# MODULE2

- **Spatial Domain: Some Basic Intensity Transformation Functions,**
- Histogram Processing, Fundamentals of Spatial Filtering, Smoothing
- Spatial Filters, Sharpening Spatial Filters
- [Text: Chapter 3: Sections 3.2 to 3.6]

- **Frequency Domain: Preliminary Concepts, The Discrete Fourier**
- Transform (DFT) of Two Variables, Properties of the 2-D DFT, Filtering
- in the Frequency Domain, Image Smoothing and Image Sharpening
- Using Frequency Domain Filters, Selective Filtering.
- [Text: Chapter 4: Sections 4.2, 4.5 to 4.10]

# Image Enhancement- Spatial Domain

# Introduction

- What is image enhancement?
  - A process of enhancing the visual quality of images due to non ideal image acquisition process (e.g., motion blurring, out-of-focus, poor illumination, coarse quantization etc.)
- Image visual quality assessment
  - Objective quality metrics (e.g., MSE) might not always match subjective quality scores
  - Human vision system (HVS) is the ultimate JUDGE.

# Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a  specific application.
- The suitableness is up to each application.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images

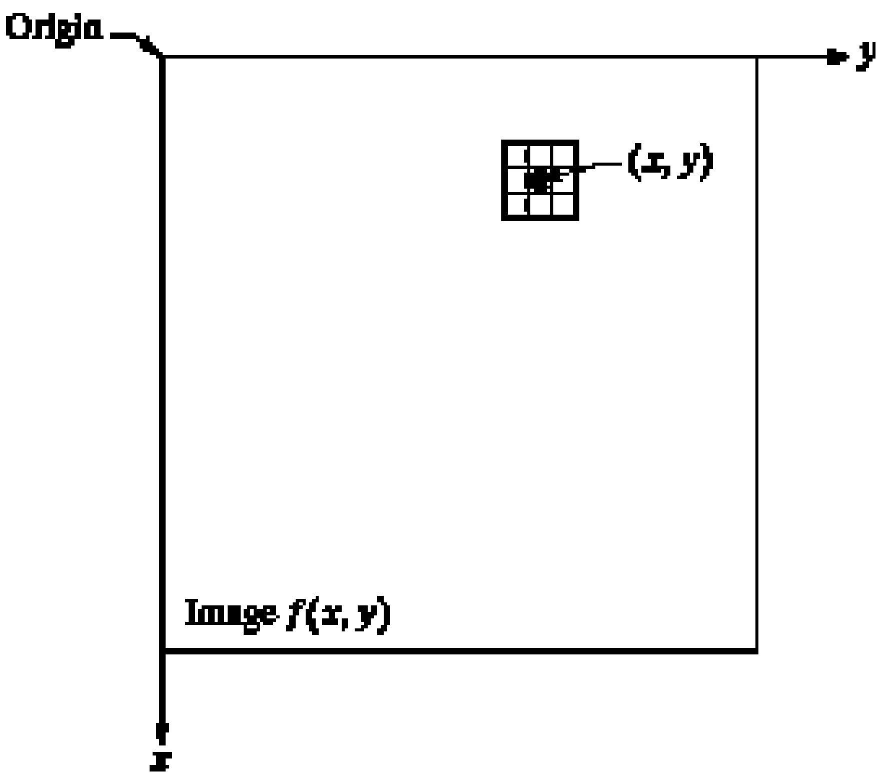
# 2 domains

- Spatial Domain : (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain :
  - Techniques are based on modifying the Fourier transform of an image
- There are some enhancement techniques based on various combinations of methods from these two categories.

# Good images

- For human visual
  - The visual evaluation of image quality is a highly subjective process.
  - It is hard to standardize the definition of a good image.
- For machine perception
  - The evaluation task is easier.
  - A good image is one which gives the best machine recognition results.
- A certain amount of trial and error usually is required before a particular image enhancement approach is selected.

# Spatial Domain



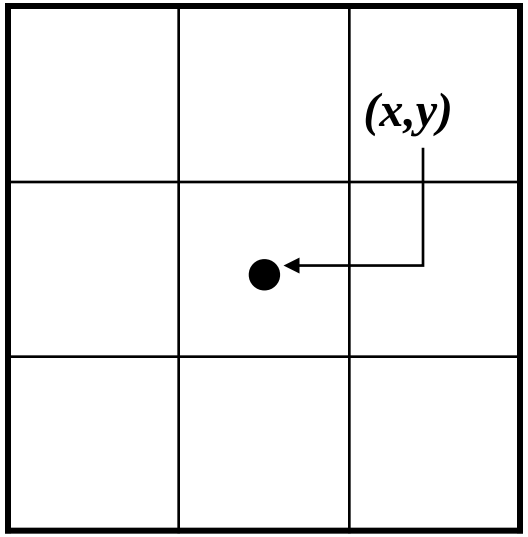

- Procedures that operate directly on pixels.

$$g(x,y) = T[f(x,y)]$$

where

- $f(x,y)$ is the input image
- $g(x,y)$ is the processed image
- $T$ is an operator on $f$ defined over some neighborhood of $(x,y)$

# Mask/Filter




- Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) or circular subimage area centered at (x,y)
- The center of the subimage is moved from pixel to pixel starting at the top of the corner

# Point Processing

- Neighborhood = 1x1 pixel
- $g$ depends on only the value of $f$ at $(x,y)$
- $T$ = gray level (or intensity or mapping) transformation function

$$s = T(r)$$

- Where
  - $r$ = gray level of $f(x,y)$
  - $s$ = gray level of $g(x,y)$

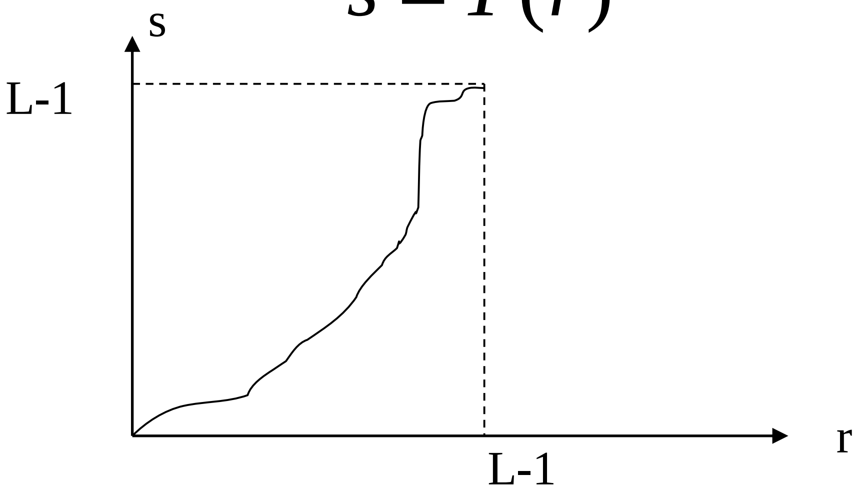
# Enhancement by Point Processing

- These are methods based only on the intensity of single pixels.
  - r denotes the pixel intensity before processing.
  - s denotes the pixel intensity after processing.

# Point Operations Overview

Point operations are **zero-memory** operations where a given gray level r∈[0,L-1] is mapped to another gray level r∈[0,L-1] according to a transformation

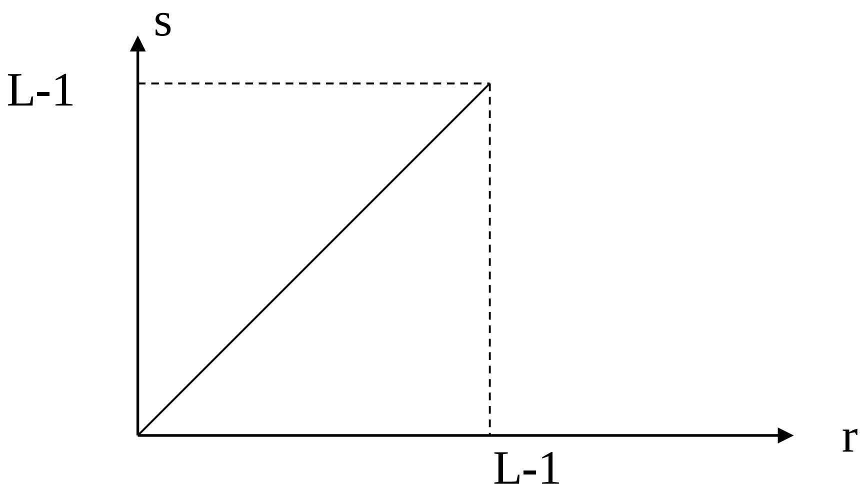
$$s = T(r)$$




L=256: for grayscale images

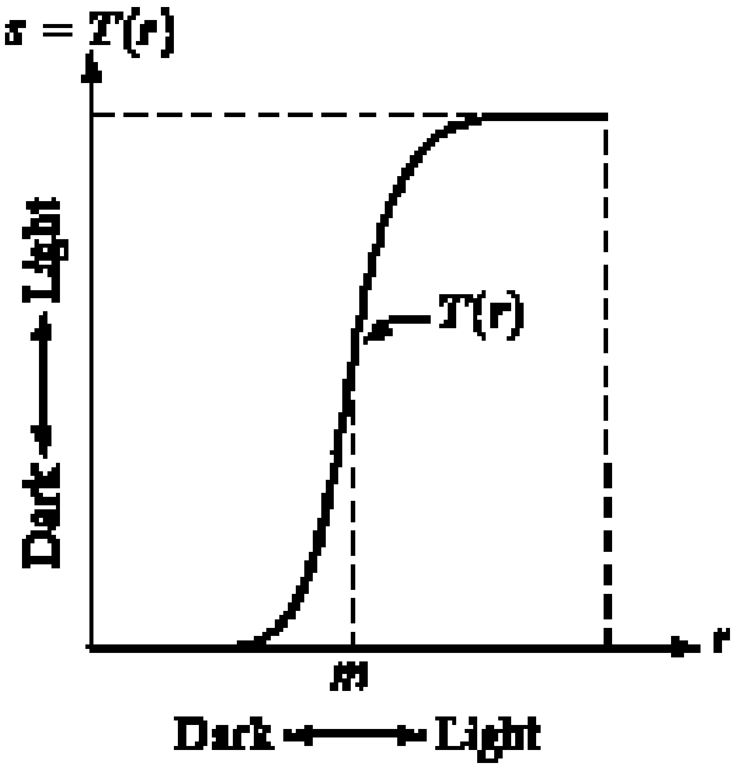
# Lazy Man Operation

$$s = r$$



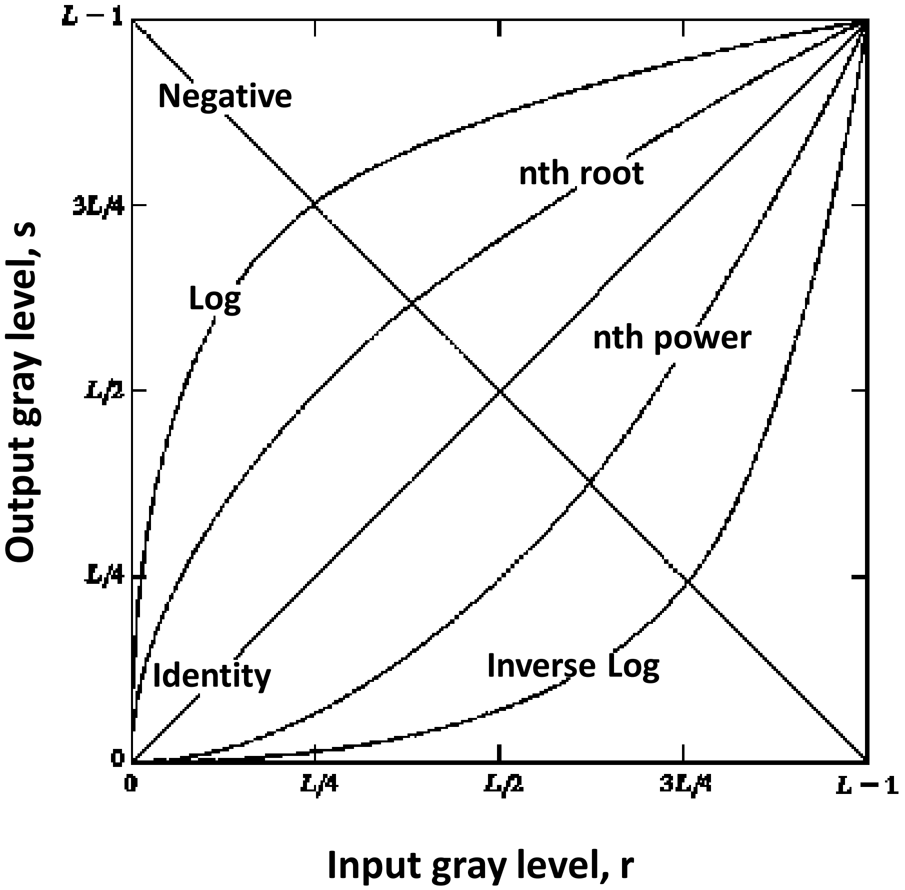

No influence on visual quality at all

# Contrast Stretching



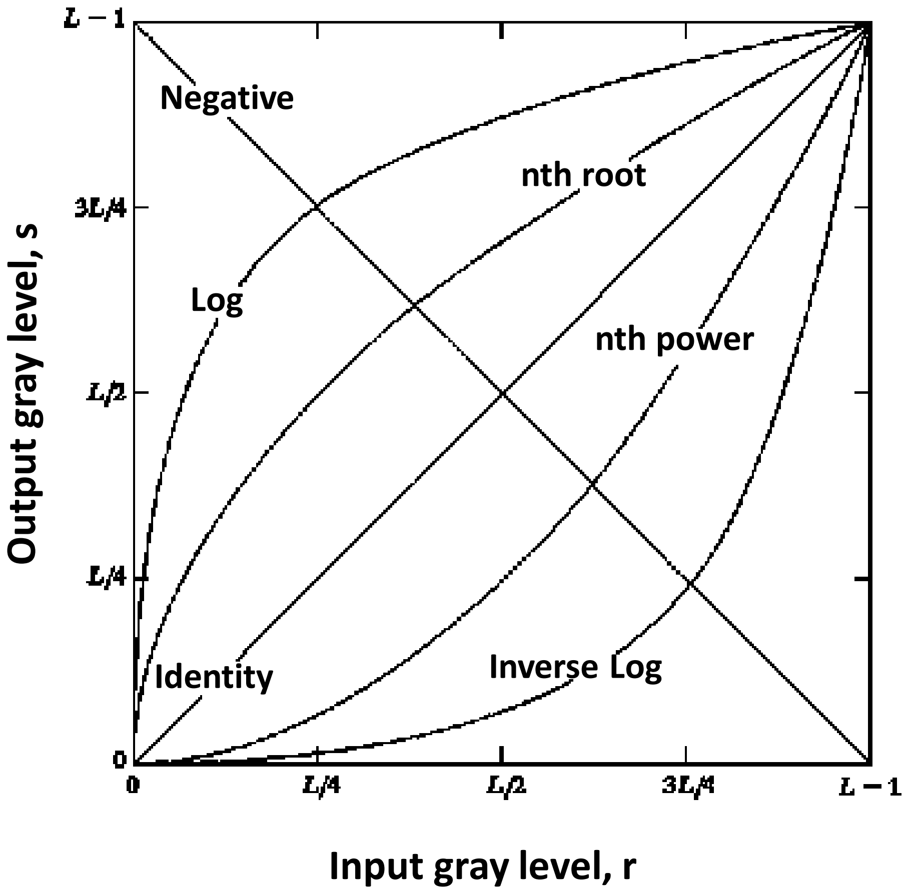

- Produce higher contrast than the original by
  - darkening the levels below m in the original image
  - Brightening the levels above m in the original image

# 3 basic gray-level transformation functions




- Linear function
  - Negative and identity transformations
- Logarithm function
  - Log and inverse-log transformation
- Power-law function
  - $n^{th}$ power and $n^{th}$ root transformations

# Image Negatives

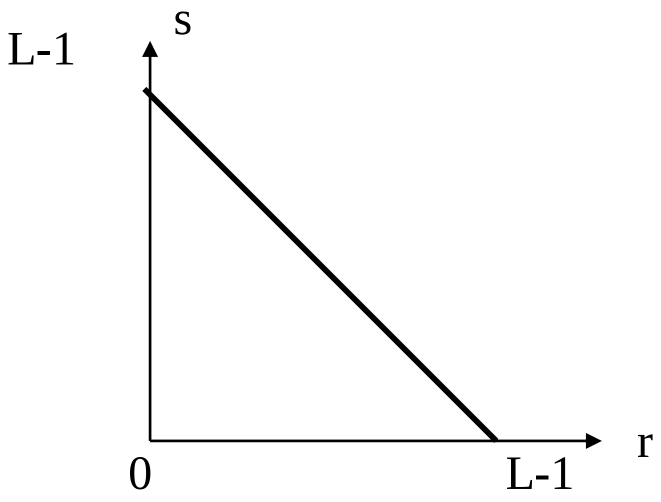



- An image with gray level in the range **[0, L-1]** where $L = 2^n$ ; n = 1, 2…
- Negative transformation :

$$s = L - 1 - r$$

- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.

# Digital Negative



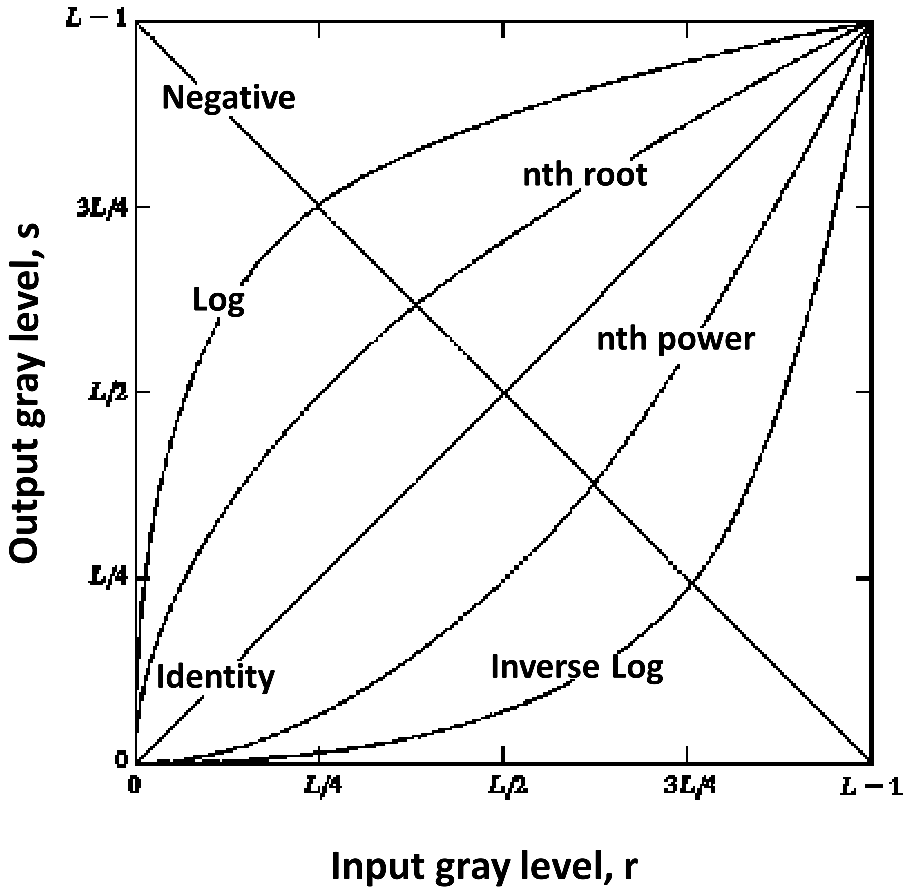

$$s = L - 1 - r$$

# Some Examples


$$I'(x,y) = I(x,y) + b$$


$$I'(x,y) = a \cdot I(x,y)$$

# Log Transformations



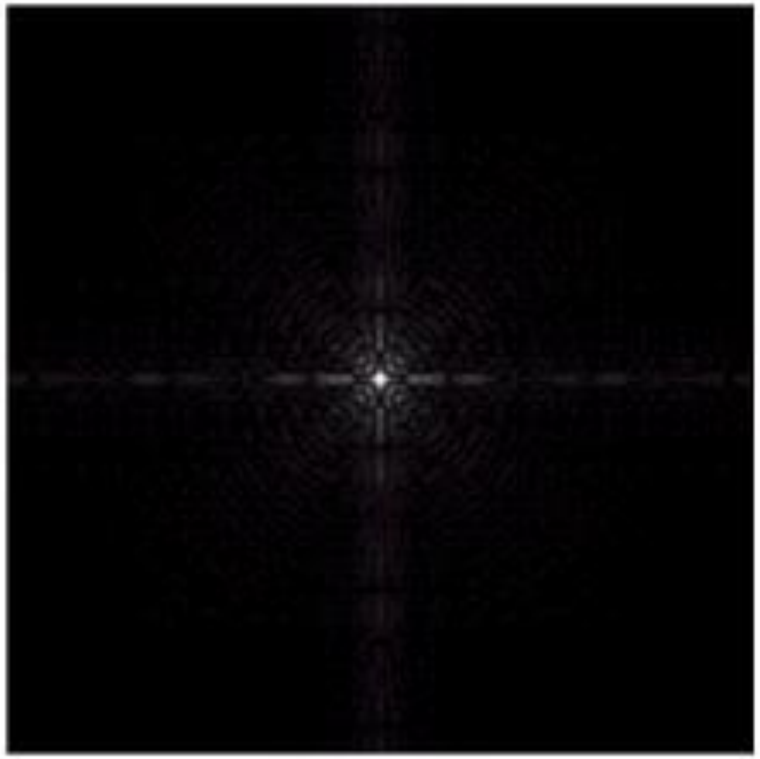

$$s = c \log (1+r)$$

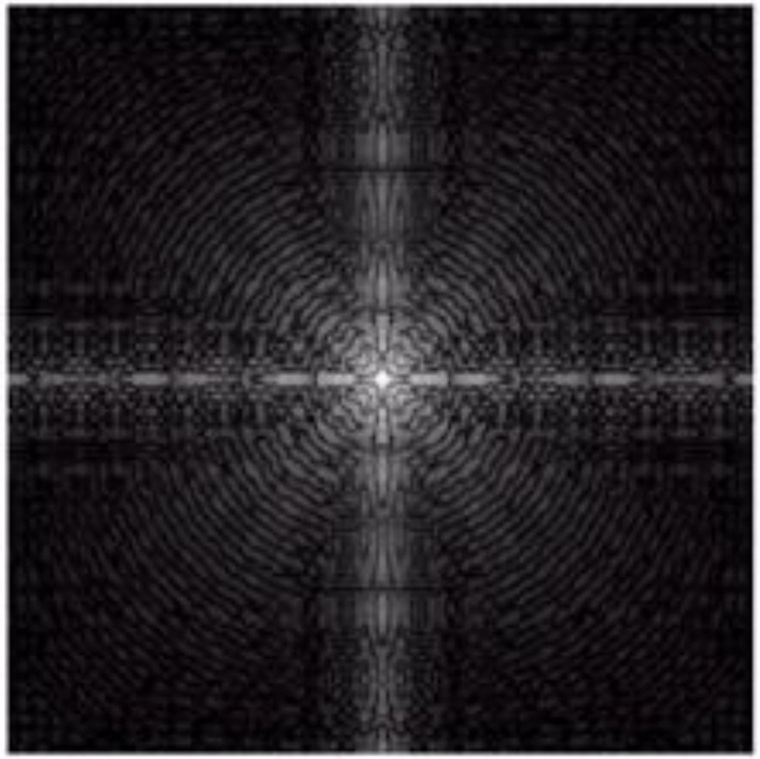
- c is a constant and $r \geq 0$
- Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- Opposite is true of higher values of input values.
- Used to expand the values of dark pixels in an image while compressing the higher-level values.

# Log Transformations

- It compresses the dynamic range of images with large variations in pixel values
- Example of image with dynamic range: Fourier spectrum image
- It can have intensity range from 0 to $10^6$ or higher.
- We can’t see the significant degree of detail as it will be lost in the display.

# Example of Logarithm Image



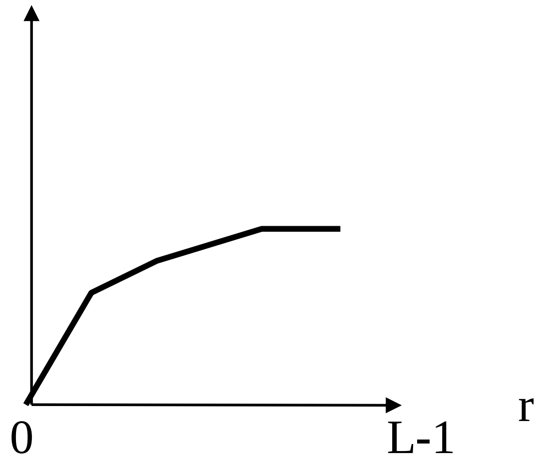


**Fourier Spectrum with range = 0 to 1.5 x $10^6$**

**Result after apply the log transformation with c = 1, range = 0 to 6.2**

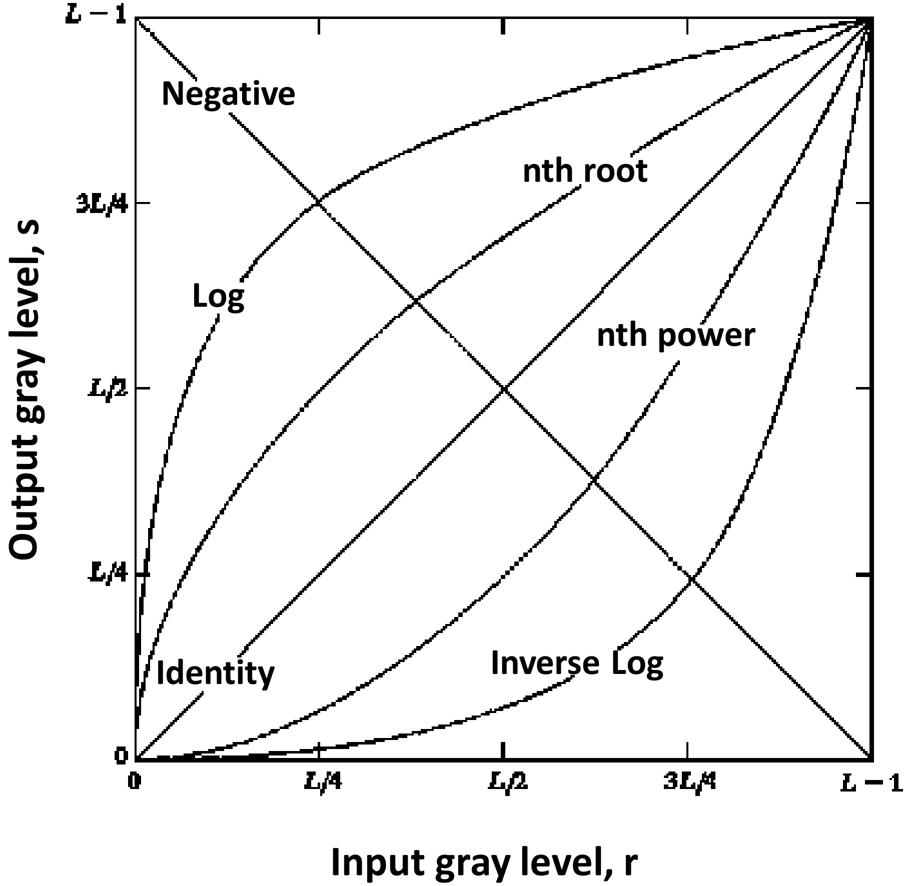
# Range Compression

$$s = c\log_{10}(1+r)$$




*c=100*

# Inverse Logarithm Transformations




- Do opposite to the Log Transformations
- Used to expand the values of high pixels in an image while compressing the darker-level values.

# Power-Law Transformations

$$s = cr^{\gamma}$$

c, $\gamma$ : positive constants

- Gamma correction

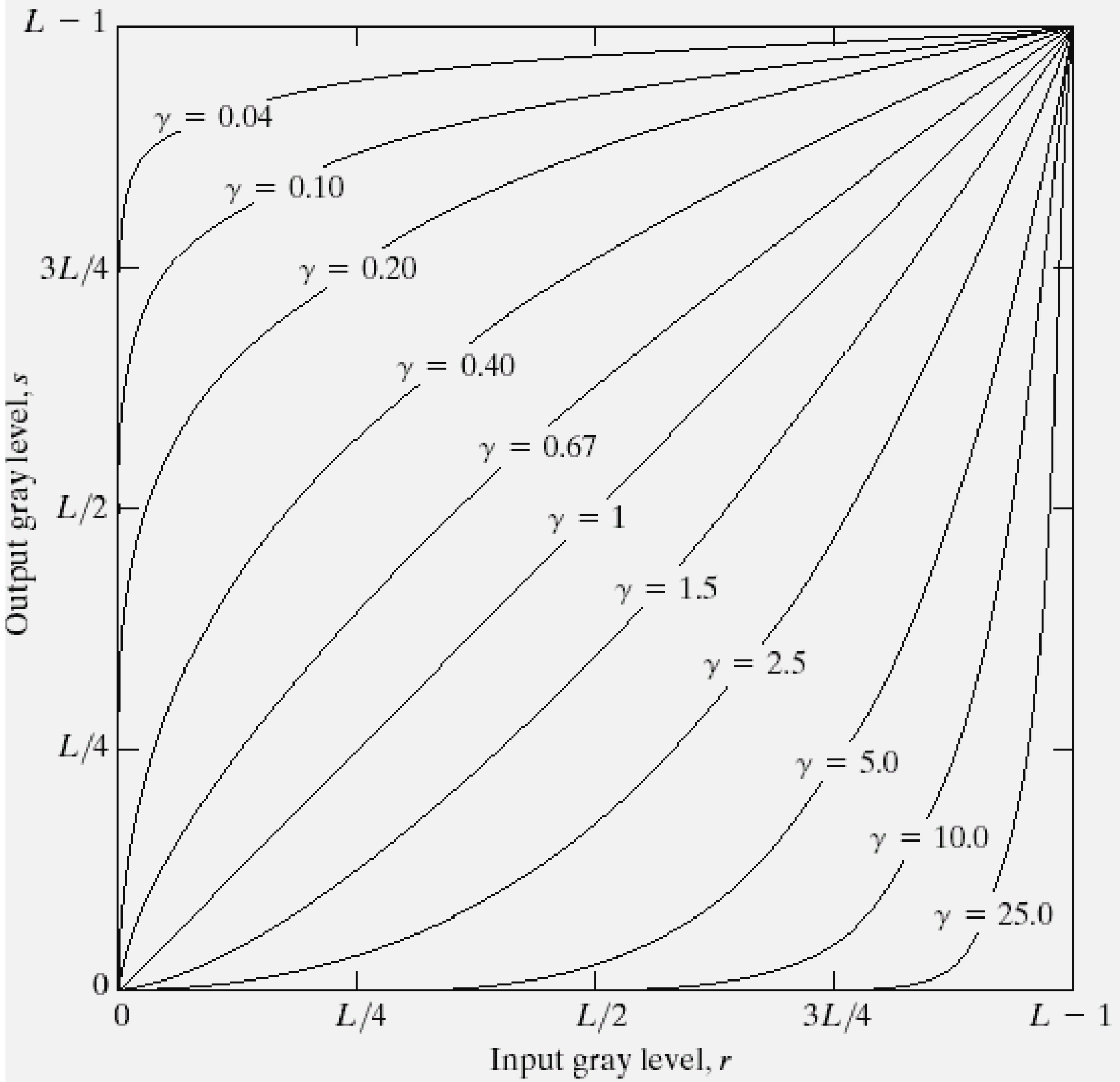



**FIGURE 3.6** Plots of the equation $s = cr^{\gamma}$ for various values of $\gamma$ ($c = 1$ in all cases).

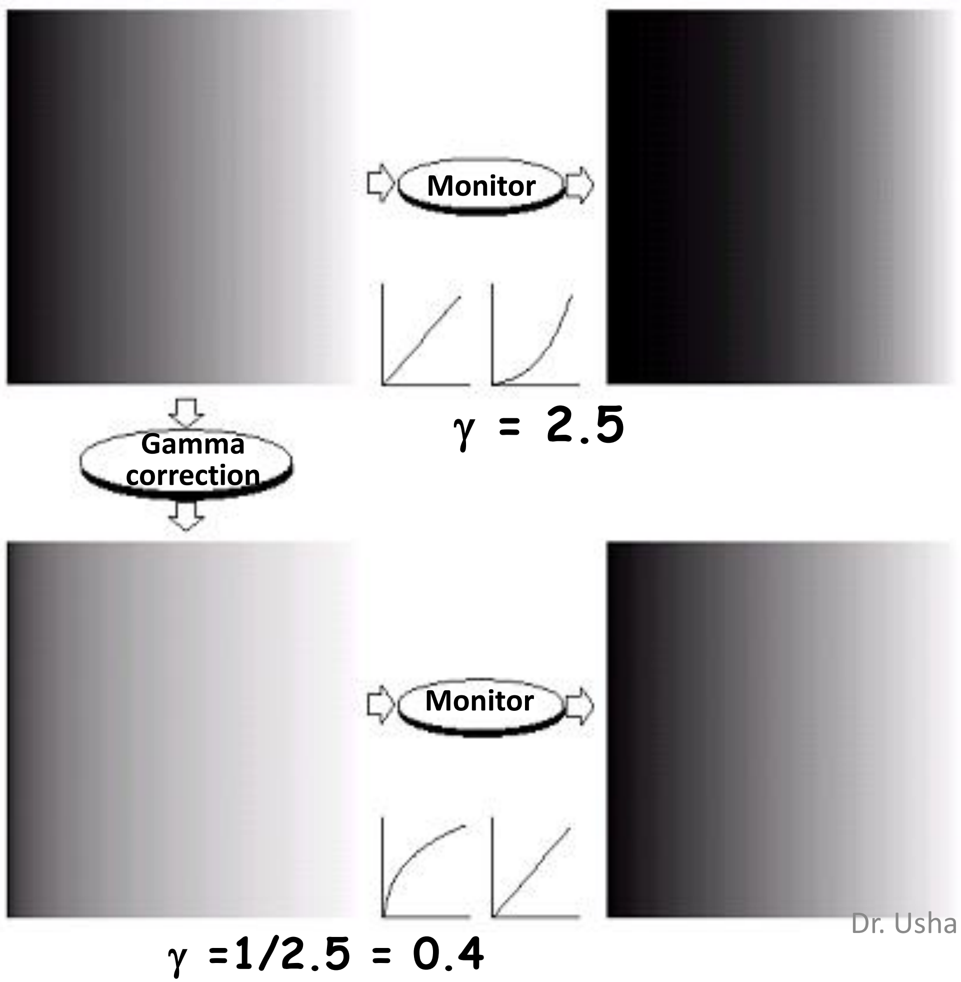
$\gamma$=c=1: identity

# Power-Law Transformations

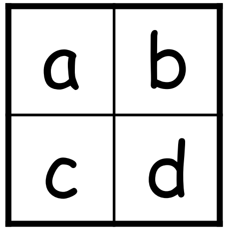
$$s = cr^{\gamma}$$

- c and $\gamma$ are positive constants
- Power-law curves with fractional values of $\gamma$ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- $c = \gamma = 1$ ⇨ Identity function

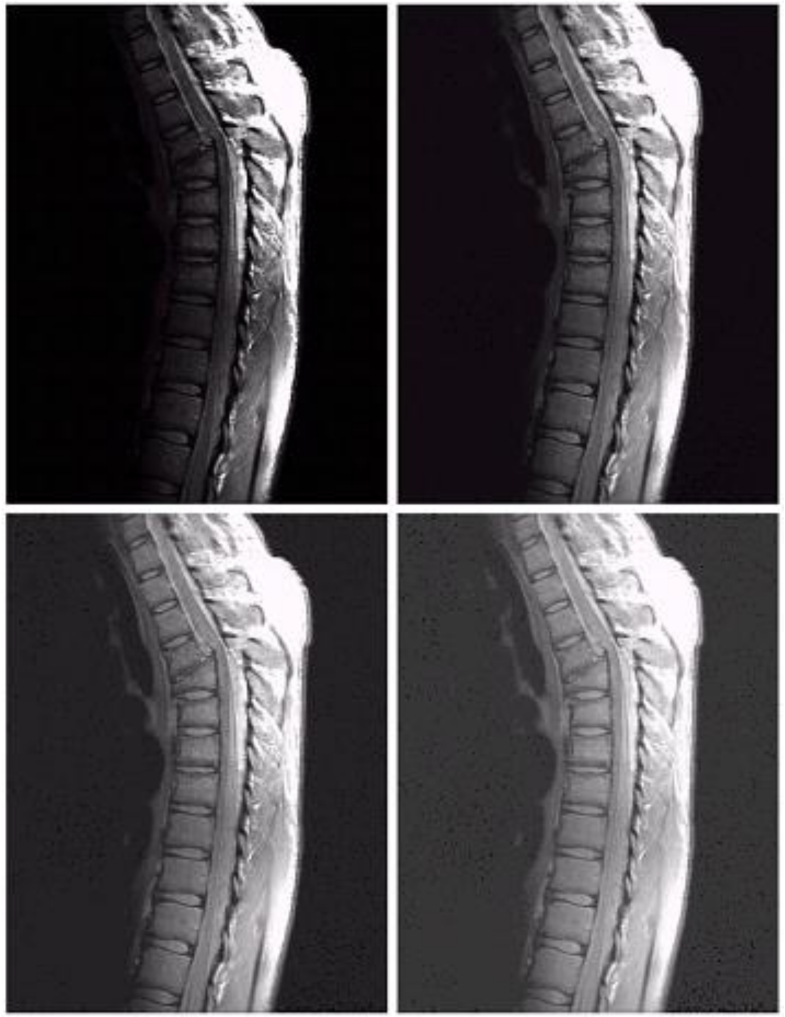
# Gamma correction




- Cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with $\gamma$ varying from 1.8 to 2.5
- The picture appears darker.
- Gamma correction is done by preprocessing the image before inputting it to the monitor with $s = cr^{1/\gamma}$




# Another example : MRI



(a) a magnetic resonance image of an upper thoracic human spine with a fracture dislocation and spinal cord impingement

- The picture is predominately dark
- An expansion of gray levels are desirable ⇨ needs $\gamma < 1$

(b) result after power-law

transformation with $\gamma$ = 0.6, c=1

(c) transformation with $\gamma$ = 0.4

(best result)

(d) transformation with $\gamma$ = 0.3

(under acceptable level)

# Effect of decreasing gamma

- When the $\gamma$ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight “wash-out” look, especially in the background

| | |
|---|---|
| a | b |
| c | d |

# Another example

(a) image has a washed-out appearance, it needs a compression of gray levels ⇨ needs $\gamma > 1$

(b) result after power-law

transformation with $\gamma$ = 3.0 (suitable)

(c) transformation with $\gamma$ = 4.0 (suitable)

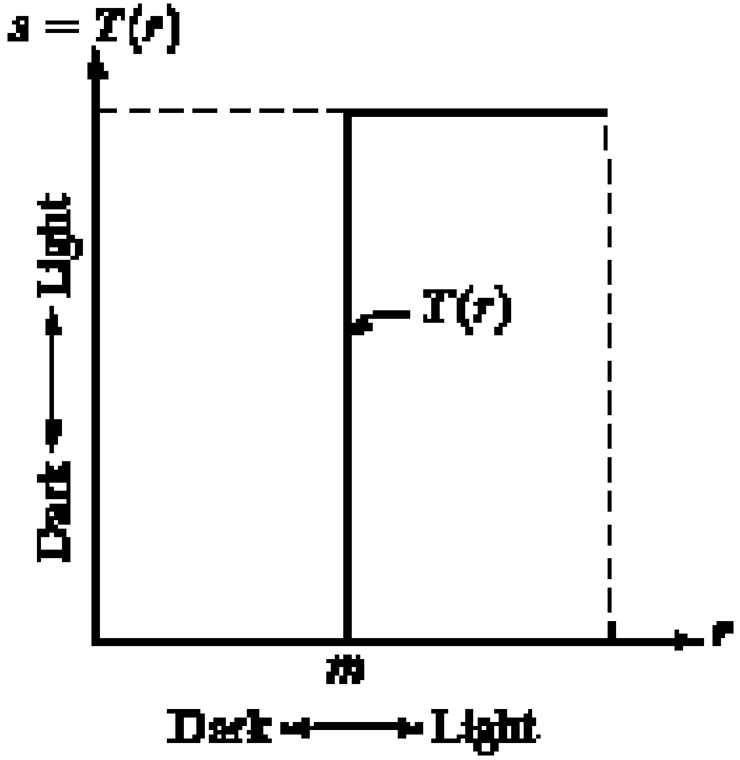
(d) transformation with $\gamma$ = 5.0

(high contrast, the image has areas that are too dark, some detail is lost)

# Some Simple
# Intensity Transformations

- Piecewise-Linear Transformation Functions:
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing

# Thresholding

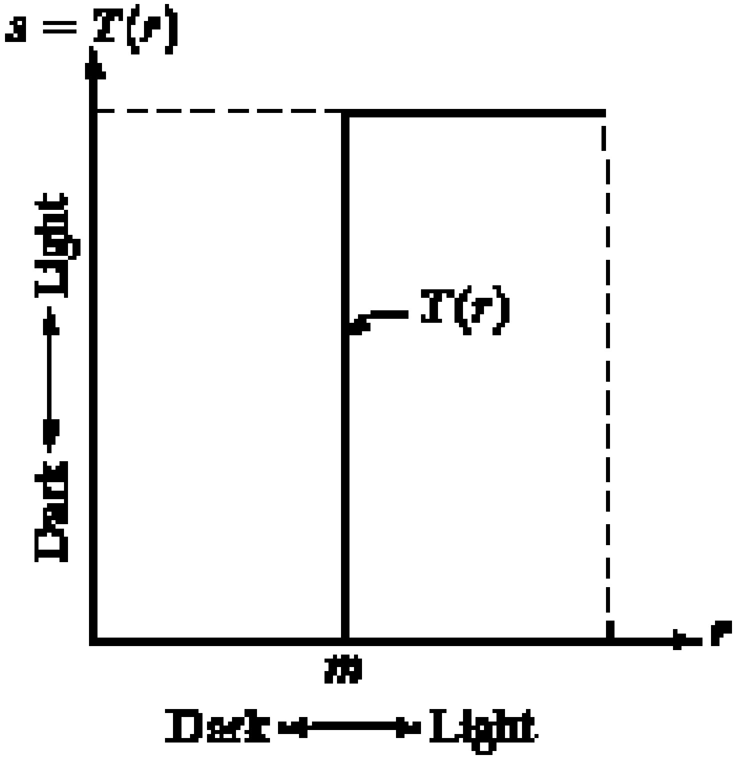



- Produces a two-level image

For the given sub-image, if m=140

Find the new image after thresholding

| | | | |
|---|---|---|---|
| **0** | **10** | **50** | **100** |
| **5** | 95 | 150 | 200 |
| **110** | 150 | 190 | 210 |
| **175** | 210 | 255 | 100 |



# ??

$$I'(x,y) = \begin{cases} 255 & \text{if } I(x,y) > 150 \\ 0 & \text{otherwise} \end{cases}$$

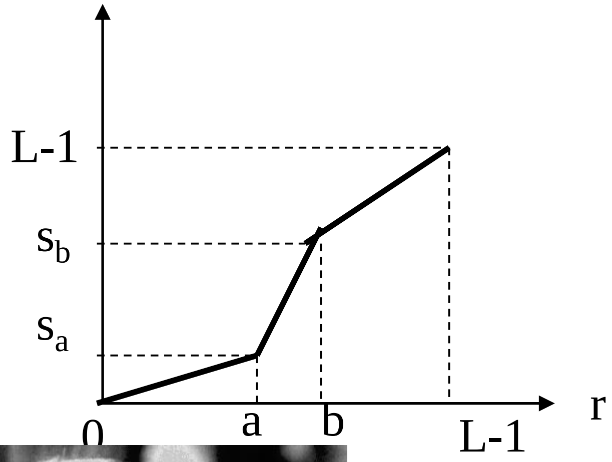
# Piecewise-Linear Transformation Functions

- Advantage:
  - The form of piecewise functions can be arbitrarily complex
- Disadvantage:
  - Their specification requires considerably more user input

- Low contrast images can occur due to
  - poor or non uniform lighting conditions
  - due to non linearity in imaging sensors
  - due to small dynamic range of imaging sensors
- The slope of transformation is chosen greater than unity in the region of stretch.
- The parameters a & b are obtained by examining histogram

# Contrast Stretching

$$s = \begin{cases} \alpha r & 0 \le r < a \\ \beta(r-a)+s_a & a \le r < b \\ \gamma(r-b)+s_b & b \le r < L \end{cases}$$



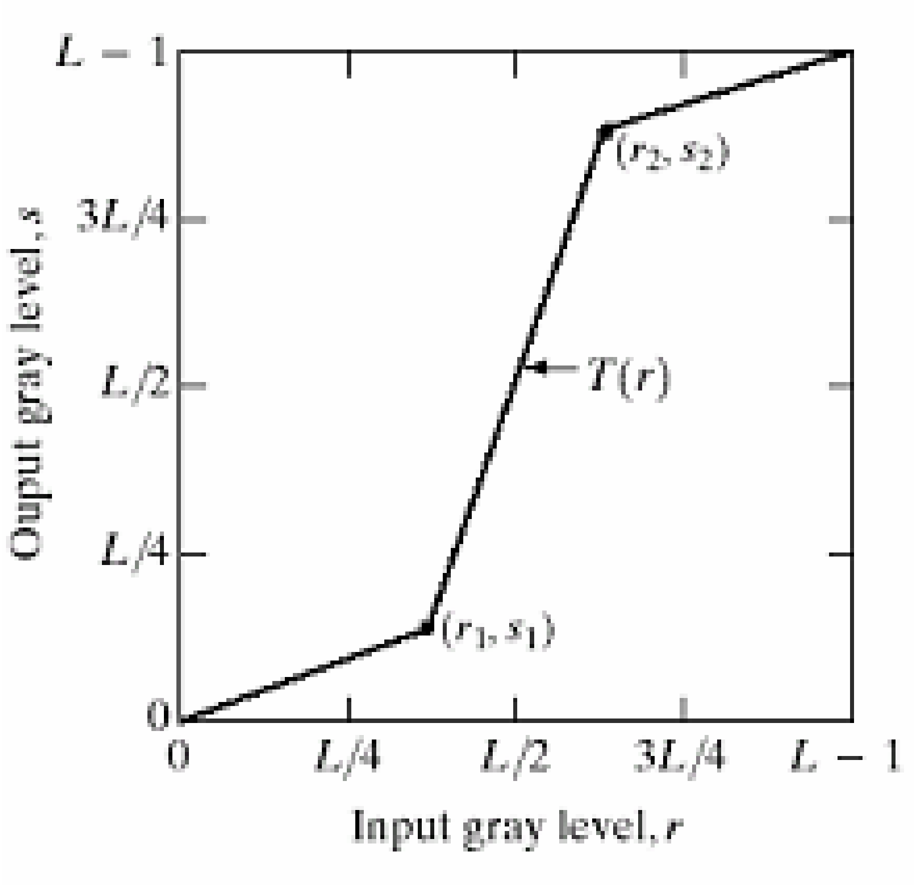

$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, s_a = 30, s_b = 200$$

For the given sub-image, if

$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, s_a = 30, s_b = 200$$

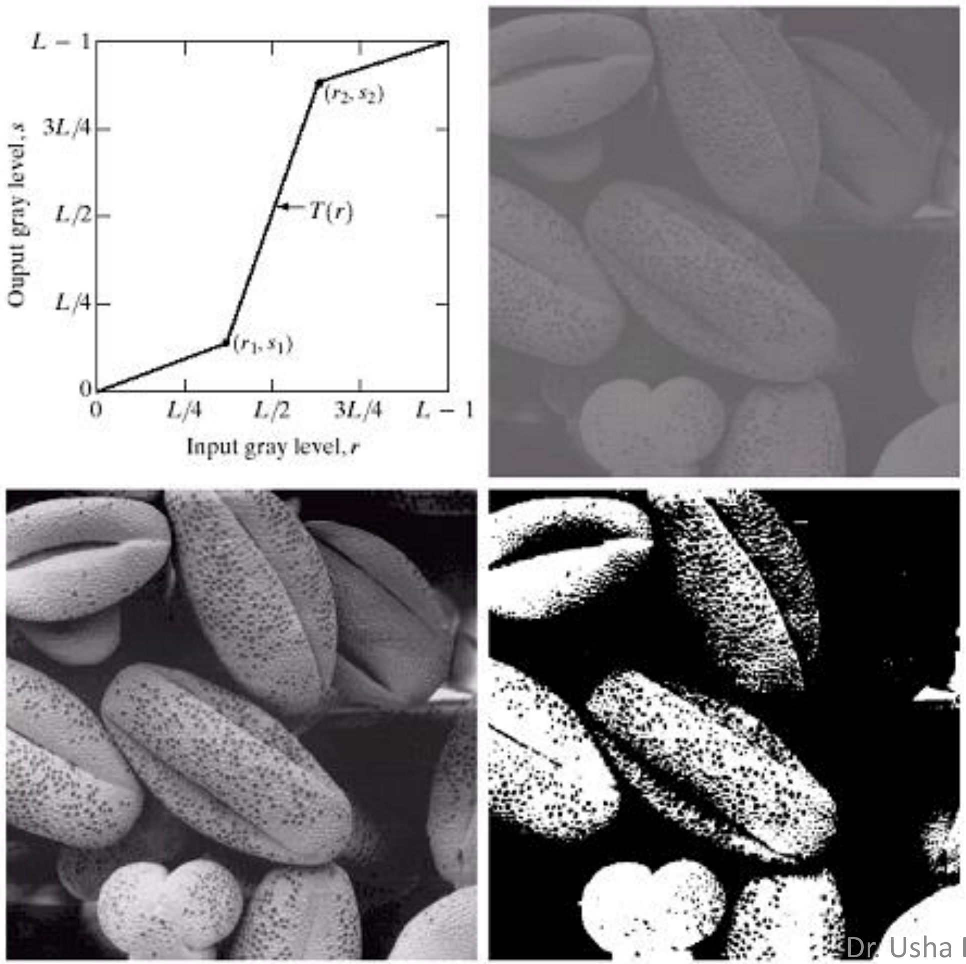
Find the new image after contrast stretching

| | | | |
|---|---|---|---|
| **0** | **10** | **50** | **100** |
| **5** | 95 | 150 | 200 |
| **110** | 150 | 190 | 210 |
| **175** | 210 | 255 | 100 |



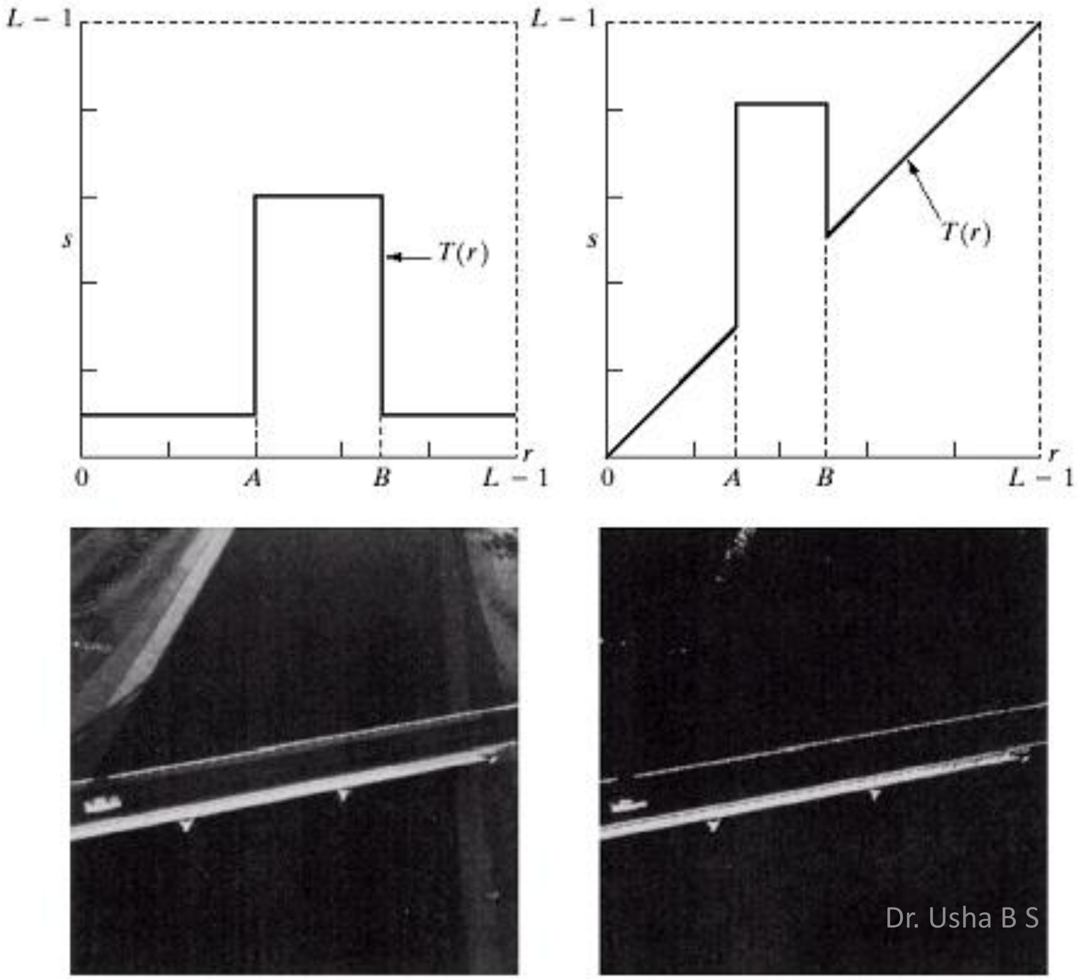

- If s1=r1, s2=r2
- no change in the gray level values
- If r1=r2, s1=0, s2=L-1
- it is Thresholding function
- r1<=r2, s1<=s2

# Contrast Stretching




- Increase the dynamic range of the gray levels in the image
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$
- (d) result of thresholding

# Gray-level slicing

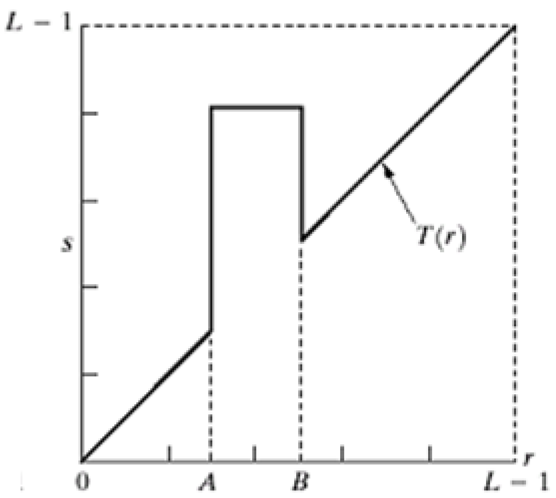



- Highlighting a specific range of gray levels in an image
  - Display a high value of all gray levels in the range of interest and a low value for all other gray levels
- (a) transformation highlights range [A,B] of gray level and reduces all others to a constant level
- (b) transformation highlights range [A,B] but preserves all other levels

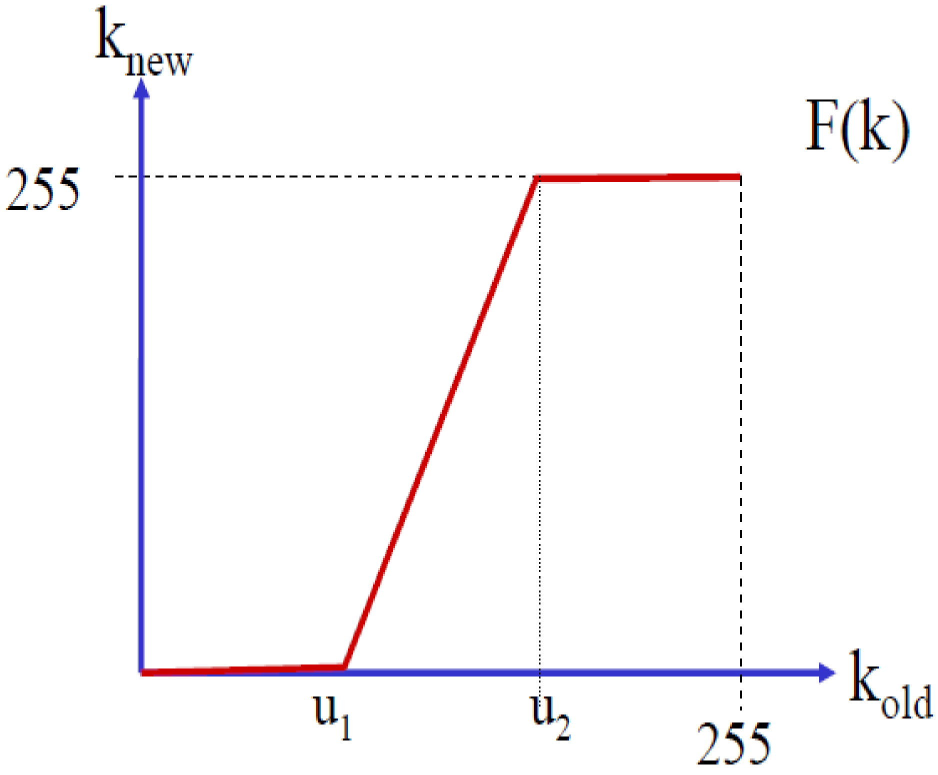
For the given sub-image, if A=50 and B =170, M=200

Find the new image after gray level slicing

| | | | |
|---|---|---|---|
| **0** | **10** | **50** | **100** |
| **5** | 95 | 150 | 200 |
| **110** | 150 | 190 | 210 |
| **175** | 210 | 255 | 100 |



# Clipping Function

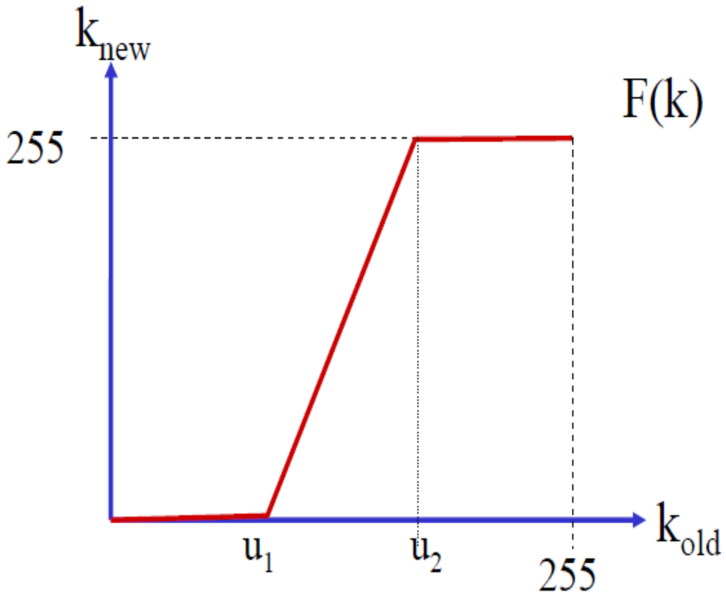



IIf most of the gray-levels in the image are in [u1 u2], the following mapping increases the image contrast.

For the given sub-image, if u1=50 and u2 =170, β=2

Find the new image after gray level slicing

| | | | |
|---|---|---|---|
| **0** | **10** | **50** | **100** |
| **5** | 95 | 150 | 200 |
| **110** | 150 | 190 | 210 |
| **175** | 210 | 255 | 100 |



# Clipping Function

21