Intensity Transformation and Spatial Filtering

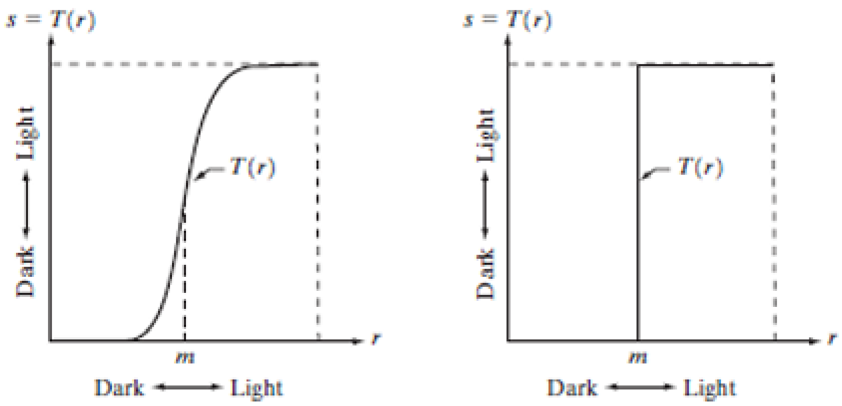
Intensity transformations modify an image's pixel brightness and contrast by directly altering the intensity values of individual pixels, not their spatial relationships, using functions like Negative Transformations, Logarithmic Transformations, Power-Law Transformations (Gamma Correction), Contrast Stretching, and Thresholding.

These point processing techniques enhance details, adjust exposure, and create binary images by mapping input pixel intensities to output intensities based on their value.

The general form of intensity transformation will be

$$s = T(r)$$

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‟.



For example, if T(r) has the form shown in Fig. (a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation function into a narrow range of s, toward black. The opposite effect takes place for values of r above m. In the limiting case shown in Fig. (b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.

Basic Gray Level Transformation / Intensity Transformation:

1) Linear Transformation:

Linear transformation includes simple identity and negative transformation. Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called **identity transformation**. It has been shown below:

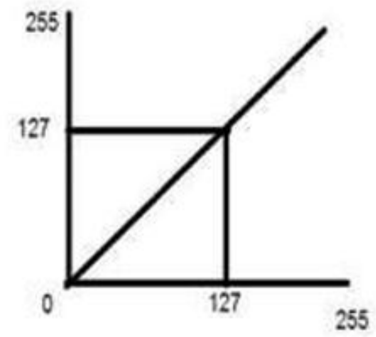


Fig. Linear transformation between input and output

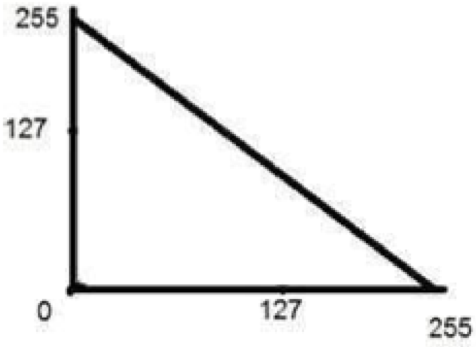
Negative Transformation:

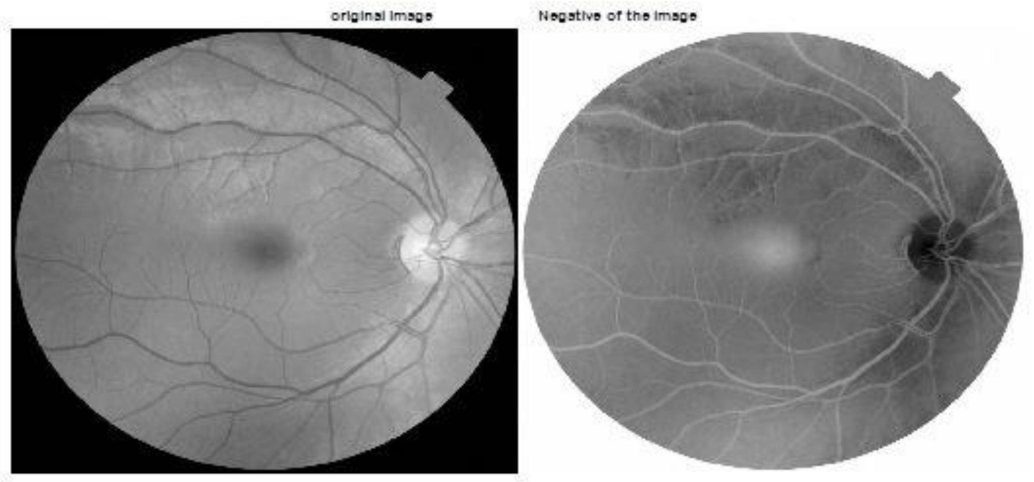
The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.

IMAGE NEGATIVE: The image negative with gray level value in the range of [0, L-1] is obtained by negative transformation given by S = T(r) or S = L -1 – r
Where r= gray level value at pixel (x,y)

L is the largest gray level consists in the image.

It results in getting photograph negative. It is useful when for enhancing white details embedded in dark regions of the image.





2) LOGARITHMIC TRANSFORMATIONS:

Logarithmic transformation further contains two types of transformation. Log transformation and inverse log transformation.

LOG TRANSFORMATIONS:

The log transformations can be defined by this formula:

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

Where S and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So, 1 is added, to make the minimum value at least 1. During log transformation, the dark pixels in an image are expanded as compared to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.

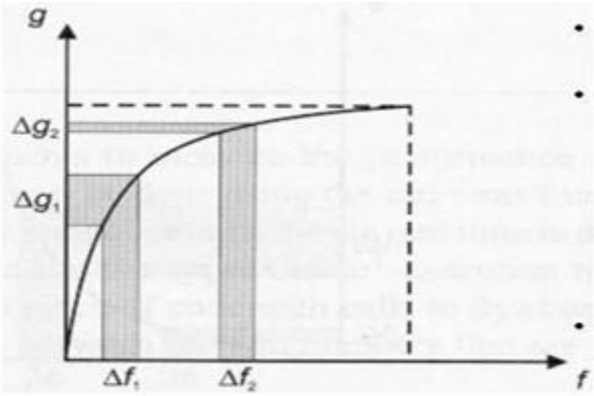
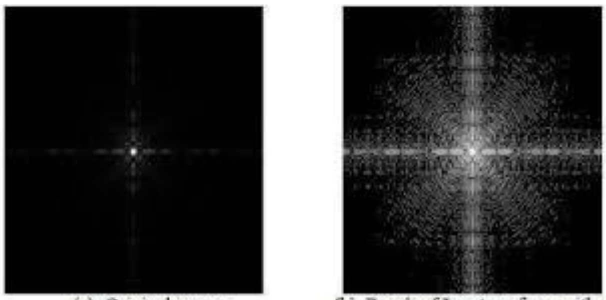


Fig. Log Transformation Curve input vs output

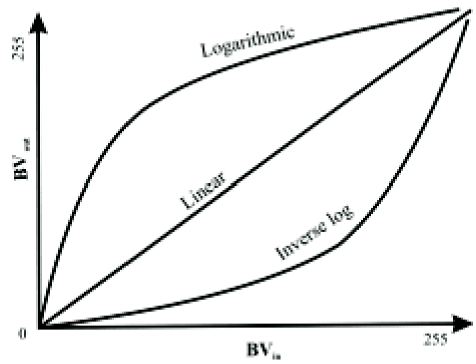


(a) Original image (b) Result of Log transform with $c = 1$

INVERSE LOG TRANSFORMATIONS:

The inverse log transformation, also known as the **exponential transformation**, is a gray-level transformation applied in image processing. It is the opposite of the log transformation and is used to enhance images by modifying their intensity levels.

$$s = c * (\exp(r) - 1)$$



The exponential function causes a non-linear mapping of input intensity values to output intensity values. Lower input pixel values will result in relatively smaller changes in output values, while higher input pixel values will experience a more significant increase, thus expanding the brighter regions of the image.

Purpose:

- **Expands bright regions and compresses dark regions:**

Unlike the log transformation which expands dark pixel values, the inverse log transformation expands the values of brighter pixels while compressing the values of darker pixels.

- **Darkens images:**

It is particularly useful for images that are too light or have a wide range of bright intensity levels that need to be compressed for better visualization.

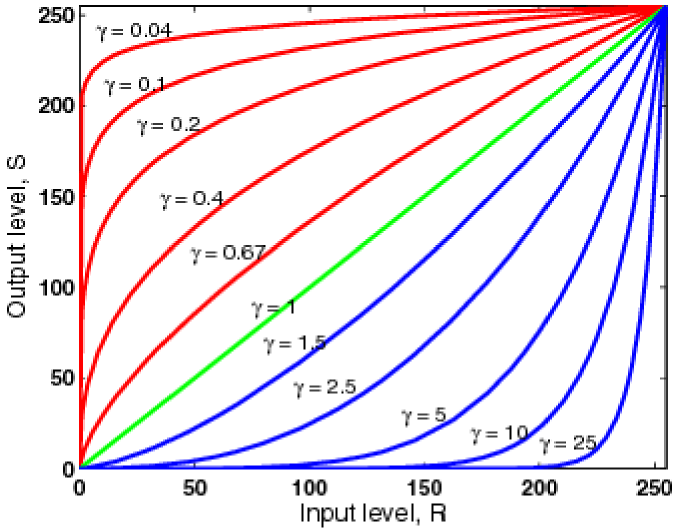
3) POWER – LAW (GAMMA) TRANSFORMATIONS:

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

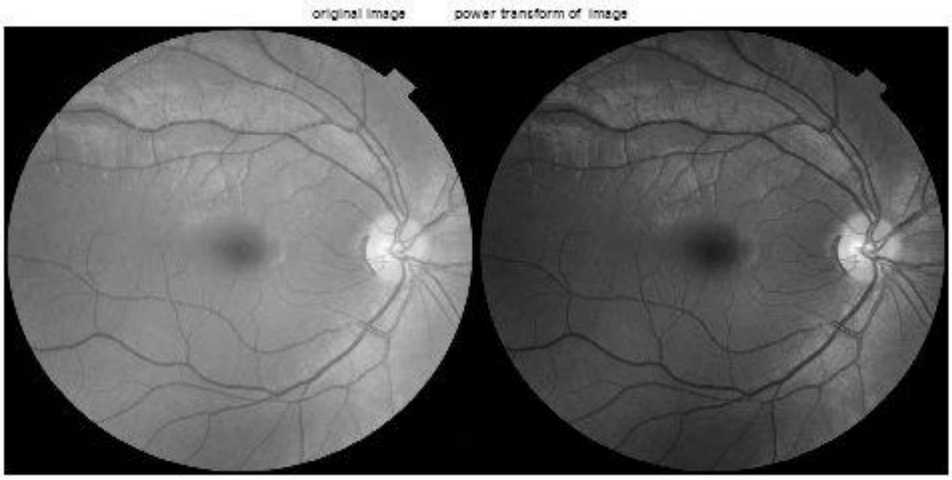
$$s = c \cdot r^{\gamma}$$

How it Works

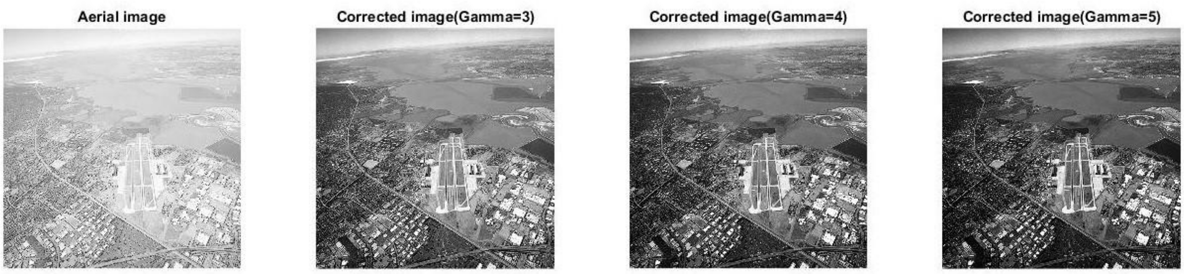
1. **Input and Output:**
   Each pixel's input intensity (**r**) is processed individually.
2. **The Formula:**
   The input intensity is raised to the power of γ, multiplied by a constant **c**, to produce the new output intensity (**s**).
3. **Gamma Value (γ):**
   - **γ < 1:** Maps a smaller range of low input intensities to a larger range of output levels, making dark areas appear brighter.
   - **γ > 1:** Does the opposite, mapping a larger range of low input intensities to a smaller range, making bright areas more prominent.
   - **γ = 1:** Results in an identity transform, meaning the output is the same as the input.



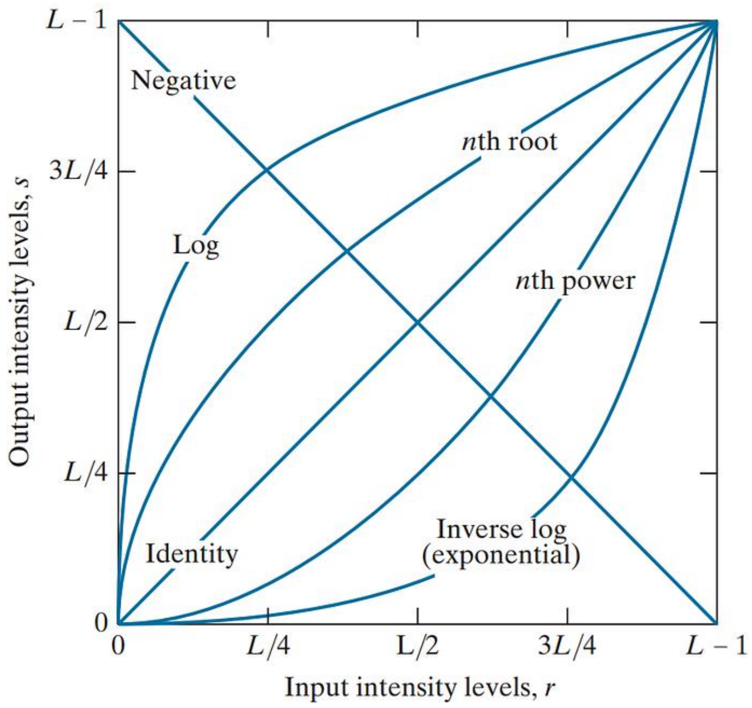
γ = 0.04
γ = 0.1
γ = 0.2
γ = 0.4
γ = 0.67
γ = 1
γ = 1.5
γ = 2.5
γ = 5
γ = 10
γ = 25
Output level, S
Input level, R



original image
power transform of image



Aerial image
Corrected image(Gamma=3)
Corrected image(Gamma=4)
Corrected image(Gamma=5)

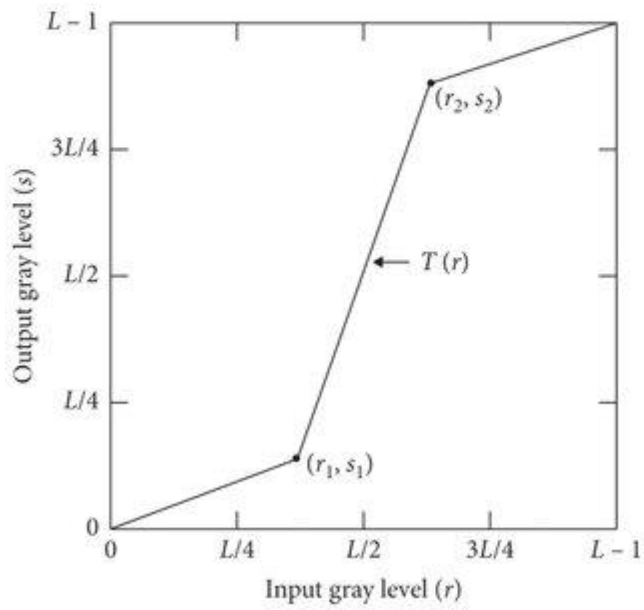


4) PIECEWISE-LINEAR TRANSFORMATION FUNCTIONS:

A piecewise linear transformation is a spatial domain image enhancement technique that manipulates pixel intensities by defining a transformation function as a series of connected linear segments, allowing for custom control over image contrast and brightness.

Common applications include contrast stretching, which expands the intensity range to use the full display capacity, and intensity-level slicing, which highlights or suppresses specific gray level ranges.

The transformation function is defined by a set of "break points" or control points that determine the shape and slope of each linear segment, enabling targeted image adjustments.



How it Works

1. **Define Break Points:**

The user selects key intensity values, or "break points," in the original image (e.g., r1, r2, r3, etc.).

2. **Assign Output Values:**

For each break point, an output value is specified (e.g., s1, s2, s3, etc.).

3. **Form Linear Segments:**

The function connects these points with straight lines, forming a piecewise linear transformation.

4. **Apply Transformation:**

Each pixel's original intensity r is mapped to a new intensity s by finding the corresponding point on the piecewise linear curve.

Common Applications

- **Contrast Stretching:**
- Contrast stretching is an image enhancement technique that improves the visual contrast of an image by expanding its range of intensity values to span a desired, often wider, range. This process is most effective for

images with a narrow range of intensity values, which typically appear dull or washed out.

The core idea is to map the original pixel values to a new set of values, using a transformation function, to enhance the visual distinction between features.

How contrast stretching works

In a typical 8-bit grayscale image, the range of possible intensity values is 0–255. If a low-contrast image has pixel values clustered in a narrow band—for example, between 60 and 158—a standard contrast stretch will redistribute these values across the full 0–255 range. This process makes the dark parts darker and the bright parts brighter, revealing details that were previously hidden.

The formula for a basic linear contrast stretch is:

$$S = \frac{(r - r_{min})}{(r_{max} - r_{min})} \times (S_{max} - S_{min}) + S_{min}$$

- r is the original pixel intensity.
- r_{min} and r_{max} are the minimum and maximum intensity values in the original image.
- S is the new, stretched pixel intensity.
- S_{min} and S_{max} are the minimum and maximum values of the desired output range, typically 0 and 255 for an 8-bit image.

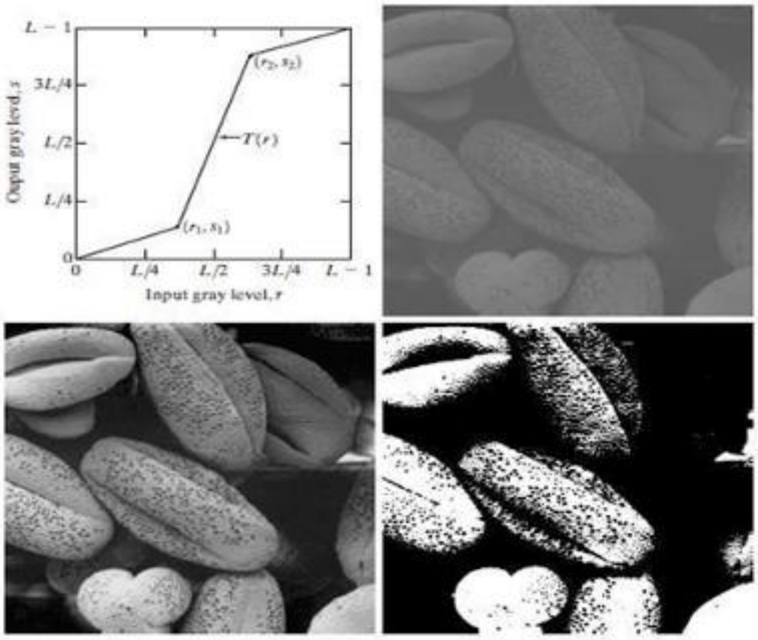


Fig. x Contrast Stretching.
(a) Form of transformation function
(b) A low-contrast stretching.
(c) Result of contrast stretching
(d) Result of thresholding

Linear contrast stretching

This is the simplest form, where the intensity values are expanded uniformly across the new range.

- **Simple linear stretch:** The minimum and maximum pixel values in the image are mapped to the minimum and maximum of the target range (e.g., 0 and 255).
- **Piece-wise linear stretch:** Allows for more targeted control over contrast. Multiple breakpoints can be set to stretch, compress, or clip different ranges of intensity values within the image. This is useful for images with bimodal or complex histograms.

Non-linear contrast stretching

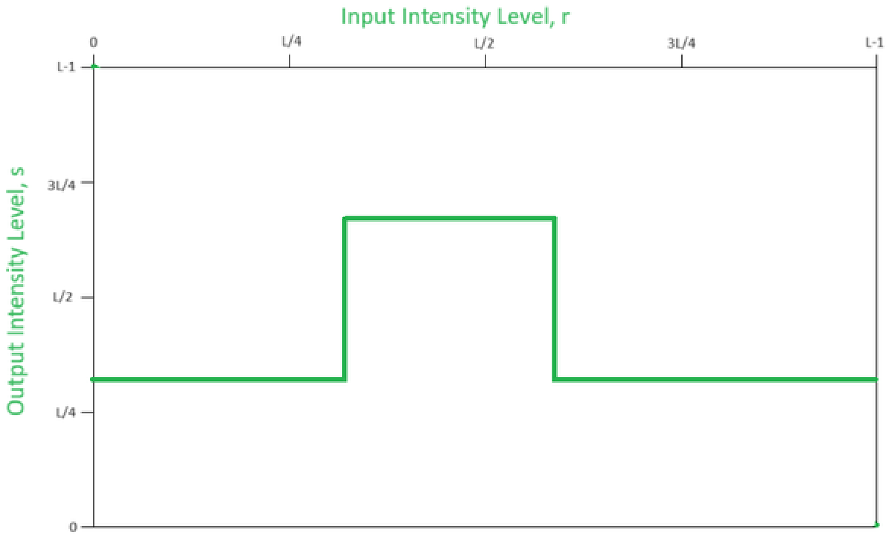
Unlike linear stretching, non-linear methods do not expand the intensity values uniformly. They use logarithmic, exponential, or other functions to redistribute the pixel values.

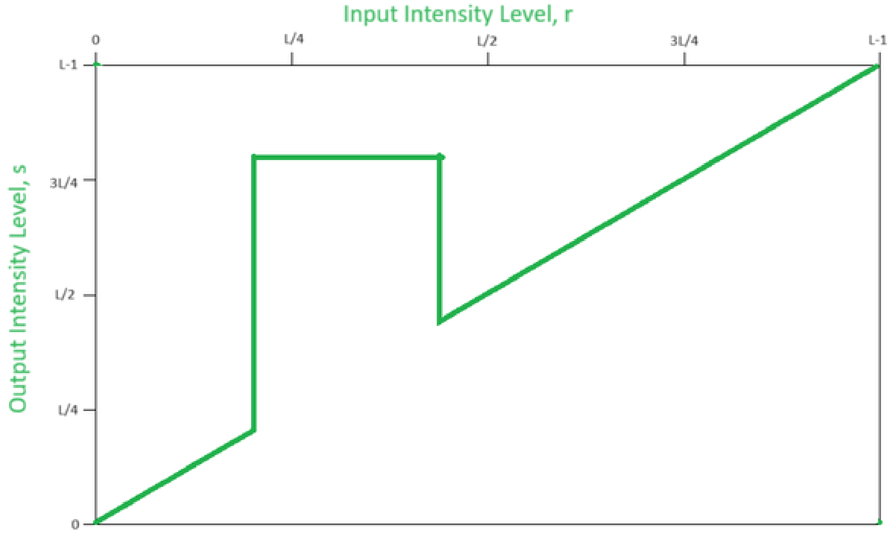
- **Logarithmic stretching:** Uses a logarithmic function to brighten dark pixels and compress the range of lighter pixels. This is ideal for enhancing details in the darker regions of an image.
- **Power-law stretching:** Uses a power-law function to expand the range of brighter pixels while compressing the darker ones, highlighting features in the lighter regions of an image.

- **Gray-Level Slicing:**
- Gray level slicing is an image processing technique that isolates and enhances a specific range of pixel intensity levels, effectively "slicing" through the grayscale values. This point processing method can either suppress other levels, displaying only the desired range as bright, or preserve the other levels while brightening the target range. It is used to highlight features in images like medical scans, satellite images, and for identifying specific structures based on their brightness.
- How it Works

  Gray level slicing involves transforming the pixel values of an image based on a defined range. For a range of gray levels, say from A to B:

- **Slicing without background:**
- Pixels with gray levels between A and B are set to a high value (often white), while all other pixels are set to a low value (often black). This completely removes the background, leaving only the target feature.
- **Slicing with background:**
- Pixels within the range A and B are brightened, but pixels outside this range are left at their original intensity values. This technique highlights the specific features of interest while still providing context from the rest of the image.





- **Applications**
- **Medical imaging:**

  Used in X-rays and CT scans to isolate and enhance specific abnormalities or tissues that fall within a particular gray level range.

- **Satellite imagery:**

  Can highlight specific features on the ground, such as cracks in road surfaces, by bringing them out of the background.

- **Industrial inspection:**

Useful for detecting flaws or variations in manufactured products that appear as distinct gray level patterns.

Purpose

- The primary purpose of gray level slicing is to improve the visibility of features in an image that may be hard to distinguish due to poor contrast. By manipulating the intensity of pixels in a specific range, it can make desired objects stand out, leading to better analysis and interpretation of the image.
- **Thresholding:**

A special case where a specific intensity level becomes a cutoff point, turning pixels below it to black and those above it to white, effectively creating a binary image.

- **Image Negatives:**

A special case where the transformation maps black to white and white to black

Key Characteristics

- **User-Defined:**

Requires user input to define the transformation function, offering flexibility.

- **Monotonicity:**

To prevent artifacts and preserve the relative order of intensities, the transformation is usually monotonic (increasing or decreasing), meaning the slopes are typically non-negative.

- **Efficiency:**

It's a simple and efficient spatial domain method for image enhancement directly manipulating pixel values

Image Histogram

In image processing, a histogram is a graphical representation of the intensity distribution of pixels within an image, showing the frequency of each pixel's intensity value. It helps analyze an image's brightness, contrast, and tonal distribution, enabling techniques like image enhancement, thresholding, and feature extraction.

Histograms can be calculated for grayscale images, where the x-axis represents intensity levels (e.g., 0-255) and the y-axis shows the number of pixels at that level, or for color images by analyzing individual color channels.

How it works

1. Pixel Counting:

The process involves scanning the image to count the number of pixels for each specific intensity value.

2. Graphing:

The collected counts are then plotted on a graph. The horizontal axis represents the intensity range, and the vertical axis represents the frequency (number of pixels) for each intensity level.

3. Normalization:

Histograms can also be normalized by dividing each count by the total number of pixels to represent the probability of occurrence for each intensity.

Example:

In Fig.1, we have a sample 5*5 image with pixel diversities from 0 to 4. In the first step for generating the histogram, we create the Histogram Table, by counting the number of each pixel intensities. Then we can easily generate the histogram by creating a bar chart based on the histogram table.

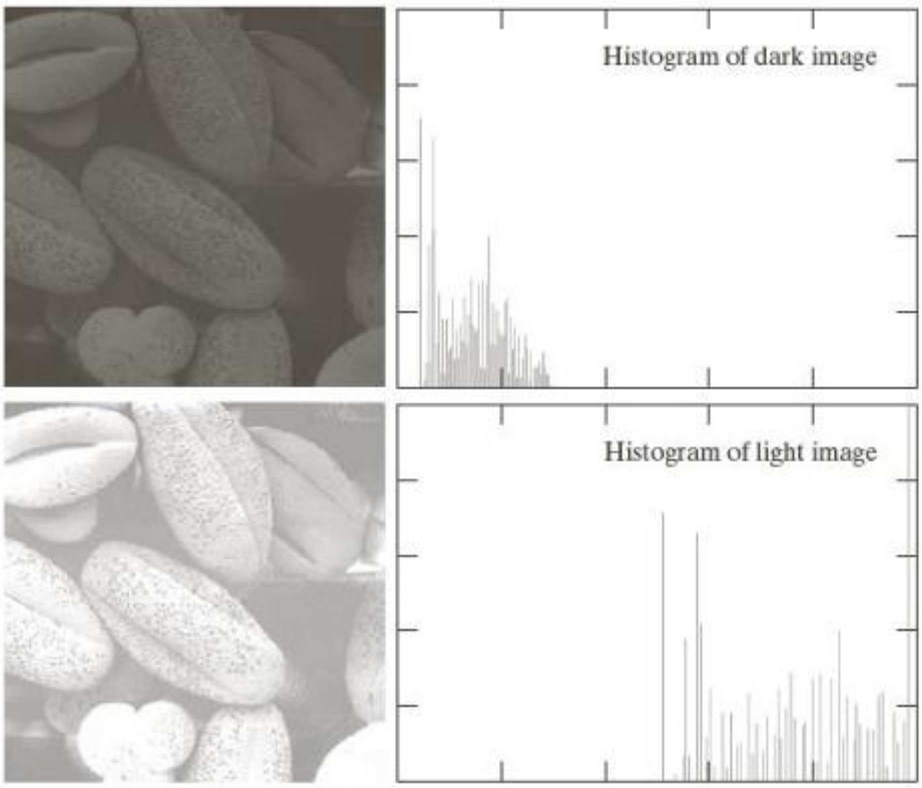
0	2	1	3	4
1	3	4	3	3
0	1	3	1	4
3	1	4	2	0
0	4	2	4	4

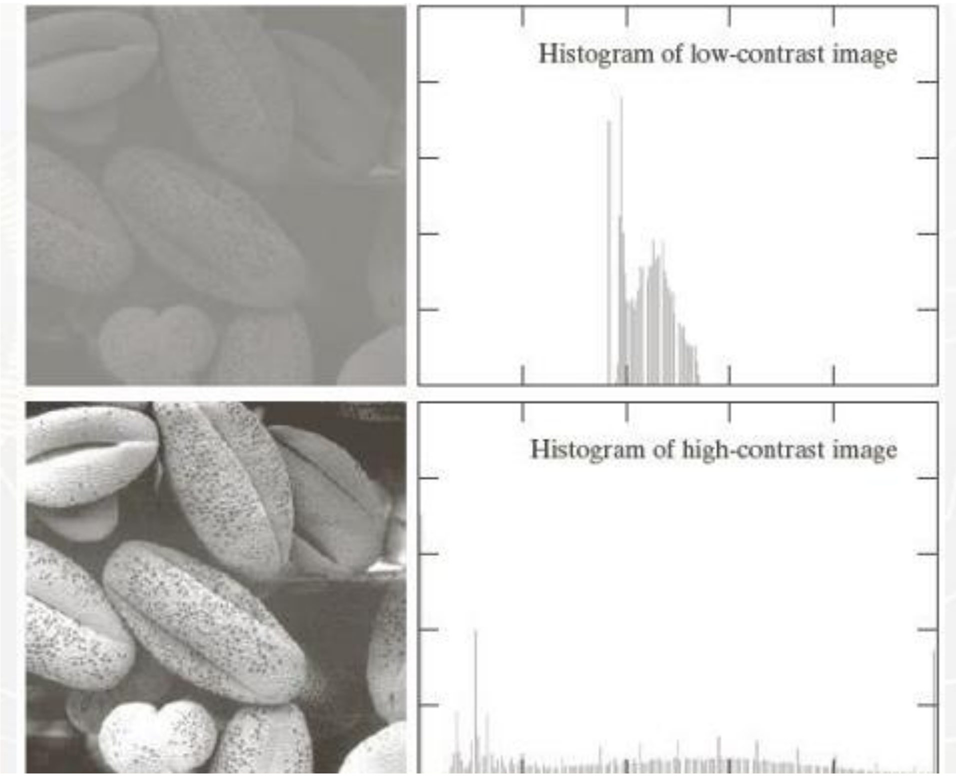
Image

Intensity	Number of Pixels
0	4
1	5
2	3
3	6
4	7

Histogram Table

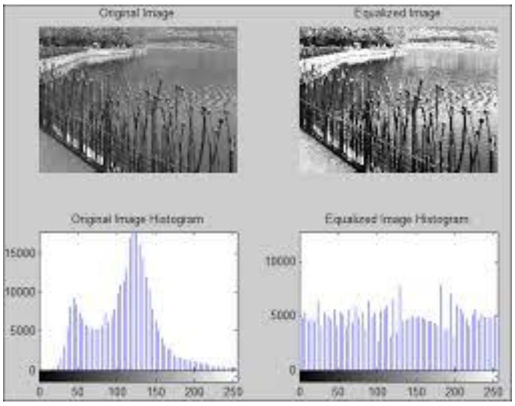
Image Histogram





Histogram Equalization:

Histogram equalization is an image processing technique used to improve an image's contrast by distributing its pixel intensity values over a wider range, making low-contrast images more detailed and clearer. It works by transforming the original intensity histogram into a flatter or more uniform histogram, thereby stretching the dynamic range and utilizing the full spectrum of intensity levels to enhance overall image quality.



How it Works

1. Calculate the Intensity Histogram:

First, an image histogram is created, which is a graph showing the frequency (number of pixels) for each intensity level in the image.

2. Determine the Probability Distribution Function (PDF):

The PDF is calculated by dividing the number of pixels at each gray level by the total number of pixels in the image.

3. Calculate the Cumulative Distribution Function (CDF):

The CDF is found by summing the PDF values up to each intensity level.

4. Transform the Intensities:

The CDF values are then multiplied by the maximum intensity level (e.g., 255 for an 8-bit image) and rounded to the nearest integer. This creates a new mapping for the pixel values.

5. Map New Values:

The original pixel intensities are then mapped to these new, equalized intensity levels, creating the final image with enhanced contrast.

Purpose and Applications

- Contrast Enhancement:

The primary goal is to increase the contrast in an image.

- Improved Detail and Clarity:

By spreading out the pixel intensities, details that were previously hidden in narrow intensity ranges become more visible.

- Medical Imaging:

Useful for enhancing details in X-rays or other medical scans, where pixel intensity can be clustered.

- Satellite Imagery:

Helps to clarify features in satellite images by improving overall contrast.

- General Image Processing:

Applied in various fields to make images more comprehensible and visually appealing.

Types of Histogram Equalization

- Global Histogram Equalization:

This method applies a single, uniform transformation to the entire image, enhancing the overall contrast.

- Local Histogram Equalization:

In this approach, the image is divided into smaller regions, and different transformations are applied to each region. This allows for more localized detail enhancement and better results in images with varying contrast levels across different areas.

Histogram Specification or Histogram Matching

Histogram specification, or histogram matching, is an image processing technique that transforms an image to match a specified target histogram, giving the output image a desired intensity distribution. It's a generalization of histogram equalization, where the goal is to create an image with a specific histogram shape rather than a flat, uniform one.

This process involves calculating the cumulative distribution functions (CDFs) of both the source and target images and mapping the source image's pixel values according to the target CDF to achieve the desired histogram.

Spatial Filtering

Spatial filtering in digital image processing is a technique that manipulates an image's pixel intensities based on the values of its neighbouring pixels to achieve effects like image smoothing, sharpening, or edge detection. It works by sliding a filter kernel (or mask) across the image, performing a calculation (like a sum-of-products for linear filters) within the kernel's neighbourhood at each pixel to produce the final filtered pixel value.

Common types include linear filters such as the Mean or Gaussian Filter for smoothing and non-linear filters like the Median filter for noise reduction, while sharpening filters like the Laplacian highlight edges.

How it Works

1. **Define a Mask/Kernel:**

A small matrix, known as a mask or kernel, is defined. This matrix contains coefficients that dictate the weighting and area of the neighboring pixels to be considered.

2. **Slide the Mask:**

The mask is systematically moved across the input image, pixel by pixel.

3. **Apply Operation:**

At each pixel, the filter performs an operation using the values of the pixels under the mask.

- **Linear Filters:** Perform a weighted sum of the neighboring pixels, often through convolution.
- **Nonlinear Filters:** Use operations like taking the median, maximum, or minimum value within the neighborhood.

4. **Produce Output Pixel:**

The result of the operation is the new intensity value for the center pixel in the output image.

Types of Spatial Filters

- **Linear Filters:**
  - **Smoothing Filters (Low-Pass):** Reduce noise and blur details by averaging pixel values. Examples include the mean/averaging filter and the Gaussian filter.
  - **Sharpening Filters (High-Pass):** Enhance edges and details by highlighting variations in intensity. Examples include gradient and Laplacian filters.
- **Nonlinear Filters:**
  - **Median Filter:** A popular order-statistics filter that replaces a pixel with the median value of its neighbors, effectively reducing salt-and-pepper noise while better preserving edges than linear filters.

Key Applications

- **Noise Reduction:** Smoothing filters (e.g., median, mean, Gaussian) are used to reduce noise in images.
- **Image Sharpening:** High-pass filters are used to emphasize edges, contours, and fine details.
- **Feature Extraction:** Filters can highlight specific features or textures within an image.

Smoothing Filter

A smoothing filter in digital image processing is a low-pass filter that reduces noise and blurs an image by averaging pixel values in a neighbourhood, effectively attenuating high-frequency components that often represent noise. Common smoothing filters include the mean filter, which uses a simple average, and the Gaussian filter, which assigns more weight to central pixels. Smoothing is used as a pre-processing step to remove noise and small details before other operations like object extraction.

How Smoothing Filters Work

1. **Low-Pass Filtering:**

   Smoothing filters function as low-pass filters, meaning they allow low-frequency information to pass through while suppressing high-frequency components. Noise is characterized by sharp, high-frequency changes in intensity, so these filters are effective at reducing it.

2. **Kernel Convolution:**

   The filter uses a small array called a kernel (or window). This kernel is convolved with the image, meaning it's slid across the image, and at each position, the pixel value is replaced with a value computed from the pixels under the kernel.

3. **Neighborhood Averaging:**

   For each pixel, the filter replaces its value with the average (or weighted average) of the intensity values of the pixels within its neighborhood.

Types of Smoothing Filters

Smoothing filters are categorized into two main types:

- **Linear Filters:**

  These filters perform a weighted average of the neighborhood pixels.

  - **Mean (or Box) Filter:** The simplest linear filter where each pixel in the neighborhood is given equal weight.
  - **Gaussian Filter:** A weighted average filter that uses a kernel shaped like a Gaussian (bell) curve, giving more weight to pixels closer to the center.

- **Nonlinear Filters (Order-Statistics Filters):**

  These filters are based on the rank or order of the pixel values in the neighborhood.

  - **Median Filter:** Replaces the center pixel's value with the median value of the pixels in the neighborhood. This is particularly effective at removing salt-and-pepper noise while preserving edges better than linear filters.

Applications of Smoothing Filter

- **Noise Reduction**:

  The primary application, removing random noise that degrades image quality.

- **Pre-processing:**

  Used to blur an image before object extraction, making it easier to identify larger features by removing small, distracting details.

- **Smoothing for Visual Perception:**

  Improves image quality for human viewers by reducing the appearance of pixelation and harsh transitions.