AGH

RESEARCH UNIVERSITY
EXCELLENCE INITIATIVE
Ministry of Science and Higher Education

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

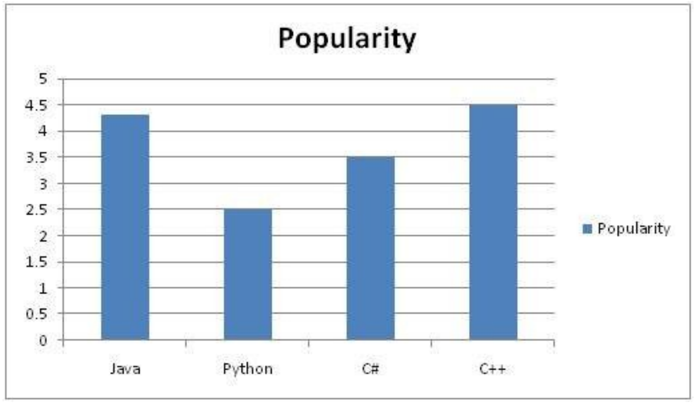
Histogram Stretching Filter

20.04.2022

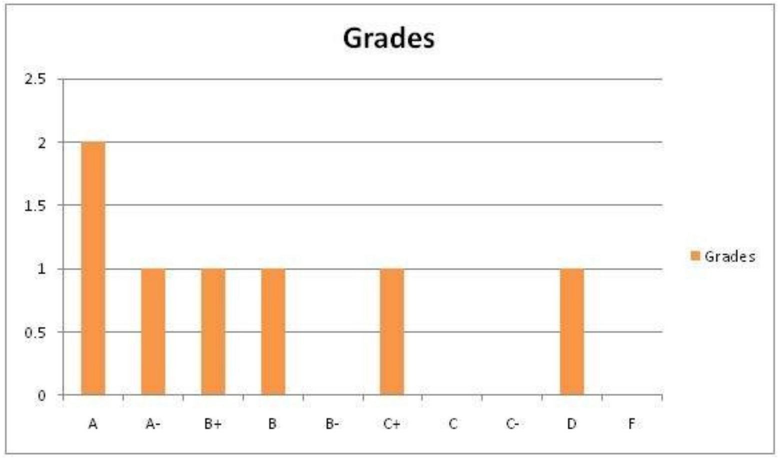
agh.edu.pl

Histogram

A graph that shows frequency of anything



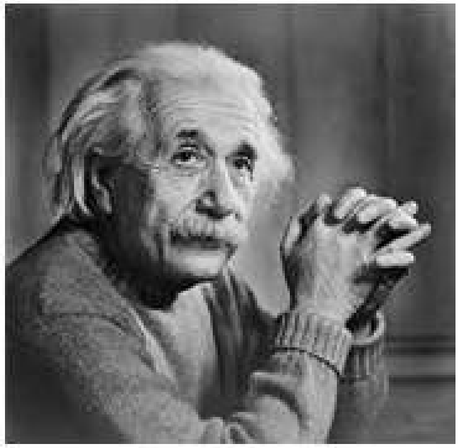
Name	Grade
John	A
Jack	D
Carter	B
Tommy	A
Lisa	C+
Derek	A-
Tom	B+



https://www.tutorialspoint.com/dip

Histogram of an image

The x axis of the histogram shows the range of pixel values. Since its an 8 bpp image, that means it has 256 levels of gray or shades of gray in it. Thats why the range of x axis starts from 0 and end at 255 with a gap of 50. Whereas on the y axis, is the count of these intensities.



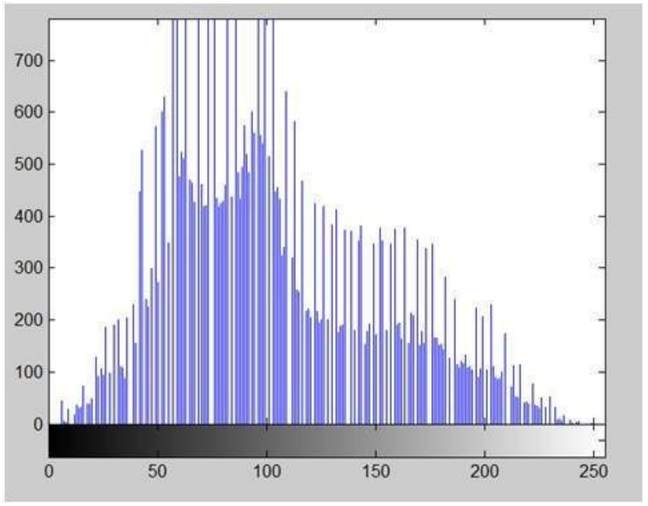


Image transformation

G(x,y) = T{ f(x,y) }

F(x,y) = input image on which transformation function has to be applied.
G(x,y) = the output image or processed image.
T is the transformation function.

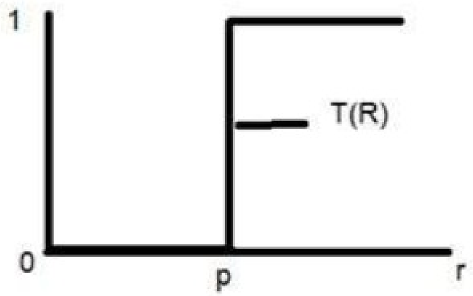
This relation between input image and the processed output image can also be represented as.

s = T (r)

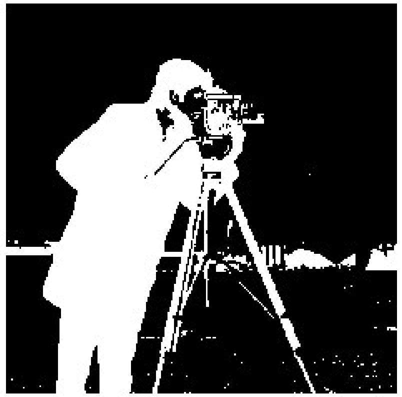
where r is actually the pixel value or gray level intensity of f(x,y) at any point. And s is the pixel value or gray level intensity of g(x,y) at any point.

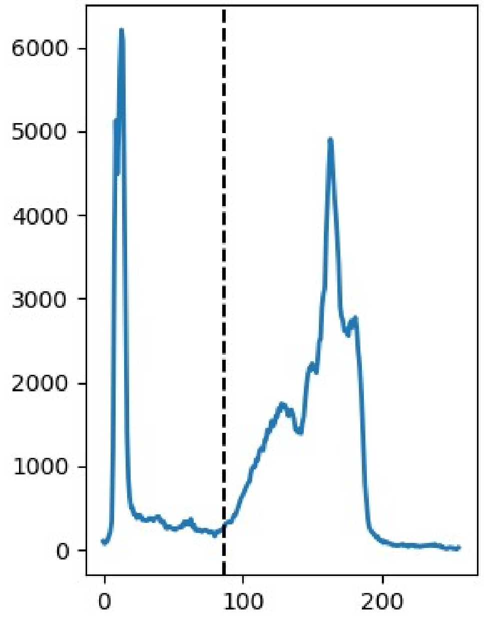
https://www.tutorialspoint.com/dip

Image transformation – example - thresholding



$$g(x,y) = \begin{cases} 0 & f(x,y) < p \\ 1 & f(x,y) > p \end{cases}$$

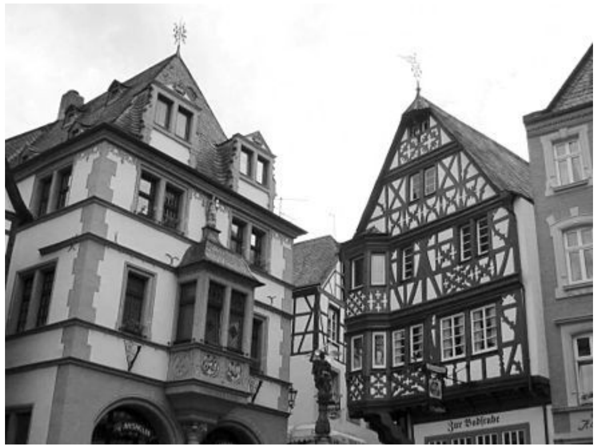




https://www.tutorialspoint.com/dip

Image transformation – example – thresholding
Otsu’s method

In computer vision and image processing, Otsu's method, named after Nobuyuki Otsu (大津展之 , Ōtsu Nobuyuki), is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance.



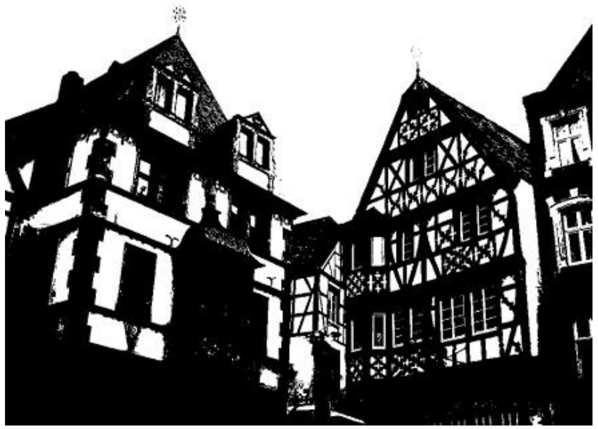
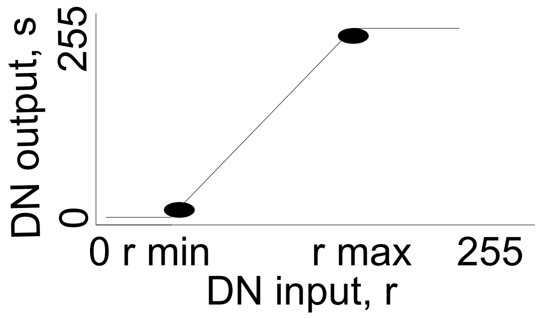


Image transformation – example - stretching

Contrast stretching is only possible if minimum intensity value and maximum intensity value are not equal to the possible minimum and maximum intensity values. Otherwise, the image generated after contrast stretching will be the same as input image.

General Formula for Contrast Stretching:

$$s = (r - r_{min}) \frac{(I_{max} - I_{min})}{(r_{max} - r_{min})} + I_{min}$$



For I min = 0 and I max = 255 (for standard 8-bit grayscale image)

$$s = 255 \times \frac{(r - r_{min})}{(r_{max} - r_{min})}$$

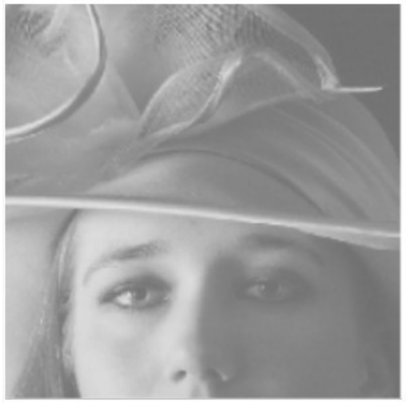
r = current pixel intensity value

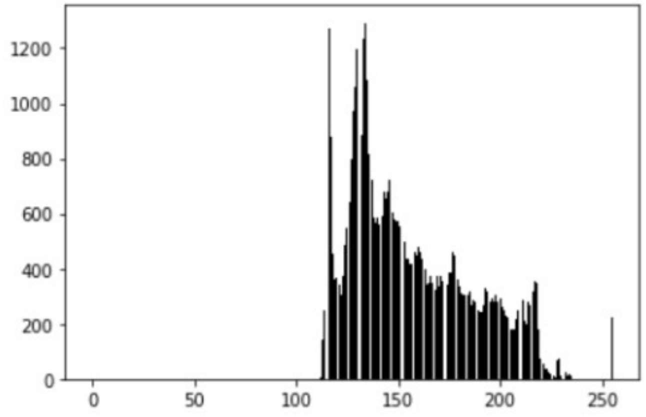
r min = minimum intensity value present in the whole image

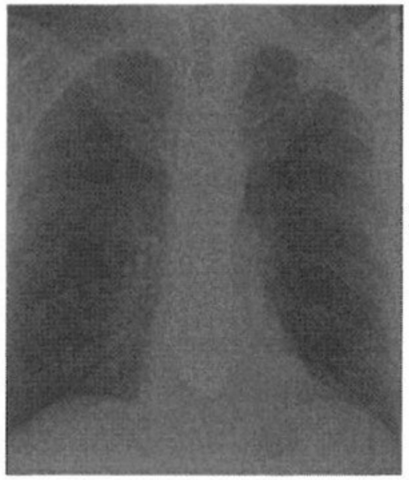
r max = maximum intensity value present in the whole imag

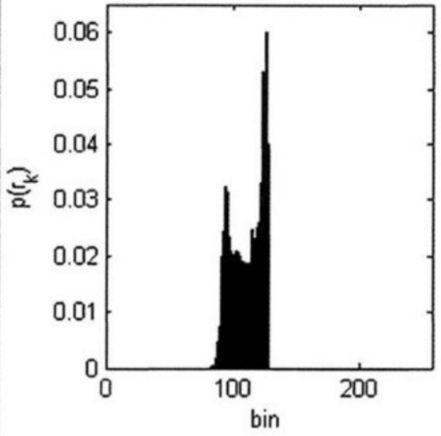
https://samirkhanal35.medium.com/contrast-stretching-f25e7c4e8e33

Image transformation – example - stretching

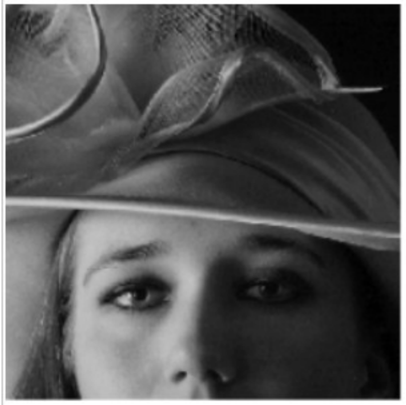


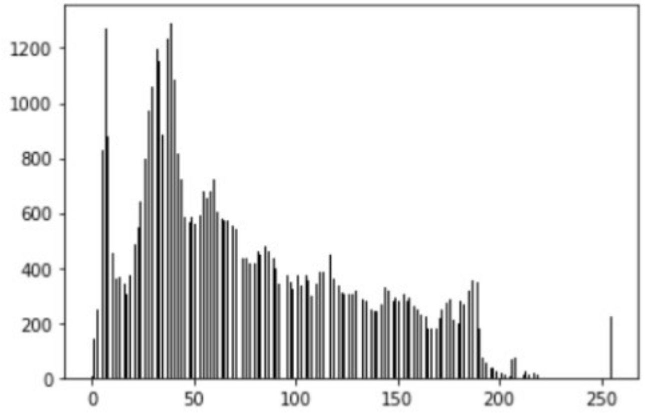


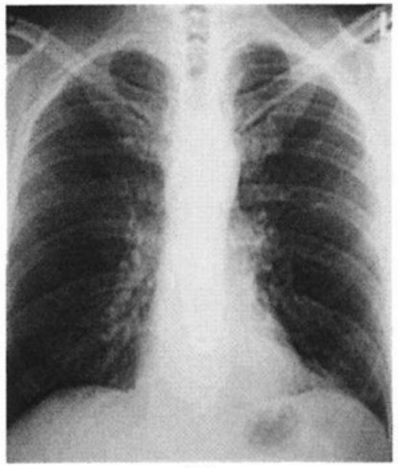


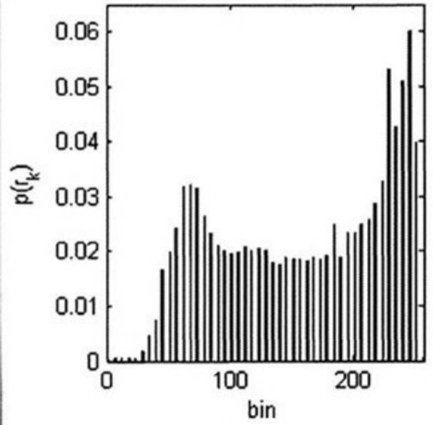


(a) (b)









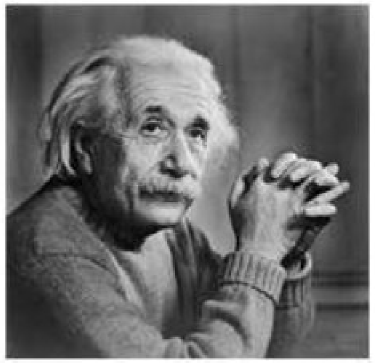
(a) (b)

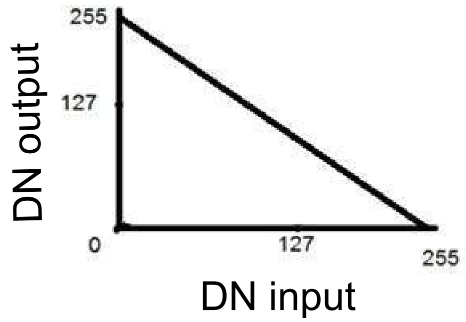
https://samirkhanal35.medium.com/contrast-stretching-f25e7c4e8e33

http://what-when-how.com/embedded-image-processing-on-the-tms320c6000-dsp/contrast-stretching-image-processing/

Negative transformation

Input Image





Output Image

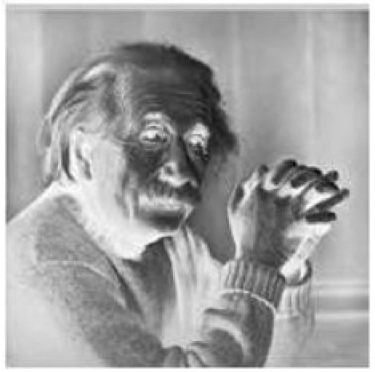


Image transformation – stretching

Contrast stretching

- Linear
- Linear with saturation i.e. 5% of histogram is cut-off
- ……..

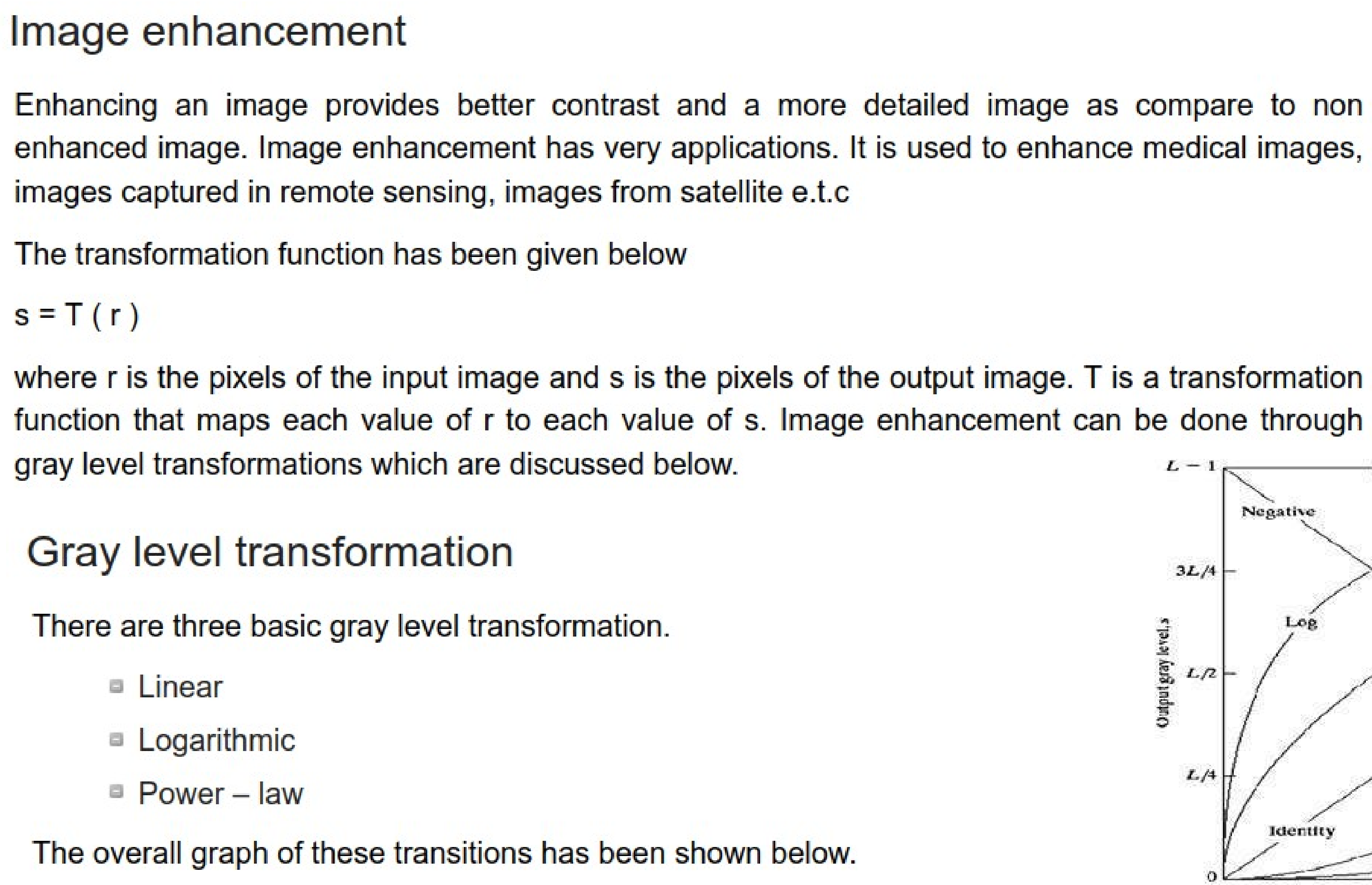


Image enhancement

Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. Image enhancement has very applications. It is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c

The transformation function has been given below

s = T (r)

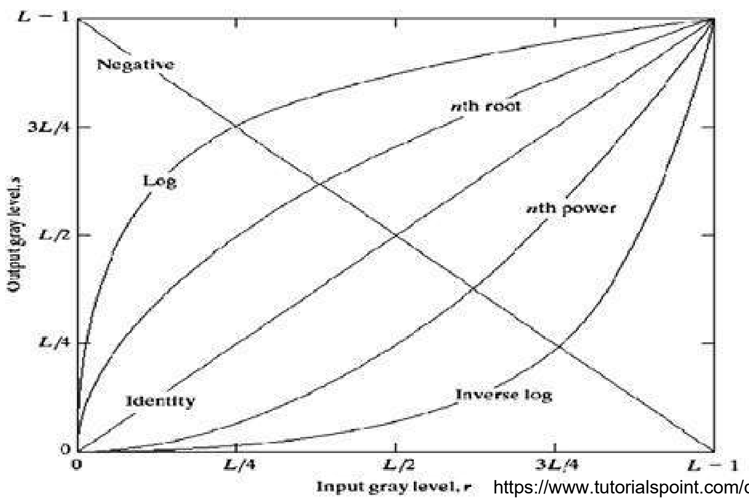
where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of r to each value of s. Image enhancement can be done through gray level transformations which are discussed below.

Gray level transformation

There are three basic gray level transformation.

- Linear
- Logarithmic
- Power – law

The overall graph of these transitions has been shown below.

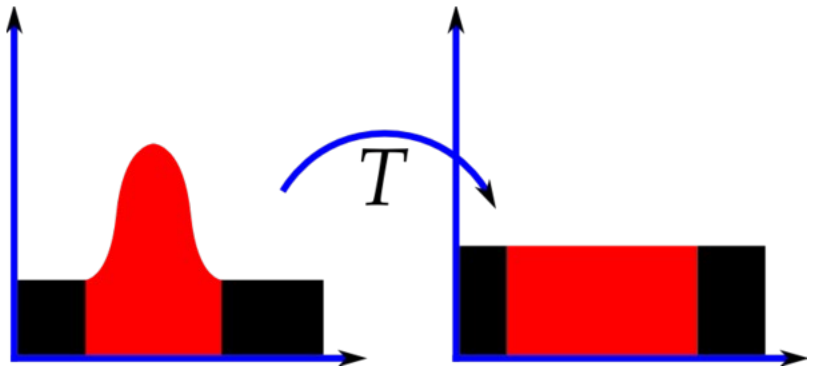


https://www.tutorialspoint.com/dip

Increasing the contrast of the image – stretching - equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.
This allows for areas of lower local contrast to gain a higher contrast.

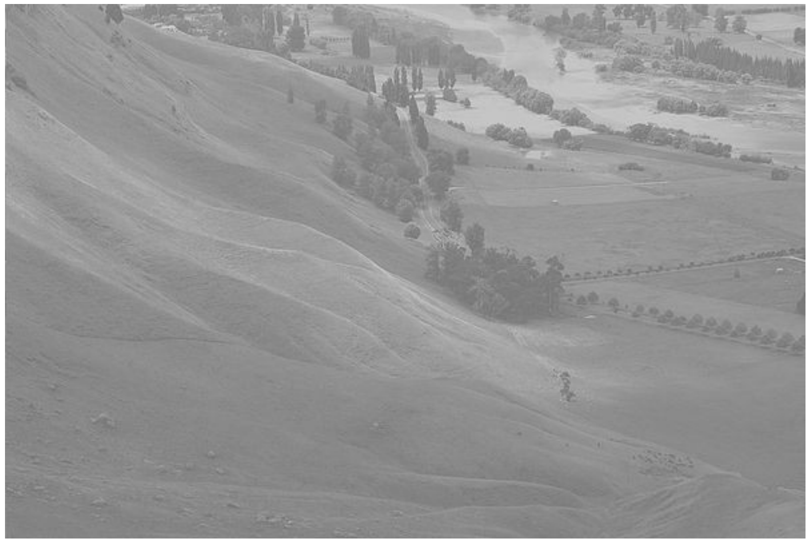
Back projection

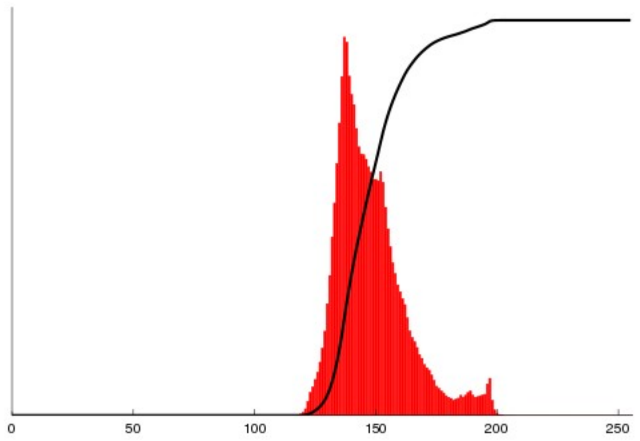


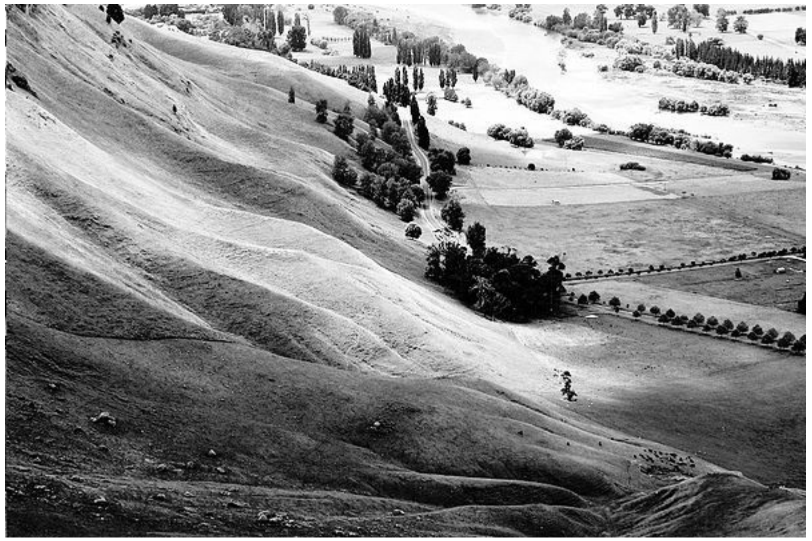
The back projection (or "project") of a histogrammed image is the re-application of the modified histogram to the original image, functioning as a look-up table for pixel brightness values.
For each group of pixels taken from the same position from all input single-channel images, the function puts the histogram bin value to the destination image, where the coordinates of the bin are determined by the values of pixels in this input group. In terms of statistics, the value of each output image pixel characterizes the probability that the corresponding input pixel group belongs to the object whose histogram is used.

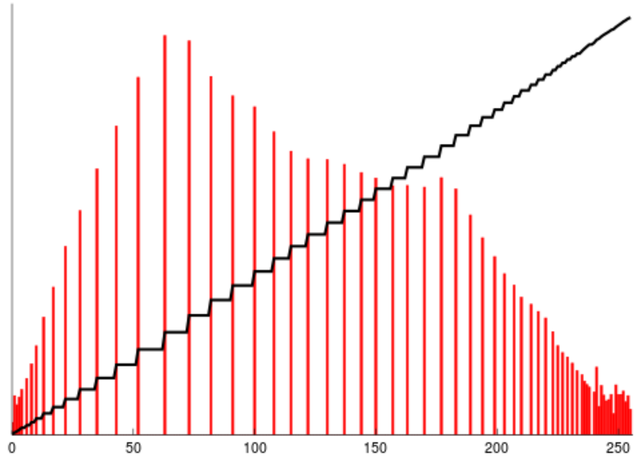
https://en.wikipedia.org/wiki/Histogram_equalization

Increasing the contrast of the image – stretching - equalization







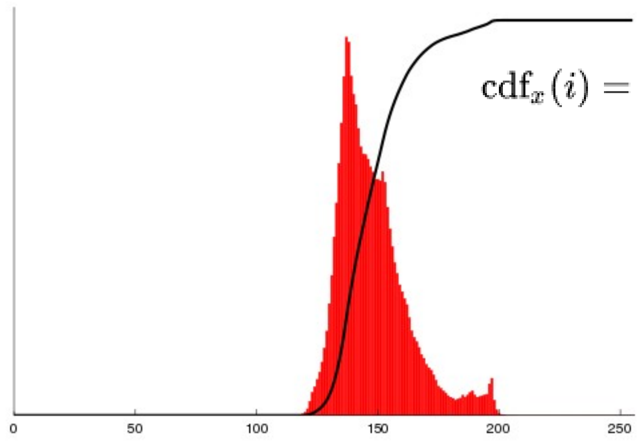


https://en.wikipedia.org/wiki/Histogram_equalization

Increasing the contrast of the image – stretching - equalization

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \le i < L$$

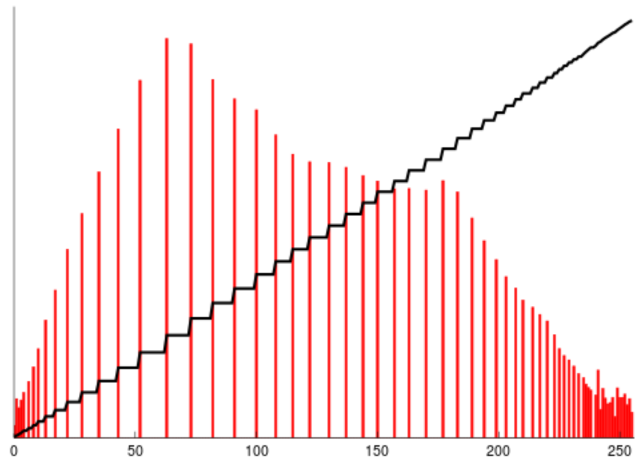
The probability of an occurrence of a pixel of level i in the image



$$\mathrm{cdf}_x(i) = \sum_{j=0}^{i} p_x(x = j)$$

The cumulative distribution function (CDF)

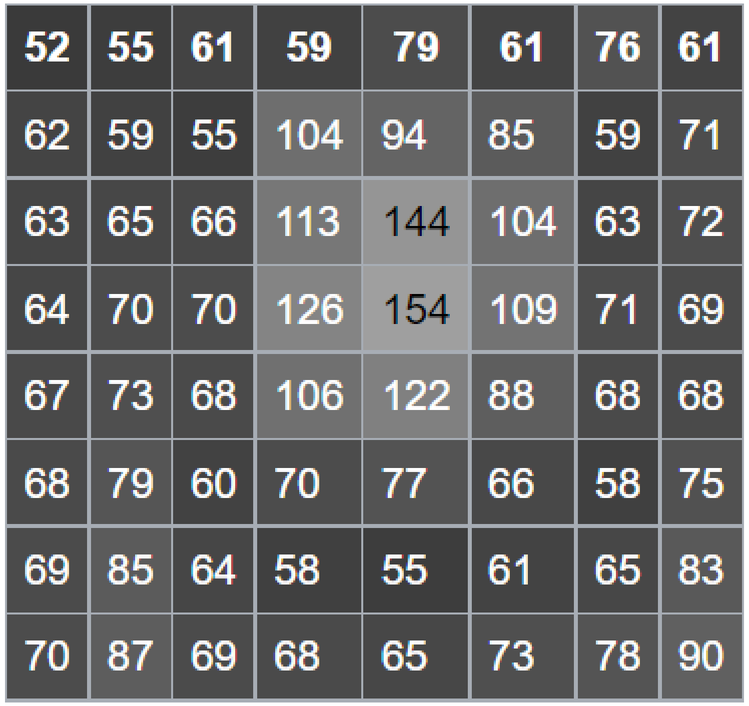
We would like to create a transformation of the form $y = T(x)$ to produce a new image {y}, with a flat histogram. Such an image would have a linearized cumulative distribution function (CDF) across the value range, i.e.



$\mathrm{cdf}_y(i) = (i+1)K$ for $0 \le i < L$ for some constant K.

https://en.wikipedia.org/wiki/Histogram_equalization

Increasing the contrast of the image – stretching - equalization

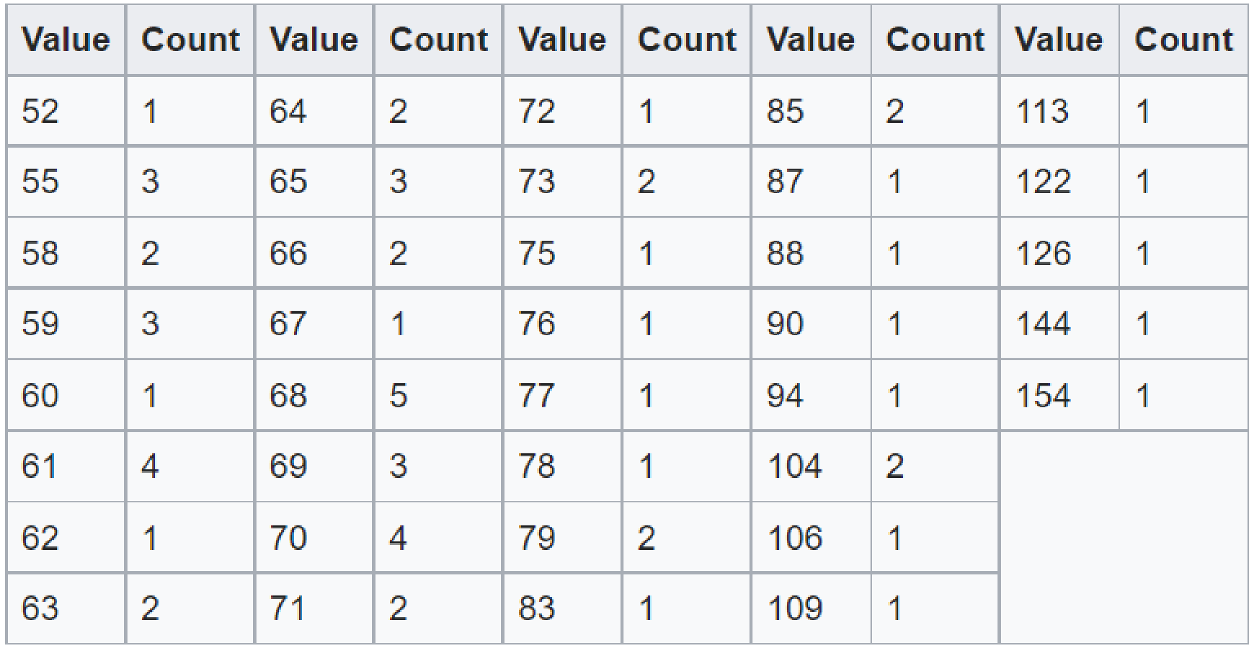


52	55	61	59	79	61	76	61
62	59	55	104	94	85	59	71
63	65	66	113	144	104	63	72
64	70	70	126	154	109	71	69
67	73	68	106	122	88	68	68
68	79	60	70	77	66	58	75
69	85	64	58	55	61	65	83
70	87	69	68	65	73	78	90

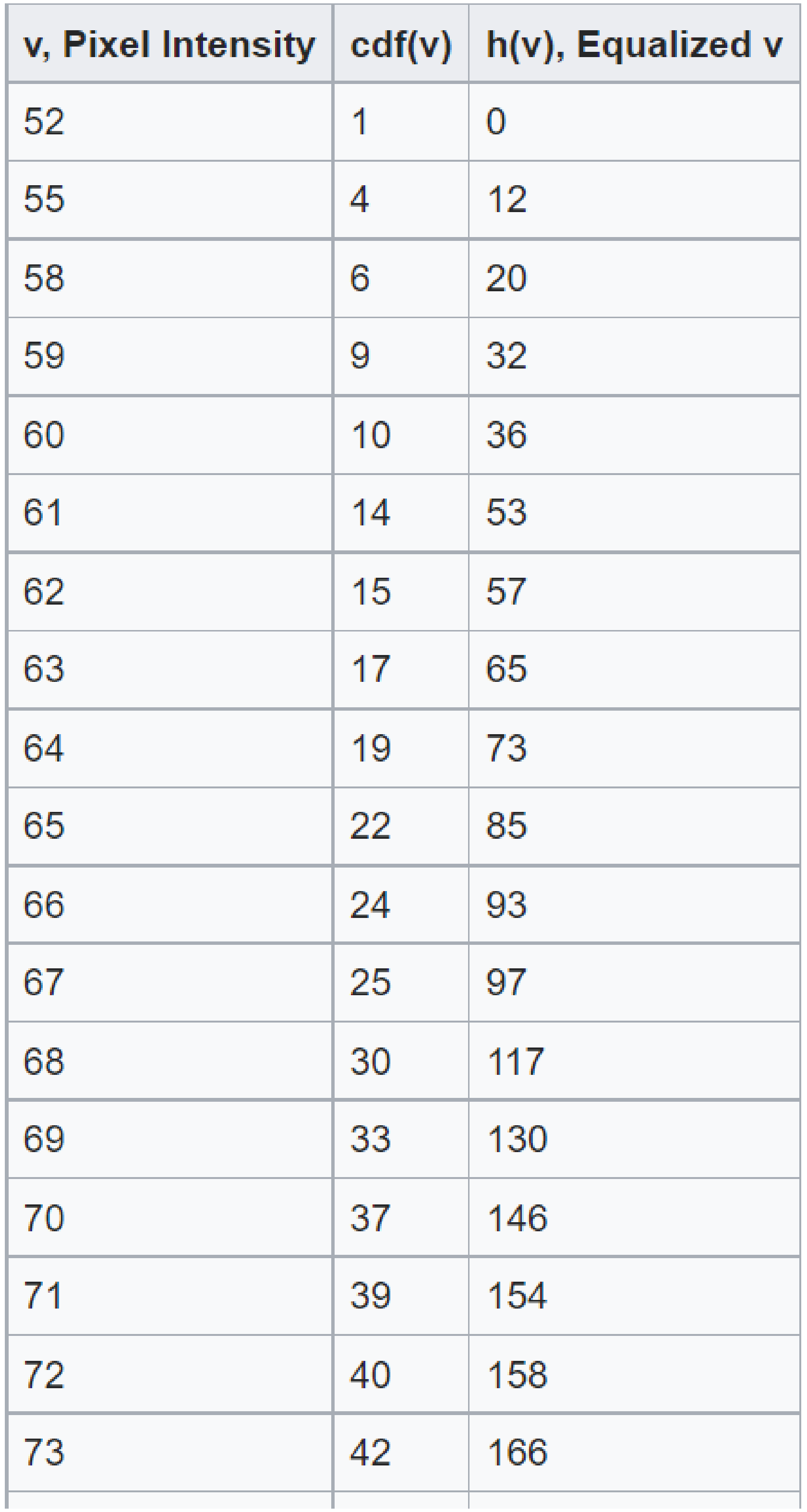
Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

https://en.wikipedia.org/wiki/Histogram_equalization

Increasing the contrast of equalization



Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
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		



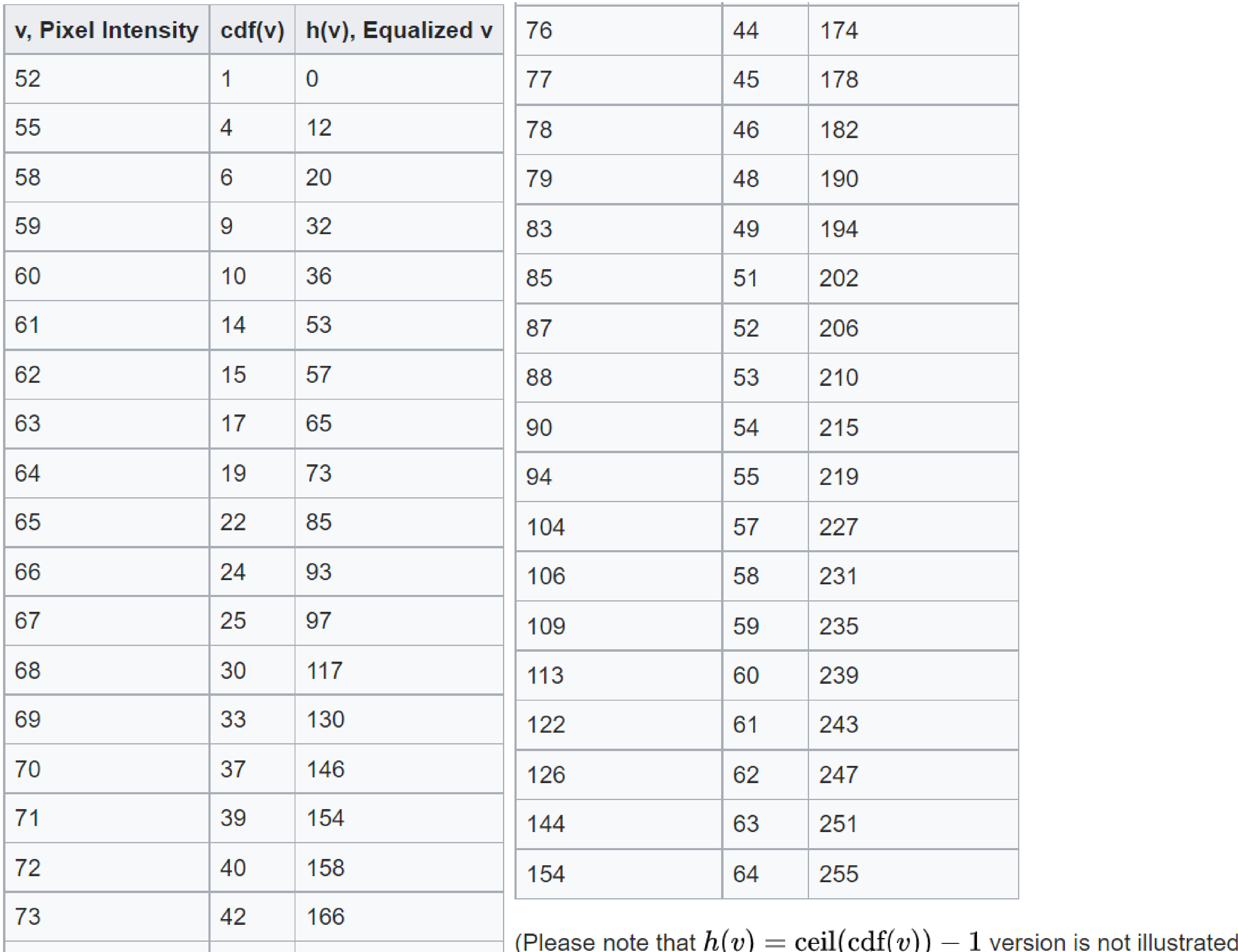
v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53
62	15	57
63	17	65
64	19	73
65	22	85
66	24	93
67	25	97
68	30	117
69	33	130
70	37	146
71	39	154
72	40	158
73	42	166

Increasing the contrast of the image – stretching - equalization

This cdf shows that the minimum value in the subimage is 52 and the maximum value is 154. The cdf of 64 for value 154 coincides with the number of pixels in the image. The cdf must be normalized to [0,255]. The general histogram equalization formula is:

$$h(v) = \mathrm{round}\left(\frac{\mathrm{cdf}(v) - \mathrm{cdf}_{\min}}{\underset{\substack{\downarrow\\ \text{cdfmax}}}{(M \times N)} - \mathrm{cdf}_{\min}} \times (L-1)\right)$$

where cdfmin is the minimum non-zero value of the cumulative distribution function (in this case 1), M × N gives the image's number of pixels (for the example above 64, where M is width and N the height) and L is the number of grey levels used (in most cases, like this one, 256).



v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53
62	15	57
63	17	65
64	19	73
65	22	85
66	24	93
67	25	97
68	30	117
69	33	130
70	37	146
71	39	154
72	40	158
73	42	166
76	44	174
77	45	178
78	46	182
79	48	190
83	49	194
85	51	202
87	52	206
88	53	210
90	54	215
94	55	219
104	57	227
106	58	231
109	59	235
113	60	239
122	61	243
126	62	247
144	63	251
154	64	255

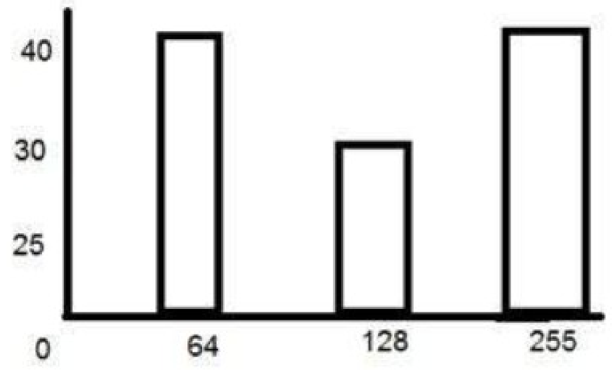
(Please note that $h(v) = \mathrm{ceil}(\mathrm{cdf}(v)) - 1$ version is not illustrated

Increasing the contrast of the image – stretching - equalization

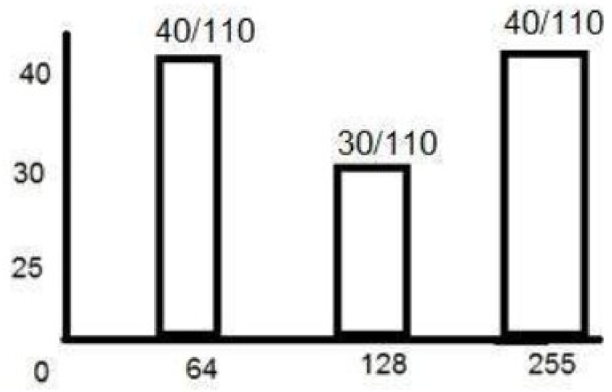
What is PMF?

PMF stands for probability mass function. As it name suggest, it gives the probability of each number in the data set or you can say that it basically gives the count or frequency of each element.

Calculating PMF from histogram



So the PMF of the above histogram is this.



https://www.tutorialspoint.com/dip

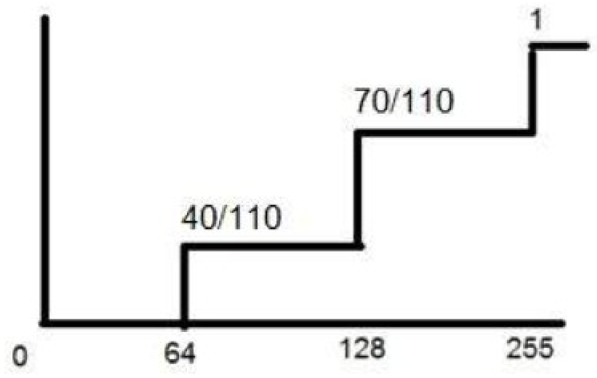
agh.edu.pl

Increasing the contrast of the image – stretching - equalization

What is CDF?

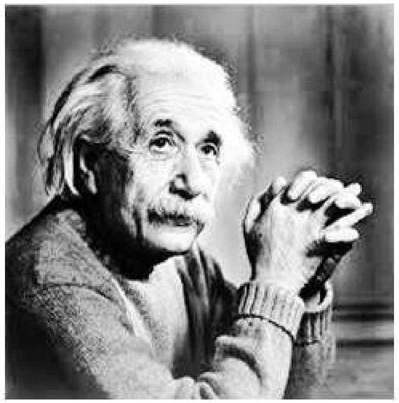
CDF stands for cumulative distributive function. It is a function that calculates the cumulative sum of all the values that are calculated by PMF. It basically sums the previous one.

Here is the CDF of the above PMF function.

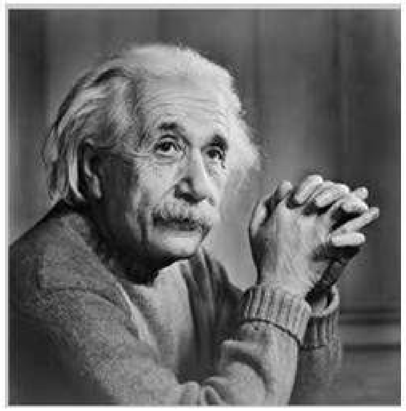


Increasing the contrast of the image – stretching - equalization

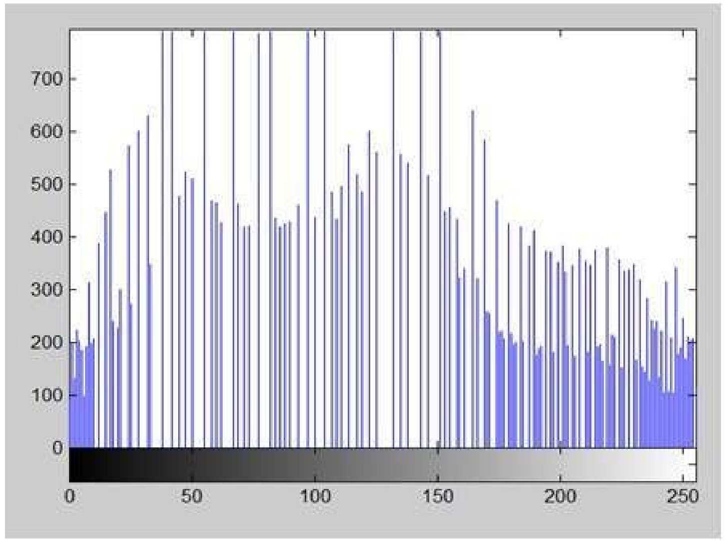
New Image



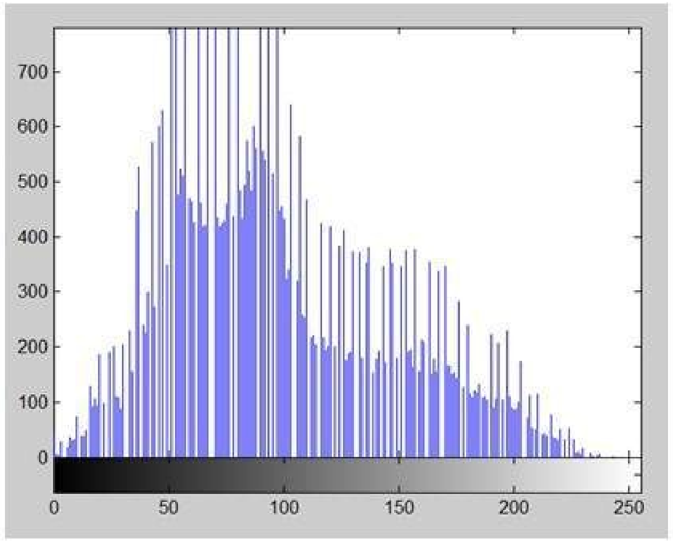
Old image



New Histogram



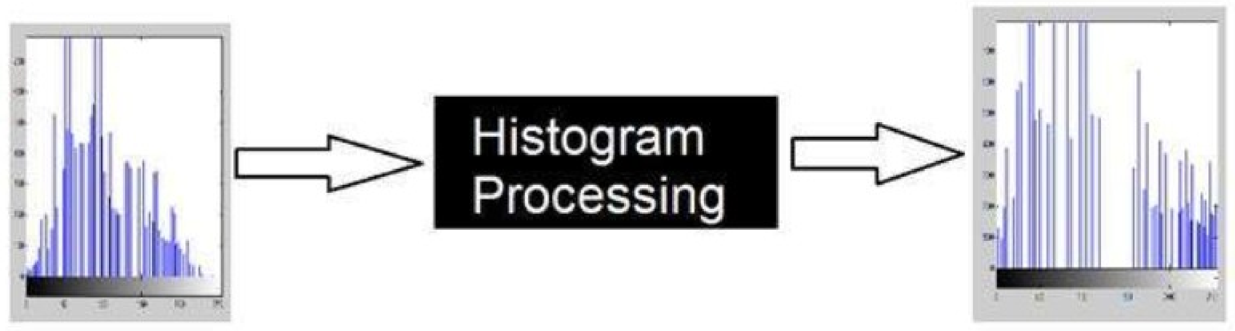
Old Histogram



https://www.tutorialspoint.com/dip

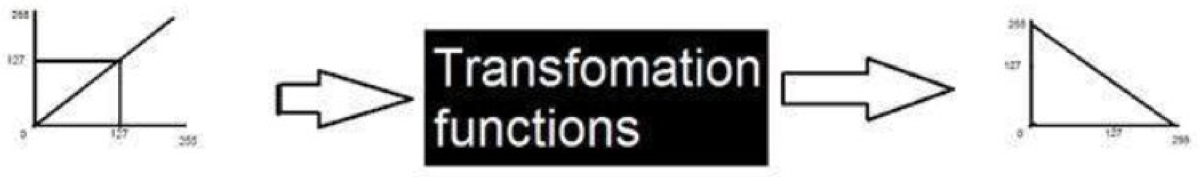
Convolution vs. Transformation

Graphs (Histograms)

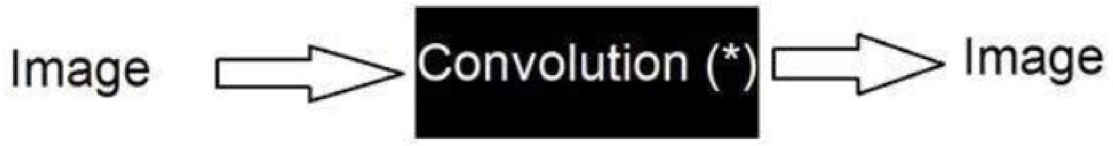


This method is known as histogram processing. We have discussed it in detail in previous tutorials for increase contrast, image enhancement, brightness e.t.c

Transformation functions



Convolution vs. Transformation



It can be mathematically represented as two ways

g(x,y) = h(x,y) * f(x,y)

It can be explained as the “mask convolved with an image”.

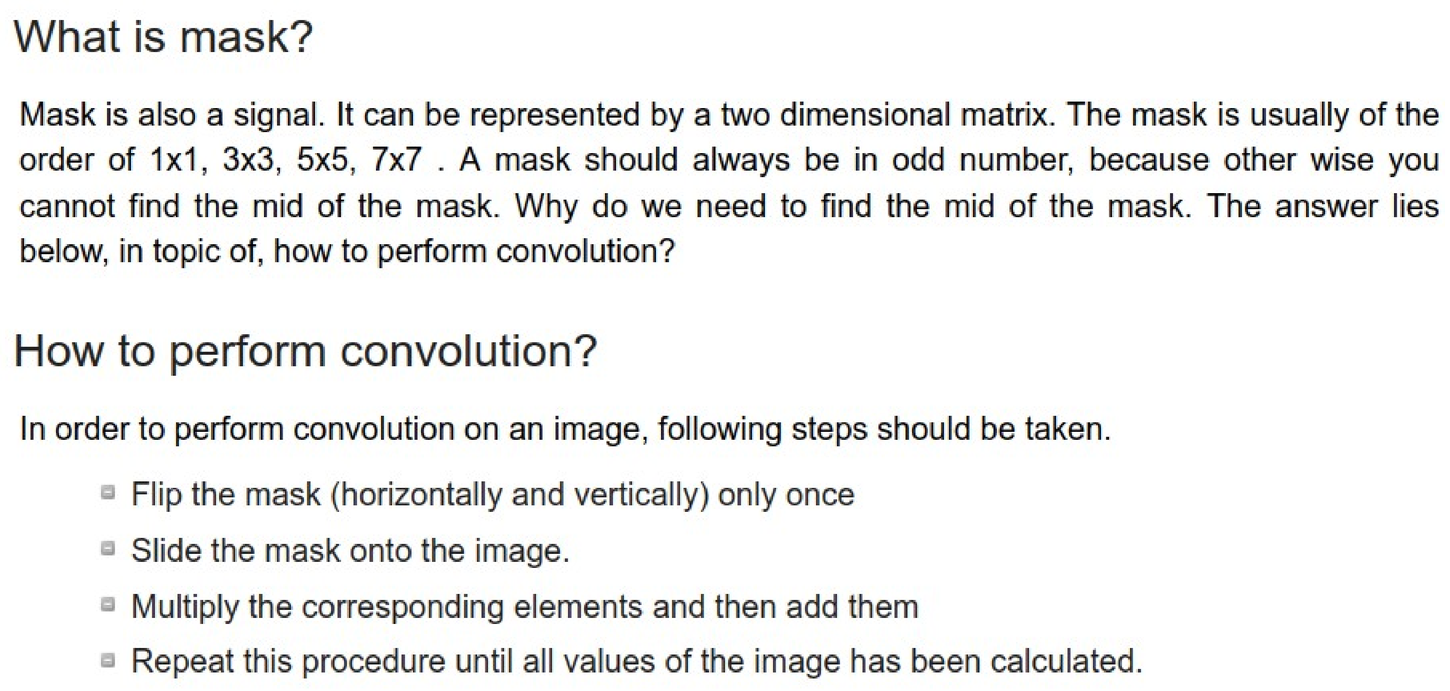
Or

g(x,y) = f(x,y) * h(x,y)

It can be explained as “image convolved with mask”.

https://www.tutorialspoint.com/dip

Convolution vs. Transformation



What is mask?

Mask is also a signal. It can be represented by a two dimensional matrix. The mask is usually of the order of 1x1, 3x3, 5x5, 7x7 . A mask should always be in odd number, because other wise you cannot find the mid of the mask. Why do we need to find the mid of the mask. The answer lies below, in topic of, how to perform convolution?

How to perform convolution?

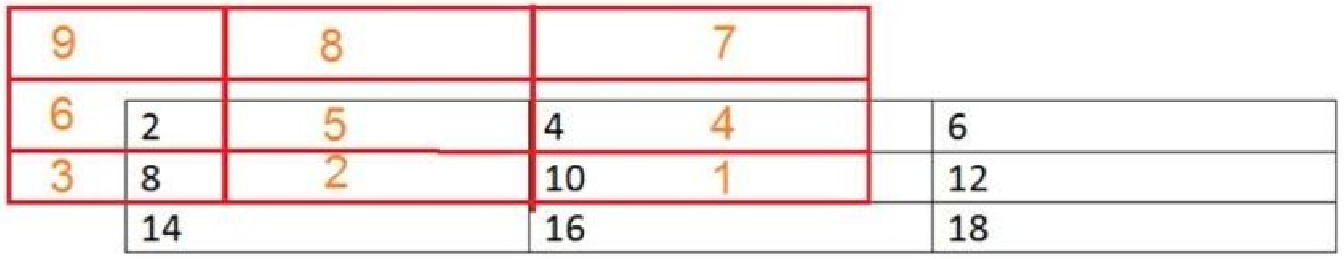
In order to perform convolution on an image, following steps should be taken.

- Flip the mask (horizontally and vertically) only once
- Slide the mask onto the image.
- Multiply the corresponding elements and then add them
- Repeat this procedure until all values of the image has been calculated.

Convolution vs. Transformation

Convolution

Convolving mask over image. It is done in this way. Place the center of the mask at each element of an image. Multiply the corresponding elements and then add them , and paste the result onto the element of the image on which you place the center of mask.



The box in red color is the mask, and the values in the orange are the values of the mask. The black color box and values belong to the image. Now for the first pixel of the image, the value will be calculated as

First pixel = (5*2) + (4*4) + (2*8) + (1*10)

= 10 + 16 + 16 + 10

= 52

Place 52 in the original image at the first index and repeat this procedure for each pixel of the image.

Convolution vs. Transformation

The use of filters in image processing means that the values of the surrounding points are taken into account to calculate the new value of a point. Each pixel from its surroundings contributes its own weight when the calculation is performed. These weights are stored in the form of a mask. Typical mask sizes are 3 x 3, 5 x 5 or 7 x 7. Mask sizes are usually odd because the pixel in the middle represents the pixel for which the filter transformation operation is performed. Let's analyze filtering based on a filter with a mask of 3 x 3.

$f_{-1,-1}$	$f_{0,-1}$	$f_{1,-1}$
$f_{-1,0}$	$f_{0,0}$	$f_{1,0}$
$f_{-1,1}$	$f_{0,1}$	$f_{1,1}$

Then, the new value of the component of point a with coordinates (i, j) will be calculated according to the following formula. First, we calculate the weighted sum of the point component and all its neighbors according to the weights indicated by the filter mask

$$s = f_{-1,-1} \cdot a_{i-1,j-1} + f_{0,-1} \cdot a_{i,j-1} + f_{1,-1} \cdot a_{i+1,j-1} +$$

$$f_{-1,0} \cdot a_{i-1,j} + f_{0,0} \cdot a_{i,j} + f_{1,0} \cdot a_{i+1,j} +$$

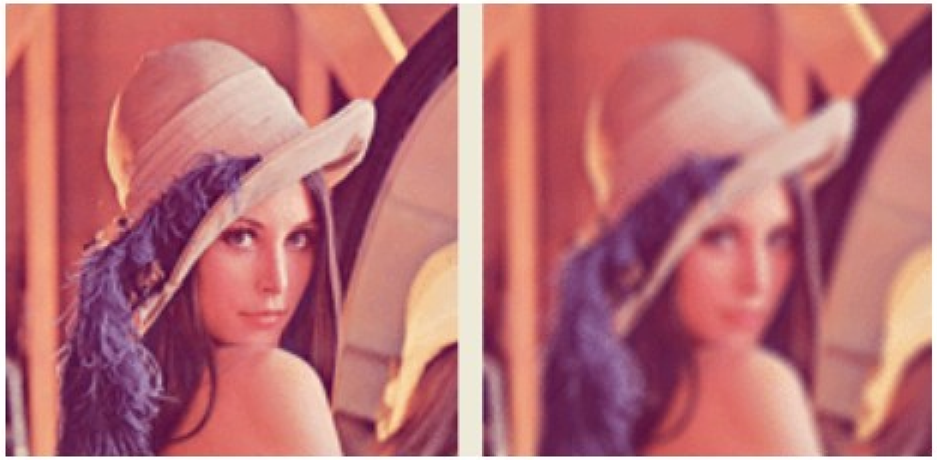
$$f_{-1,1} \cdot a_{i-1,j+1} + f_{0,1} \cdot a_{i,j+1} + f_{1,1} \cdot a_{i+1,j+1}$$

The sum thus obtained is divided by the sum of all mask weights if it is different from 0. This process of normalizing the point component value will prevent the brightness of the processed image from changing.

$$a'_{i,j} = \frac{s}{f_{-1,-1} + f_{0,-1} + f_{1,-1} + f_{-1,0} + f_{0,0} + f_{1,0} + f_{-1,1} + f_{0,1} + f_{1,1}}$$

Convolution vs. Transformation

Low-pass filters allow to pass low frequency elements of the image. High frequency elements (noise, fine details) are attenuated or even blocked. The result of such filters is reduction of noise, especially when it is one or two pixels, but also smoothing and blurring of the image. Below there is an example image before (on the left) and after applying the low-pass filter:



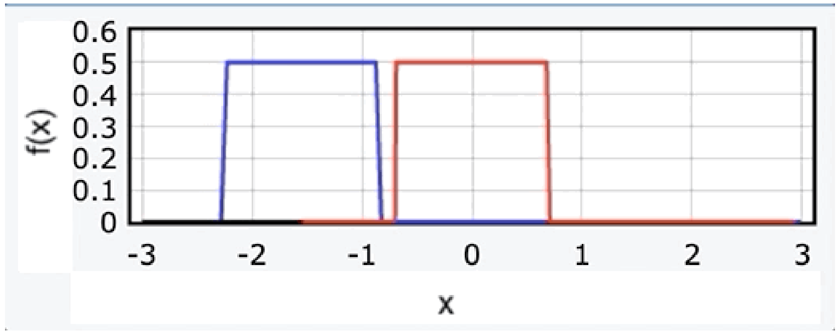
1	1	1
1	1	1
1	1	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Averaging filter - is a basic low pass filter, its result is to average each pixel together with its eight neighbors.

After http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html

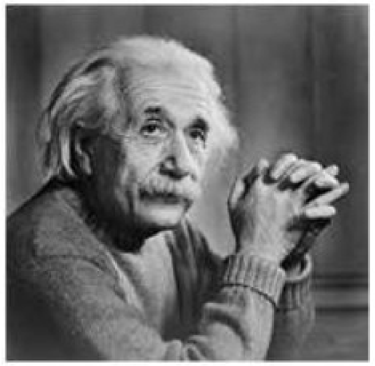
Convolution vs. Transformation



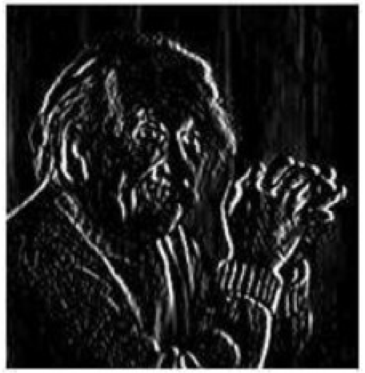
https://medium.com/swlh/calculating-the-convolution-of-two-functions-with-python-8944e56f5664

Convolution vs. Transformation

Original picture

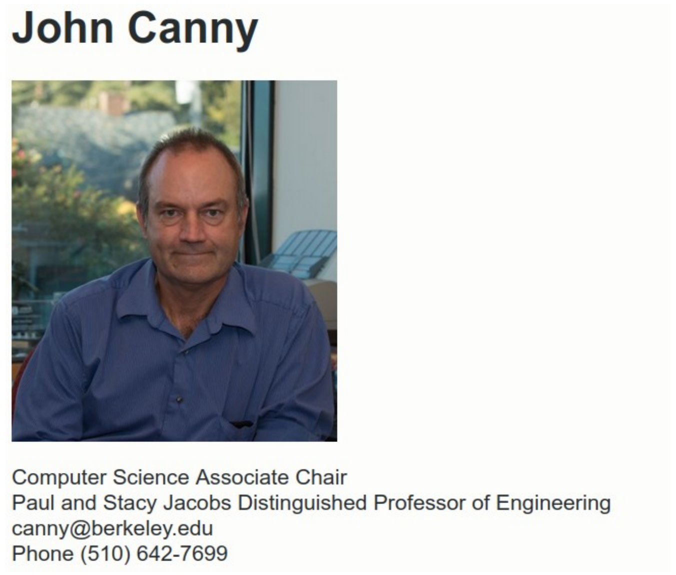


Same picture with edges



Canny edge detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.



John Canny

Computer Science Associate Chair
Paul and Stacy Jacobs Distinguished Professor of Engineering
canny@berkeley.edu
Phone (510) 642-7699

https://people.eecs.berkeley.edu/~jfc/

General criteria

The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

Workflow

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise (noise reduction)
2. Find the intensity gradients of the image
3. Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

https://en.wikipedia.org/wiki/Canny_edge_detector

Apply Gaussian filter to smooth the image in order to remove the noise (noise reduction)

A particular class of low pass filters are Gaussian filters, the coefficients of which are an aproximation of the two-dimensional function of Gaussian:

$$f(x,y) \sim e^{-\frac{x^2+y^2}{2\sigma^2}}$$

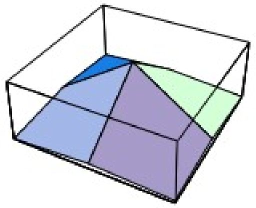
Gaussian filters are symmetric filters where the factor corresponding to the central element is the highest weight, and the coefficients have a lower value than the central element.
The size of the Gaussian filter mask should be selected depending on th value σ(variance) parameter of the Gaussian function. Usually, a mask of size 6σx 6σ is taken.

Apply Gaussian filter to smooth the image in order to remove the noise (noise reduction)

$$f(x,y) \sim e^{-\frac{x^2+y^2}{2\sigma^2}}$$

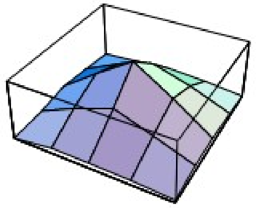
a

$$\begin{pmatrix} 1 & 4 & 1 \\ 4 & 32 & 4 \\ 1 & 4 & 1 \end{pmatrix}$$



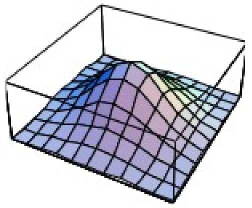
b

$$\begin{pmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 12 & 19 & 12 & 3 \\ 4 & 19 & 32 & 19 & 4 \\ 3 & 12 & 19 & 12 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{pmatrix}$$

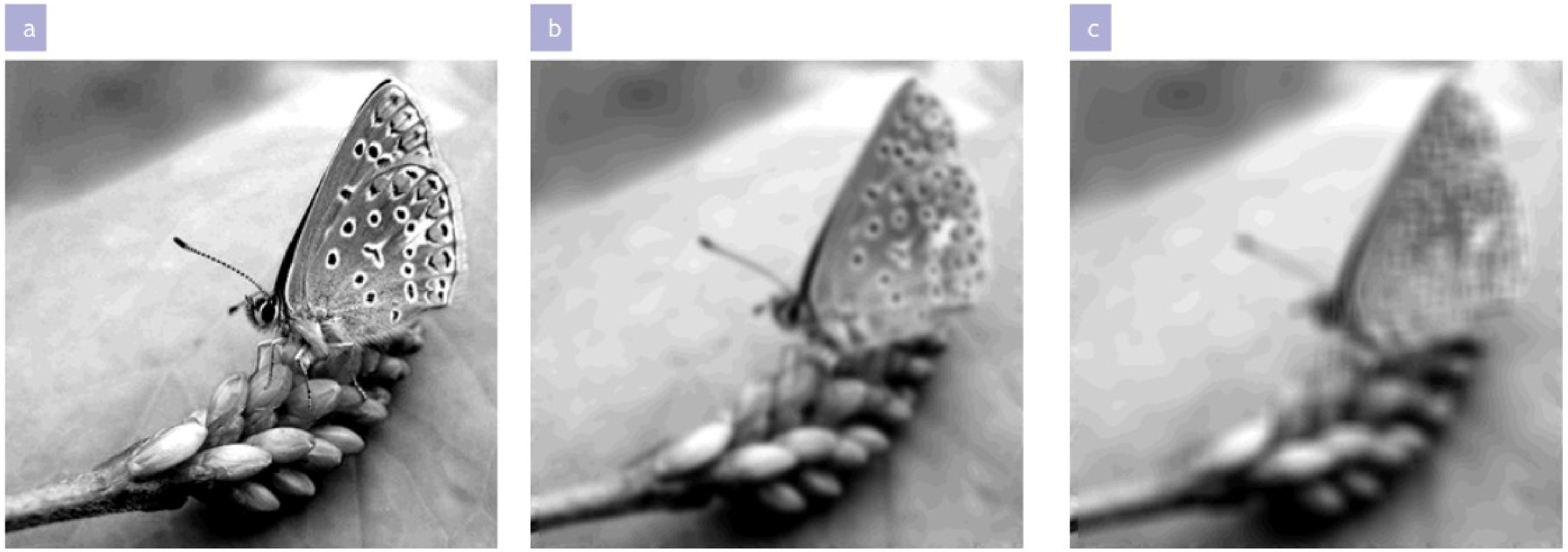


c

$$\begin{pmatrix} 0 & 0 & 1 & 2 & 2 & 3 & 2 & 2 & 1 & 0 & 0 \\ 0 & 1 & 3 & 5 & 8 & 9 & 8 & 5 & 3 & 1 & 0 \\ 1 & 3 & 7 & 13 & 18 & 21 & 18 & 13 & 7 & 3 & 1 \\ 2 & 5 & 13 & 24 & 34 & 39 & 34 & 24 & 13 & 5 & 2 \\ 2 & 8 & 18 & 34 & 50 & 56 & 50 & 34 & 18 & 8 & 2 \\ 3 & 9 & 21 & 39 & 56 & 64 & 56 & 39 & 21 & 9 & 3 \\ 2 & 8 & 18 & 34 & 50 & 56 & 50 & 34 & 18 & 8 & 2 \\ 2 & 5 & 13 & 24 & 34 & 39 & 34 & 24 & 13 & 5 & 2 \\ 1 & 3 & 7 & 13 & 18 & 21 & 18 & 13 & 7 & 3 & 1 \\ 0 & 1 & 3 & 5 & 8 & 9 & 8 & 5 & 3 & 1 & 0 \\ 0 & 0 & 1 & 2 & 2 & 3 & 2 & 2 & 1 & 0 & 0 \end{pmatrix}$$



Apply Gaussian filter to smooth the image in order to remove the noise (noise reduction)



Original Gaussian Simple low-pass

Apply Gaussian filter to smooth the image in order to remove the noise (noise reduction)

Since all edge detection results are easily affected by the noise in the image, it is essential to filter out the noise to prevent false detection caused by it. To smooth the image, a Gaussian filter kernel is convolved with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector. The equation for a Gaussian filter kernel of size (2k+1)×(2k+1) is given by:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \le i, j \le (2k+1)$$

Here is an example of a 5×5 Gaussian filter, used to create the adjacent image, with σ = 1.

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

It is important to understand that the selection of the size of the Gaussian kernel will affect the performance of the detector. The larger the size is, the lower the detector's sensitivity to noise.

https://en.wikipedia.org/wiki/Canny_edge_detector

Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (such as Roberts, Prewitt, or Sobel) returns a value for the first derivative in the horizontal direction (Gx) and the vertical direction (Gy). From this the edge gradient and direction can be determined:

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x),$$

where G can be computed using the hypot function and atan2 is the arctangent function with two arguments. The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0°, 45°, 90° and 135°). An edge direction falling in each color region will be set to a specific angle values, for instance θ in [0°, 22.5°] or [157.5°, 180°] maps to 0°.

https://en.wikipedia.org/wiki/Canny_edge_detector

Gradient magnitude thresholding or lower bound cut-off suppression

Minimum cut-off suppression of gradient magnitudes, or lower bound thresholding, is an edge thinning technique.
Lower bound cut-off suppression is applied to find the locations with the sharpest change of intensity value. The algorithm for each pixel in the gradient image is:

1.Compare the edge strength of the current pixel with the edge stre
ngth of the pixel in the positive and negative gradient directions.

2. If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction (e.g., a pixel that is pointing in the y-direction will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

https://en.wikipedia.org/wiki/Canny_edge_detector

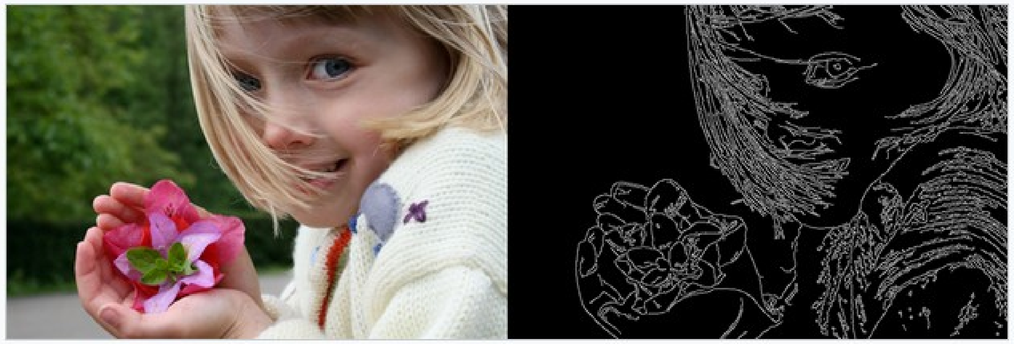
Double threshold

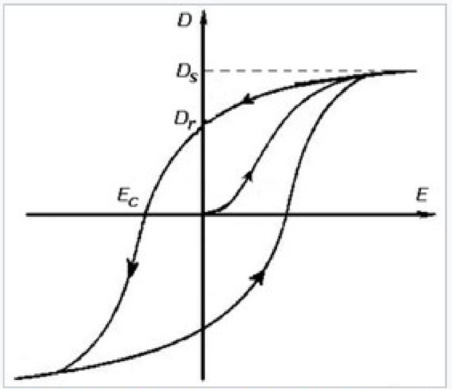
After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. If an edge pixel’s gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel’s gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's gradient value is smaller than the low threshold value, it will be suppressed. The two threshold values are empirically determined and their definition will depend on the content of a given input image.

https://en.wikipedia.org/wiki/Canny_edge_detector

Edge tracking by hysteresis

To achieve an accurate result, the weak edges caused by the latter reasons should be removed. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels.


Canny edge detection applied to a photograph



Electric displacement field D of a ferroelectric material as the electric field E is first decreased, then increased. The curves form a *hysteresis loop*.

https://en.wikipedia.org/wiki/Canny_edge_detector

https://en.wikipedia.org/wiki/Hysteresis