Module 5

Image Enhancement

> **Image Enhancement**
>
> The process of changing the pixel values of an image in order to get more detail/information, usually enhancing an image provides better contrast and a more detailed image as compared to the original.

Basic Gray Level Transformation

Transformations on an image can be generalized in the following format

$$S = T(r)$$

where S is the corresponding pixel in the output image after the transformation T is applied on the original image pixel r.

There are 3 basic gray level transformations, they are:

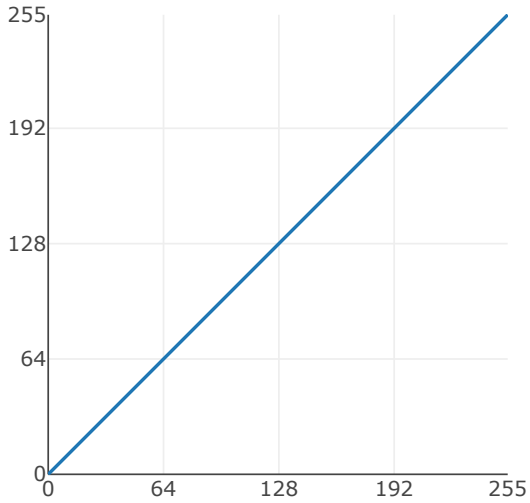
- Linear Transformation
- Log Transformation
- Power-Law Transformation

Linear Transformation

Linear transformations include simple operation that are applied to the input image to produce the output image.

Identity Transformation
This is a very simple function that just return the same pixel value without any change.

$$S = r$$



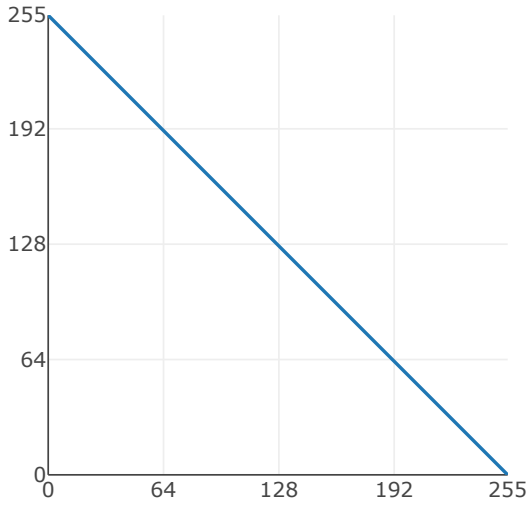
Negative Transformation
The transformation converts an image to its **"negative"** , which is the flipping of all the pixel values, for example white will become black and black will become white vise-versa.

$$S = (L - 1) - r$$

Here L is the maximum number of levels the image has, for an 8-bit image this will be $2^8 = 256$, so for an 8-bit image out equation

becomes

$$S = 255 - r$$



Logarithmic Transformation

The log transformation are processes of applying a log function to the input pixels. There are mainly two types of log transformations:

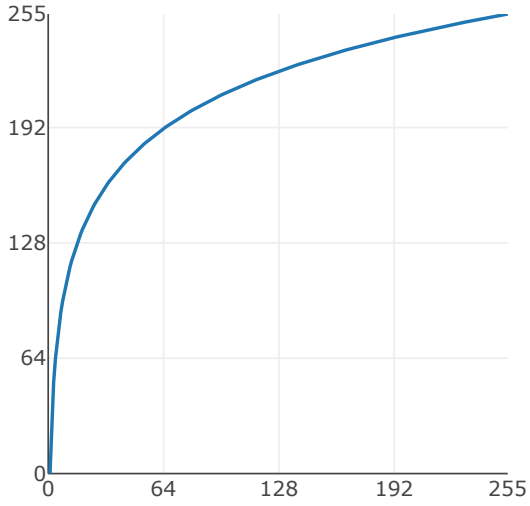
Log Transformation
By applying the transformation which can be represented as:

$$S = c \cdot log(r + 1)$$

Points To Note

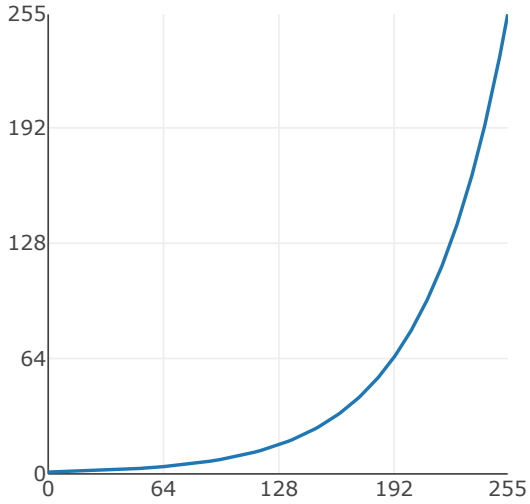
- c is an arbitrary constant
- The input is $r + 1$ rather than just r to avoid issues that may arise due to $r = 0$ like $log(0)$

The output image results to be a **high contrast image**, this is because the log transformation enhances image contrast by spreading out darker pixel values to make their differences more noticeable while compressing brighter pixel values to reduce their differences.



Inverse Log Transformation
This is the inverse of the normal log function here, the higher input pixel value, lower output pixel value and as a result a **low contrast output image** is produced.

$$S = c \cdot log^{-1}(r+1)$$



Power Law Transformation

A technique used in computer graphics to enhance the contrast of an image by applying a non-linear transformation to the pixel values.

The transformation is represented as:

$$S = c \cdot r^{\gamma}$$

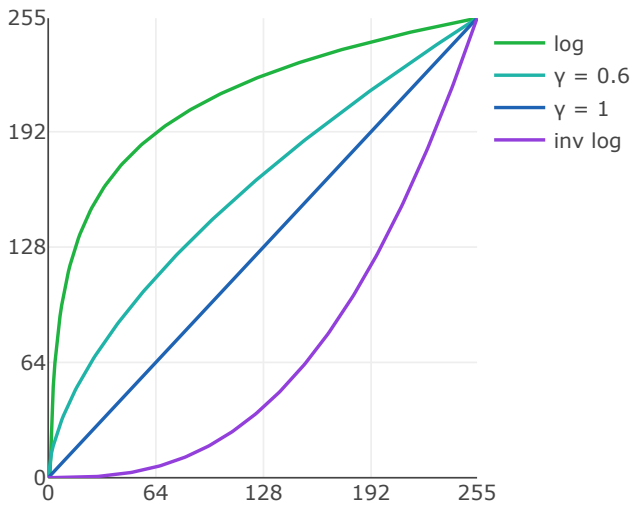
Points To Note

- The parameter γ controls the degree of enhancement.
- c is a constant that scales the output.

The output image results to be either brighter or darker depending on the value of γ . This is because when $\gamma < 1$, the transformation enhances the brightness of the image, making dark regions lighter. Conversely, when $\gamma > 1$, it enhances the darkness, making bright regions darker.

Gamma Correction

The power law is similar to the log function, but since the value of γ is variable, the power law function is much more versatile and is often used to correct contrast for displays, this process is called gamma correction.



Contrast Stretching

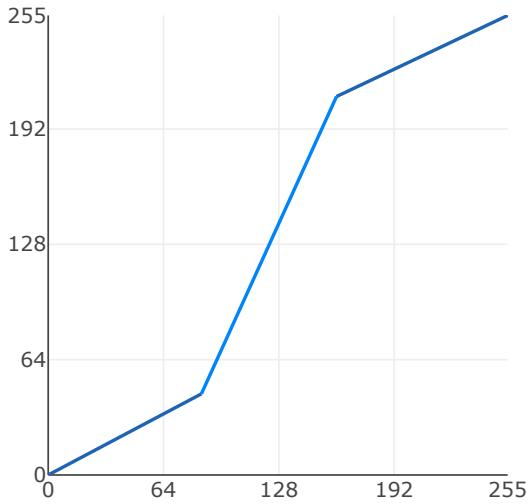
Piece-wise Linear Transformation

Rather than using a well-defined function, we use arbitrary user defined transforms.

Contrast Stretching is a piece-wise linear transformation, it is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

An example for a contrast stretching functions is:

$$S(r) = \begin{cases} x \cdot r & \text{if } 0 \leq r \leq a \\ y \cdot (r-a) + c & \text{if } a < r \leq b \\ z \cdot (r-b) + d & \text{if } b < r \leq L-1 \end{cases}$$

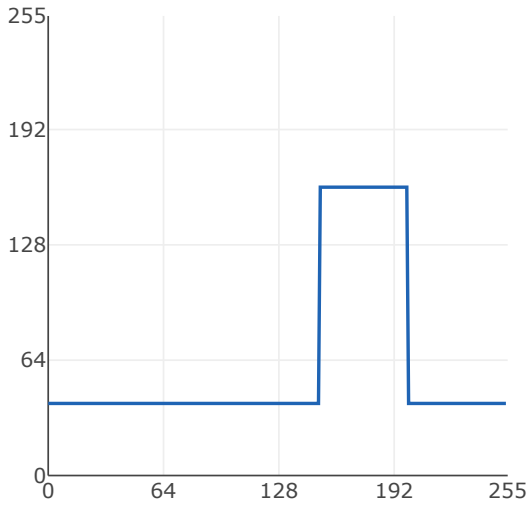


Intensity Level slicing

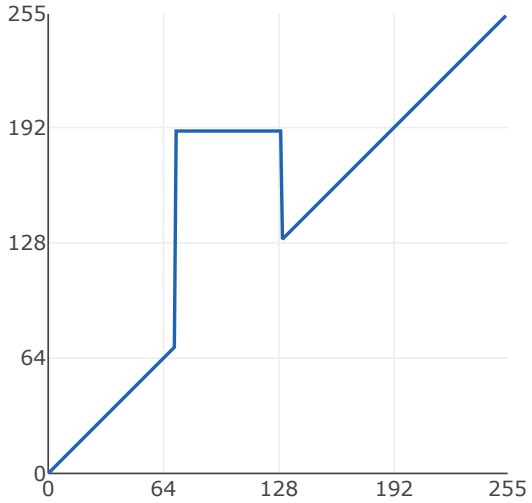
This technique is used to highlight a specific range of gray levels in a given image (thresholding). Other levels can be suppressed or maintained – Useful for highlighting features in an image.
Two basic themes are:

- One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.
- The second approach, based on the transformation brightens the desired range of gray levels but preserves gray levels unchanged.

Approach 1



Approach 2



Histogram Processing

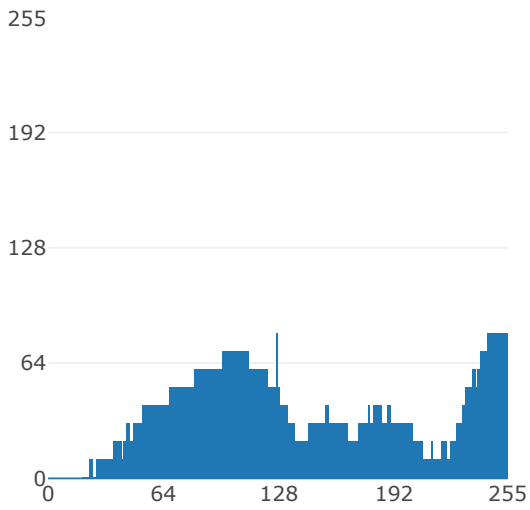
Histogram

A graphical representation that displays the distribution of numerical data through bins or intervals, showcasing the frequency of data points within those ranges.

Applications of Histogram

It is used to analyze an image. Properties of an image can be predicted by the detailed study of the histogram.

- The brightness of the image can be adjusted by having the details of its histogram.
  The contrast of the image can be adjusted according to the need by having details of the x-axis of a histogram. It is used for image equalization. Gray level intensities are expanded along the x-axis to produce a high contrast image. Histograms are used in thresholding as it improves the appearance of the image. If we have input and output histogram of an image, we can determine which type of transformation is applied in the algorithm



Histogram Equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images. Histogram equalization is used for equalizing all the pixel values of an image, so that a uniformly flattened histogram is produced.
#Todo

Spatial Filtering

Spatial filtering is a technique used in image processing to modify or enhance an image by manipulating the pixel values based on the values of neighboring pixels. This process involves the application of a filter mask (also known as a kernel, template, or window) that moves across the image, performing operations at each pixel location. Histogram equalization increases the dynamic range of pixel values and makes an equal count of pixels at each level which produces a flat histogram with high contrast image.

Points To Note

- Spatial filtering can be linear or nonlinear.
- It is used for various purposes such as noise reduction, blurring, sharpening, and edge detection.

Smoothing Linear Filters

Smoothing filters are used to reduce noise and smooth out rapid intensity variations in an image. They are also known as low-pass filters because they allow low-frequency components to pass through while attenuating high-frequency components.

Average Filter:
The average filter, also known as the mean filter, replaces each pixel value with the average of the pixel values in its neighborhood. This filter is effective in reducing random noise. This is also referred to as lowpass filters.

The equation for a 3×3 average filter is:

$$\boldsymbol{x} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

or

$$g(x,y) = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} f(x+i, y+j)$$

Median Filter:
The median filter is a nonlinear filter that replaces each pixel value with the median value of the pixel values in its neighborhood. It is particularly effective in removing salt-and-pepper noise while preserving edges.

The process involves:

1. Sorting the pixel values in the neighborhood.
2. Selecting the median value.
3. Replacing the center pixel with the median value.

Max/Min Filters:
Max and Min filters are order-statistics filters that replace each pixel value with the maximum or minimum value in its neighborhood, respectively. These filters are useful for enhancing bright or dark regions in an image.

Sharpening Linear Filters

Sharpening filters are used to enhance the edges and fine details in an image. They are also known as high-pass filters because they allow high-frequency components to pass through while attenuating low-frequency components.

First Derivative:
The first derivative of an image is used to detect edges by highlighting regions with rapid intensity changes. The gradient magnitude is commonly used to measure the strength of edges.

The gradient magnitude can be computed using the following equation:

$$\Delta f = |Gx| + |Gy|$$
$$\nabla = |Gx| + |Gy|$$

where Gx and Gy are the gradients in the x and y directions, respectively.

Second Derivative:
The second derivative is more sensitive to fine details and is used to enhance edges and other discontinuities. The Laplacian operator is a common second-order derivative used in image sharpening.

The Laplacian operator is defined as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

In discrete form, the Laplacian can be approximated using a convolution mask:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

By applying the Laplacian to an image and then subtracting the result from the original image, we can enhance the edges and fine details.

Example of a Laplacian Filter:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Fundamentals of Image Segmentation

> **Image Segmentation**
>
> Image Segmentation is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analyzing the image becomes simpler.

Similarity Detection

This fundamental approach relies on detecting similar pixels in an image – based on a threshold, region growing, region spreading, and region merging so does classification, which detects similarity based on a pre-defined (known) set of features.

Discontinuity Detection

This is a stark opposite of the similarity detection approach where the algorithm rather searches for discontinuity. Image Segmentation Algorithms like Edge Detection, Point Detection, Line Detection follow this approach – where edges get detected based on various metrics of discontinuity like intensity, etc.

Thresholding

Simple thresholding (Binary Thresholding)

$$g(x,y) = \begin{cases} 1, \text{ if } f(x,y) > T \\ 0, \text{ else } f(x,y) < T \end{cases}$$

this is a simple binary thresholding function which converts the pixels into either black or white depending on the Threshold T

Multiple thresholding

$$g(x,y) = \begin{cases} a, \text{ if } f(x,y) > T_1 \\ b, \text{ if } T_1 < f(x,y) < T_2 \\ c, \text{ if } f(x,y) < T_2 \end{cases}$$

Here the function converts the output pixels to either a, b or c depending on the thresholds T_1 and T_2.

Types of thresholding

- **Global thresholding**: T is constant and applicable over the whole image
- **Variable/ Local thresholding**: T changes over an image. T at a point (x,y) is a function of the neighborhood of (x,y)
- **Dynamic / Adaptic thresholding**: T changes over an image. T at any point(x,y) is a function of spatial coordinate (x,y)

Region Growing Technique

In the case of the Region growing method, we start with some pixel as the seed pixel and then check the adjacent pixels.
If the adjacent pixels abide by the predefined rules, then that pixel is added to the region of the seed pixel and the following process continues till there is no similarity left. This method follows the bottom-up approach.
In case of a region growing, the preferred rule can be set as a threshold.

Region Splitting and Merging Technique

In Region splitting, the whole image is first taken as a single region. If the region does not follow the predefined rules, then it is further divided into multiple regions (usually 4 quadrants) and then the predefined rules are carried out on those regions in order to decide whether to further subdivide or to classify that as a region. The following process continues till there is no further division of regions required i.e every region follows the predefined rules. In Region merging technique, we consider every pixel as an individual region. We select a region as the seed region to check if adjacent regions are similarly based on predefined rules. If they are similar, we merge them into a single region and move ahead in order to build the segmented regions of the whole image.

Edge Detection

Edge detection

Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image.

Sobel Edge Detection

The Sobel edge detection operator extracts all the edges of an image, without worrying about the directions. The main advantage of the Sobel operator is that it provides differencing and smoothing effect.

Sobel edge detection operator is implemented as the sum of two directional edges. And the resulting image is a unidirectional outline in the original image. Sobel Edge detection operator consists of 3×3 convolution kernels. G_x is a simple kernel and G_y is rotated by 90°.

$$\begin{bmatrix} -1 & 0 & +1 \\ -5 & 0 & +5 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} +1 & +5 & +1 \\ 0 & 0 & 0 \\ -1 & -5 & -1 \end{bmatrix}$$

These Kernels are applied separately to the input image

Prewitt Edge Detection

The Prewitt edge detection operator extracts all the edges of an image, without worrying about the directions. This is similar to Sobel, just that all the values in the kernel are unit values.

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Robert's Cross Operator

The Roberts cross operator, a simple and quick differential method for edge detection, approximates the gradient of an image by summing the squares of differences between diagonally adjacent pixels, aiming to produce well-defined edges with minimal background noise and edge intensities that closely match human perception.

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$