CSC360 - Image Processing - Fall 2022

Instructor: Dr. Alaa Hamdy

Eng. Noha Elmasry

Image Processing I

Lab 2
Gray Level Transformations

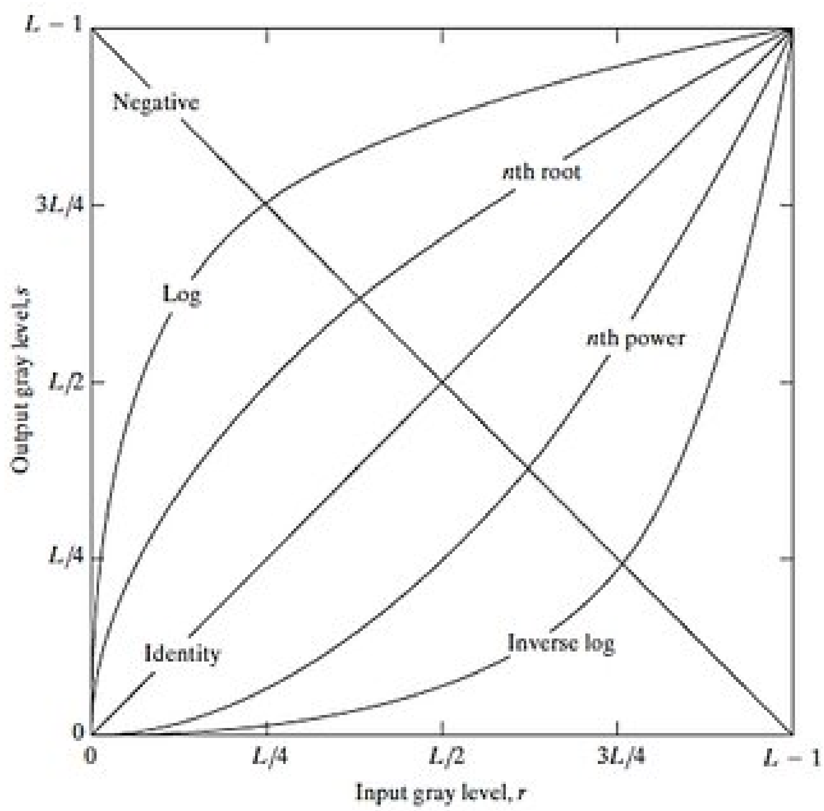
Gray Level Transformations

All Image Processing Techniques are focused on gray level transformation as it operates directly on pixels. The gray level image involves 256 levels of gray ***from 0 to 255***.

The simplest formula for image enhancement technique is:

s = T * r

Where **T** is transformation, **r** is the value of pixels **before processing**, **s** is pixel value **after processing**.



There are three types of transformation:

1. Linear transformation

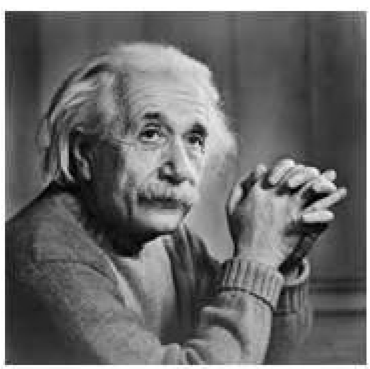
The linear transformation includes **identity transformation** and **negative transformation**.

The linear transformations can be defined by this formula

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

- In **identity transformation**, each value of the image is directly mapped to each other values of the output image.
- **Negative transformation** is the opposite of identity transformation. Here, each value of the input image is **subtracted from L-1 (maximum pixel value)** and then it is mapped onto the output image

Input Image



Output Image

Applications:

- **Linear transformations** are often used in machine learning applications. They are useful in the modeling of 2D and 3D animation, where an object's size and shape needs to be transformed from one viewing angle to the next.

- **Image negation** helps finding the details from the darker regions of the image. The **negative transformation** has its applications in the areas of Medical Imaging, Remote Sensing and others.

2. Logarithmic transformation

Logarithmic transformation is divided into two types:

- **Log transformation**
- **Inverse log transformation**

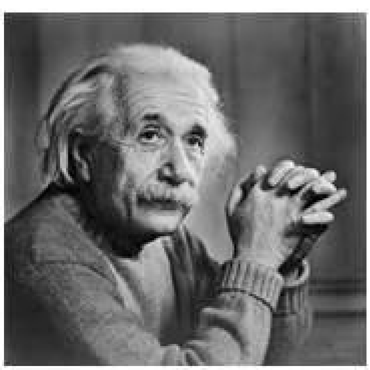
The log transformations can be defined by this formula

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 values 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 results in the following image enhancement.

Input Image



Log Transform Image

The inverse log transform is opposite to log transform.

Applications:

- **Log transformation** is arguably the most popular among the different types of transformations used to transform skewed data to approximately conform to normality.

3. Power - Law transformations (Gamma Correction)

Power Law Transformation is of two types of transformation nth power transformation and nth root transformation.

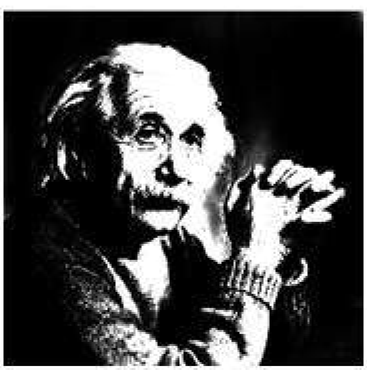
s = cr ^ γ

Here, **γ** is gamma, by which this transformation is known as **gamma transformation**.

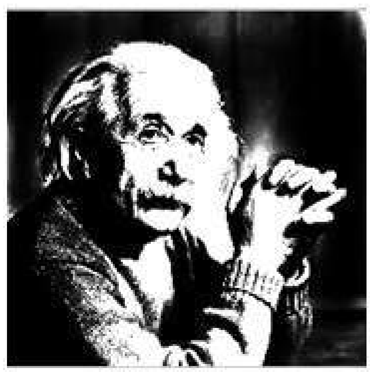
All display devices have their own gamma correction. That is why images are displayed at different intensities. These transformations are used for enhancing images.

The same image but with different gamma values has been shown here.

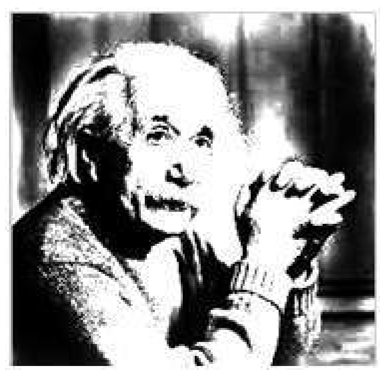
Gamma = 10



Gamma = 8



Gamma = 6



Applications:

- Gamma correction controls the overall brightness of an image. Images which are not properly corrected can look either bleached out, or too dark.

4. Gray Level Slice

Gray-level slicing aims to highlight a specific range [A… B] of gray levels. It simply maps all gray levels in the chosen range to a high value. Other gray levels are either mapped to a low value or left unchanged.

Steps of Gray Level Slicing:

1. Define min_thr and max_thr
2. Check if r lies in the range you defined in step 1.
    a. If yes: s = 255
    b. If no: s = r

Input Image

Gray level slicing with background

Gray level slicing without background

Applications:

- **Gray level slicing** manipulates groups of intensity levels in an image up to specific range by diminishing rest or by leaving them alone. This transformation is applicable in medical images and satellite images such as X-ray flaws, CT scan.

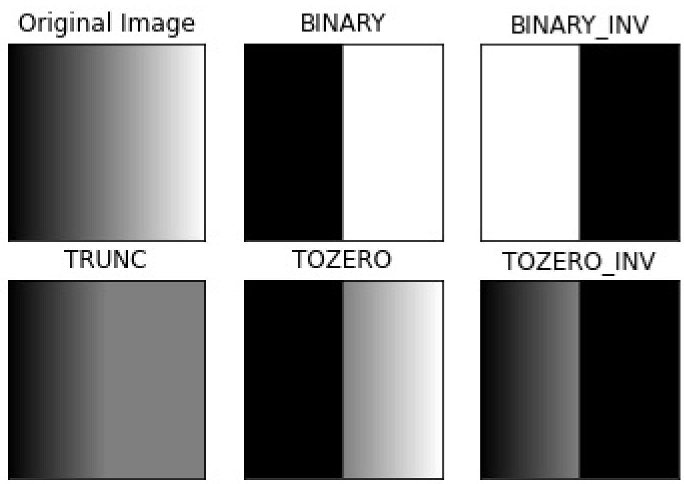
5. Brightness Thresholding

Brightness Thresholding is a special case from Gray Level Slicing. It is a binary threshold.

Steps of Brightness Thresholding:

1. Define thr
2. Check if r is less than thr
    a. If yes: s = 0
    b. If no: s = 255

Different types of thresholding:



image

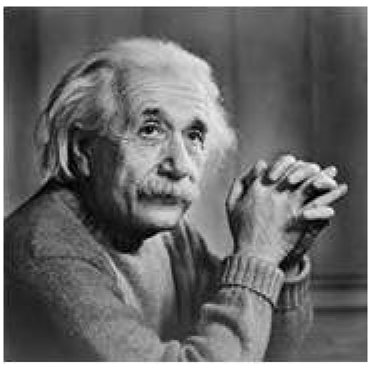
Ready made function:

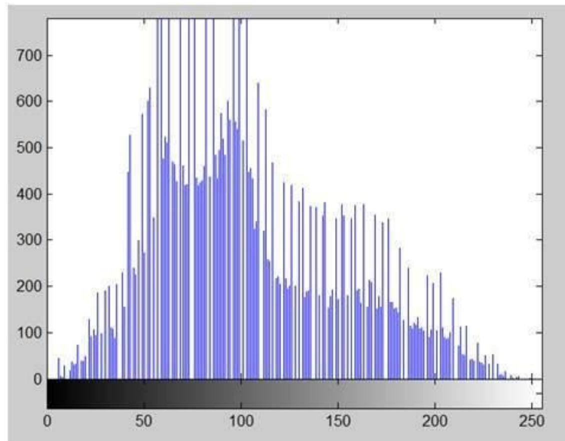
```
ret,thresh = cv.threshold(img,min_thr,max_thr,thr_type)
```

6. Histogram

An image histogram is **a graphical representation of the number of pixels in an image as a function of their intensity**. Histograms are made up of bins, each bin representing a certain intensity value range.

Example:





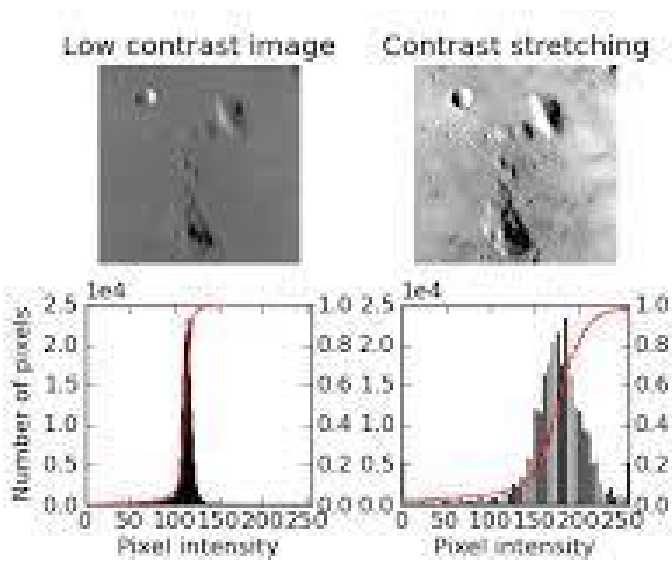
The x axis of the histogram shows the range of pixel values. Since it's an 8 bpp image, that means it has 256 levels of gray or shades of gray in it. That's why the range of x axis starts from 0 and end at 255 with a gap of 50. Whereas on the y axis, is the count of these intensities.

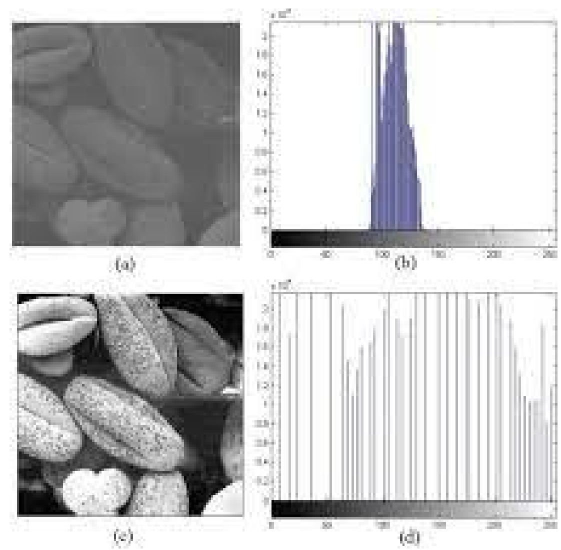
Applications:

The histograms have wide applications in image brightness. Not only in brightness, but histograms are also used in adjusting contrast of an image. Another important use of histogram is to equalize an image. And last but not the least, histogram has wide use in thresholding. This is mostly used in computer vision.

7. Contrast Stretching

Contrast stretching (often called normalization) is **a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values**, e.g. the the full range of pixel values that the image type concerned allows.





Steps of Contrast Stretching:

1. Define r_min and r_max from the input image.
2. Define new desired max and min for output image (s_min and s_max).
3. Find new value for each pixel as follows:

$$s = (r - rmin) * \left(\frac{smax - smin}{rmax - rmin}\right) + smin$$

NB: If r_min = s_min and/or r_max = s_max, then an issue arises where the desired stretching is not performed.

Two techniques can be used in this case:

- ❖ Cut-off fraction:

  $$p = c * peak$$

  - p → new peak that determines new r_min and r_max.
  - c →cut-off fraction constant
  - $peak$ → actual peak value of the count from original histogram.

- ❖ Percentage:

  $$p = \% * \#\ of\ pixels$$

  - p → number of pixels to ignore from left and right in order to get new r_min and r_max.
  - % →percentage constant.
  - $\#\ of\ pixels$ → total number of pixels in the input image.

Applications:

Contrast stretching is used when the gray level distribution is narrow due to poor illumination, lack of dynamic range in the imaging sensor or others.