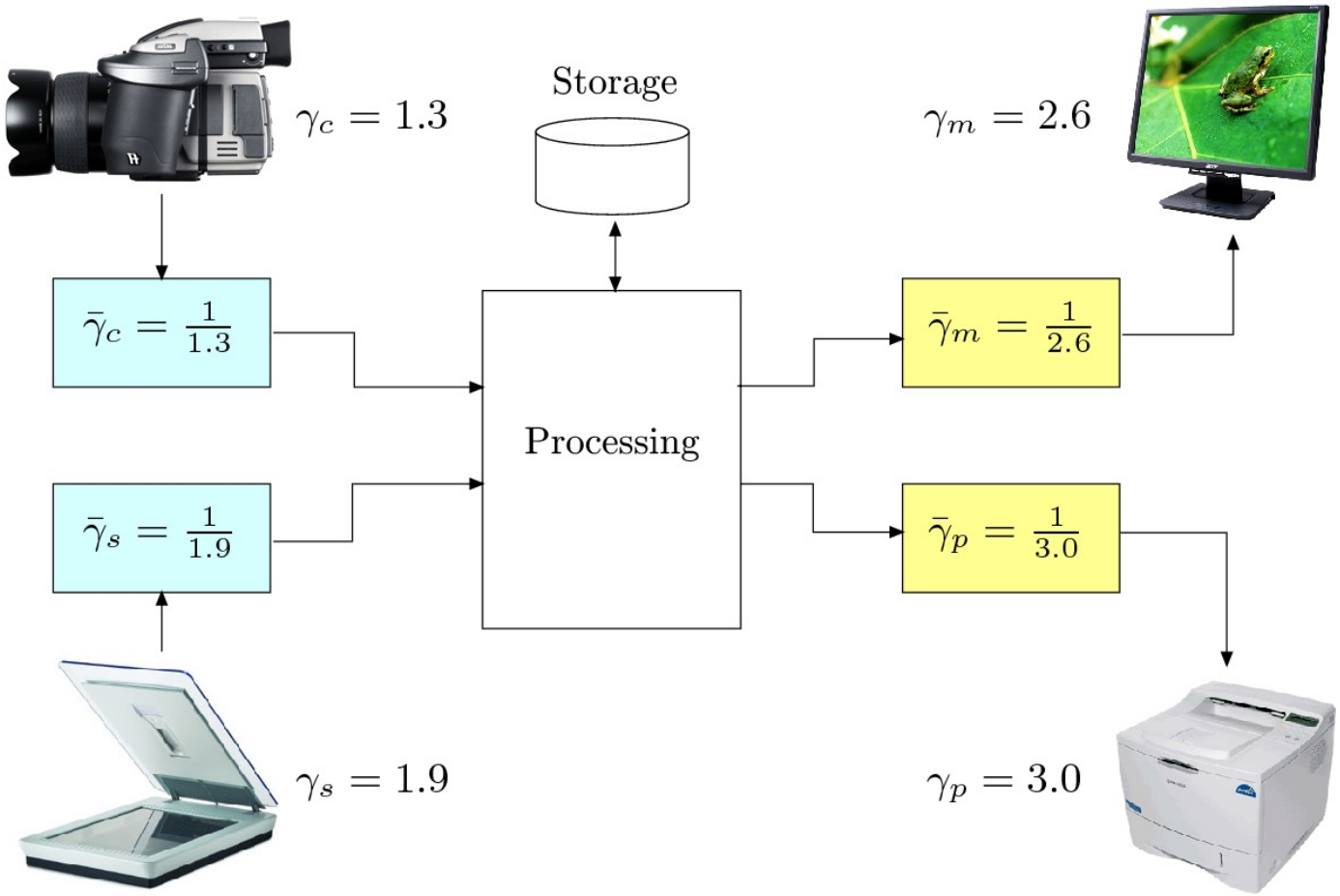
$$a' = f_{\text{hs}}(a)$$

$$a' = \mathsf{P}_R^{-1}\big(\mathsf{P}_A(a)\big)$$

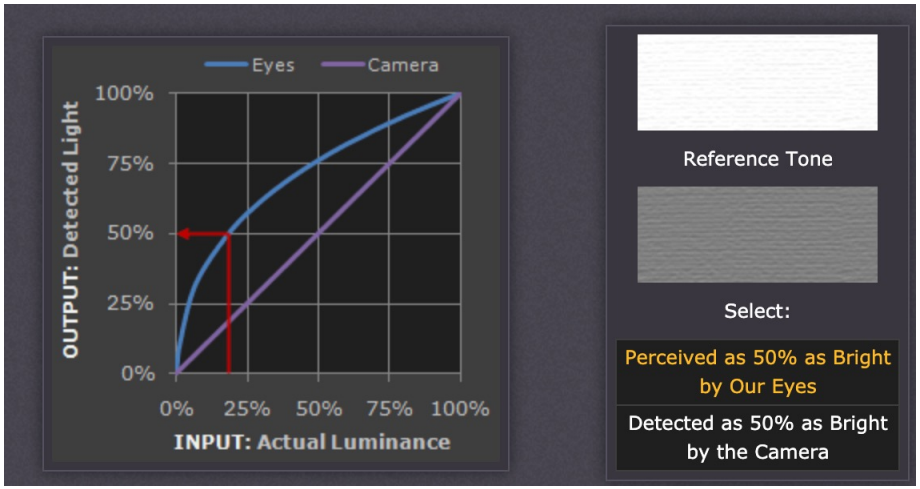
$$f_{\text{hs}}(a) = a' = \mathsf{P}_R^{-1}\big(\mathsf{P}_A(a)\big)$$

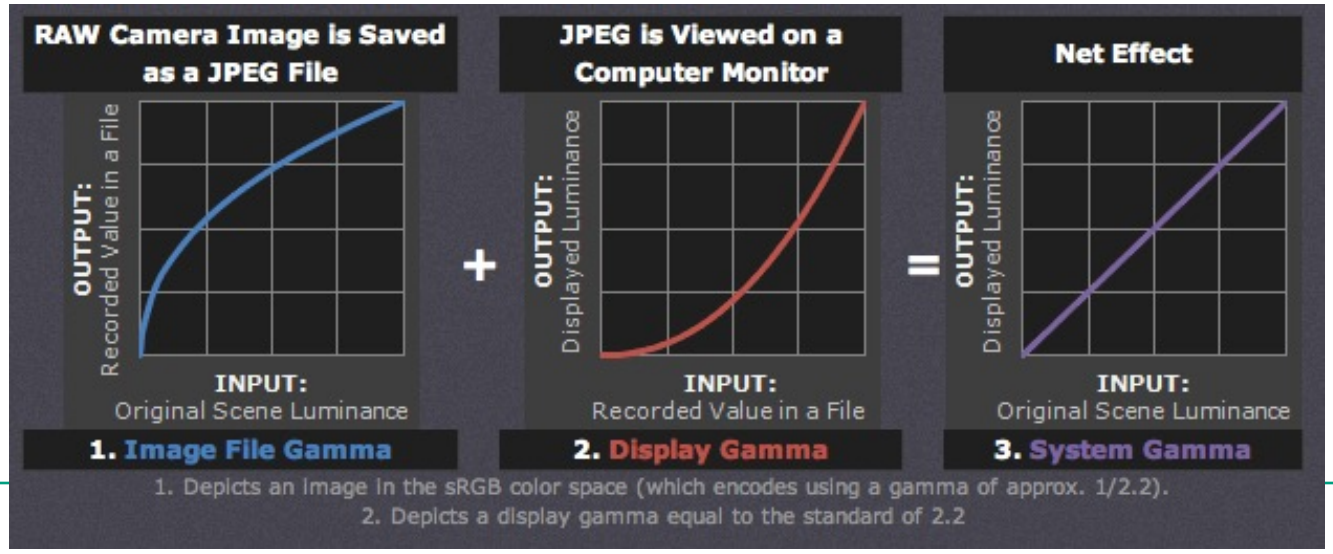
$$\mathsf{P}_{A'}(i) \approx \mathsf{P}_R(i) \qquad \text{for } 0 \le i < K$$

Gamma scale


$\gamma_c = 1.3$
Storage
$\gamma_m = 2.6$
$\bar{\gamma}_c = \frac{1}{1.3}$
Processing
$\bar{\gamma}_m = \frac{1}{2.6}$
$\bar{\gamma}_s = \frac{1}{1.9}$
$\bar{\gamma}_p = \frac{1}{3.0}$
$\gamma_s = 1.9$
$\gamma_p = 3.0$

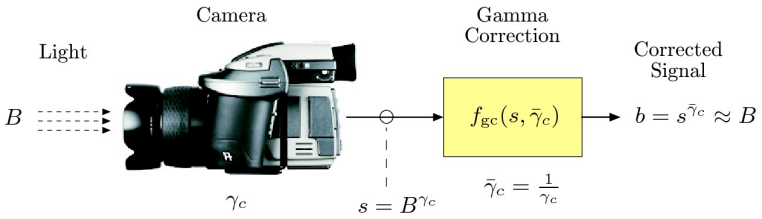
Gamma scale

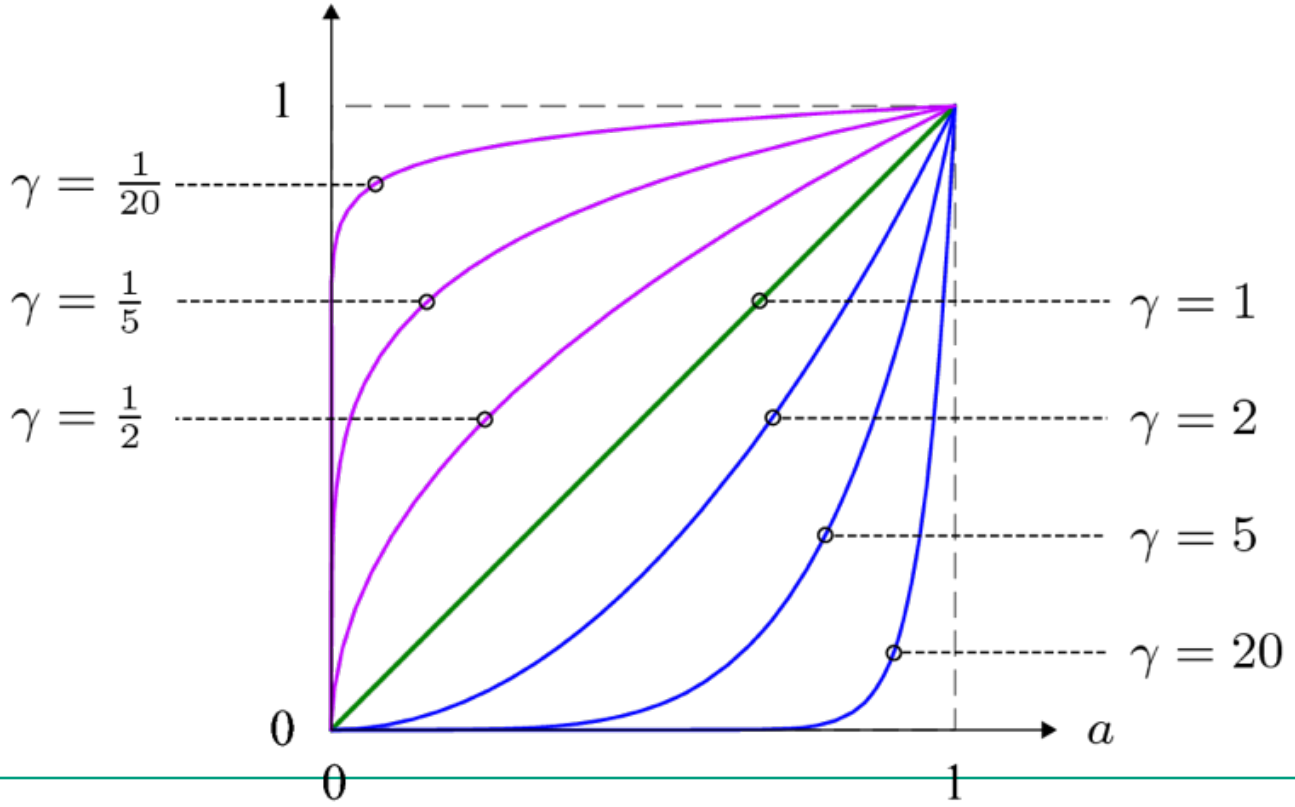




http://www.cambridgeincolour.com/tutorials/gamma-correction.htm

GAMMA


Camera
Light
Gamma Correction
Corrected Signal
B
$f_{\mathrm{gc}}(s, \bar{\gamma}_c)$
$b = s^{\bar{\gamma}_c} \approx B$
γ_c
$s = B^{\gamma_c}$
$\bar{\gamma}_c = \frac{1}{\gamma_c}$

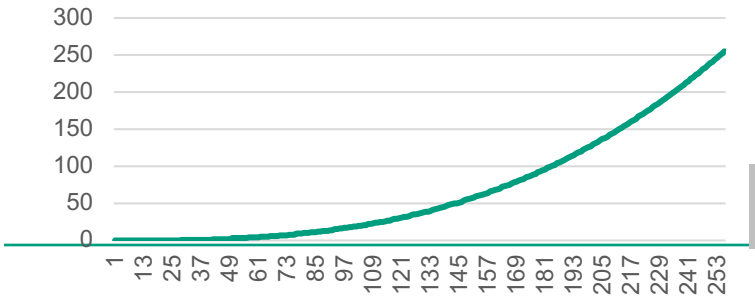

1
$\gamma = \frac{1}{20}$
$\gamma = \frac{1}{5}$
$\gamma = \frac{1}{2}$
$\gamma = 1$
$\gamma = 2$
$\gamma = 5$
$\gamma = 20$
0
a
0
1

«Using gamma» in an artistic way

Original

Gamma 0.2
Histogram eq.

Difference between
Original and
Gamma 0.2
Histogram eq.



Gamma in ImageJ: Process – Math - Gamma

ALPHA BLENDING

- A technique to play with transparency between two images

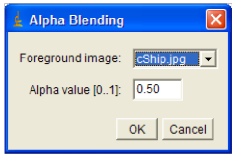
$$I'(u,v) \leftarrow \alpha \cdot I_{\mathrm{BG}}(u,v) + (1-\alpha) \cdot I_{\mathrm{FG}}(u,v)$$

I_{BG}

$\alpha = 0.25$

I_{FG}

$\alpha = 0.50$



`GenericDialog`

$\alpha = 0.75$