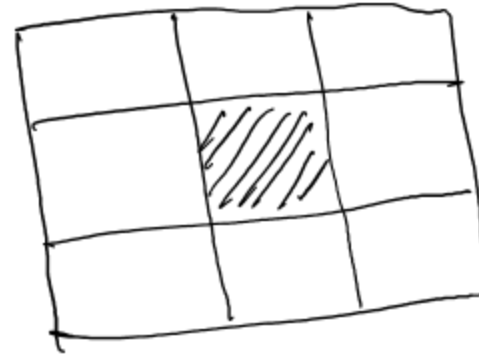


RGB



Grayscale



pixel



# Point Processes

Computational Photography (CSCI 3240U)

Faisal Z. Qureshi

<http://vclab.science.ontariotechu.ca>



# Acknowledgments

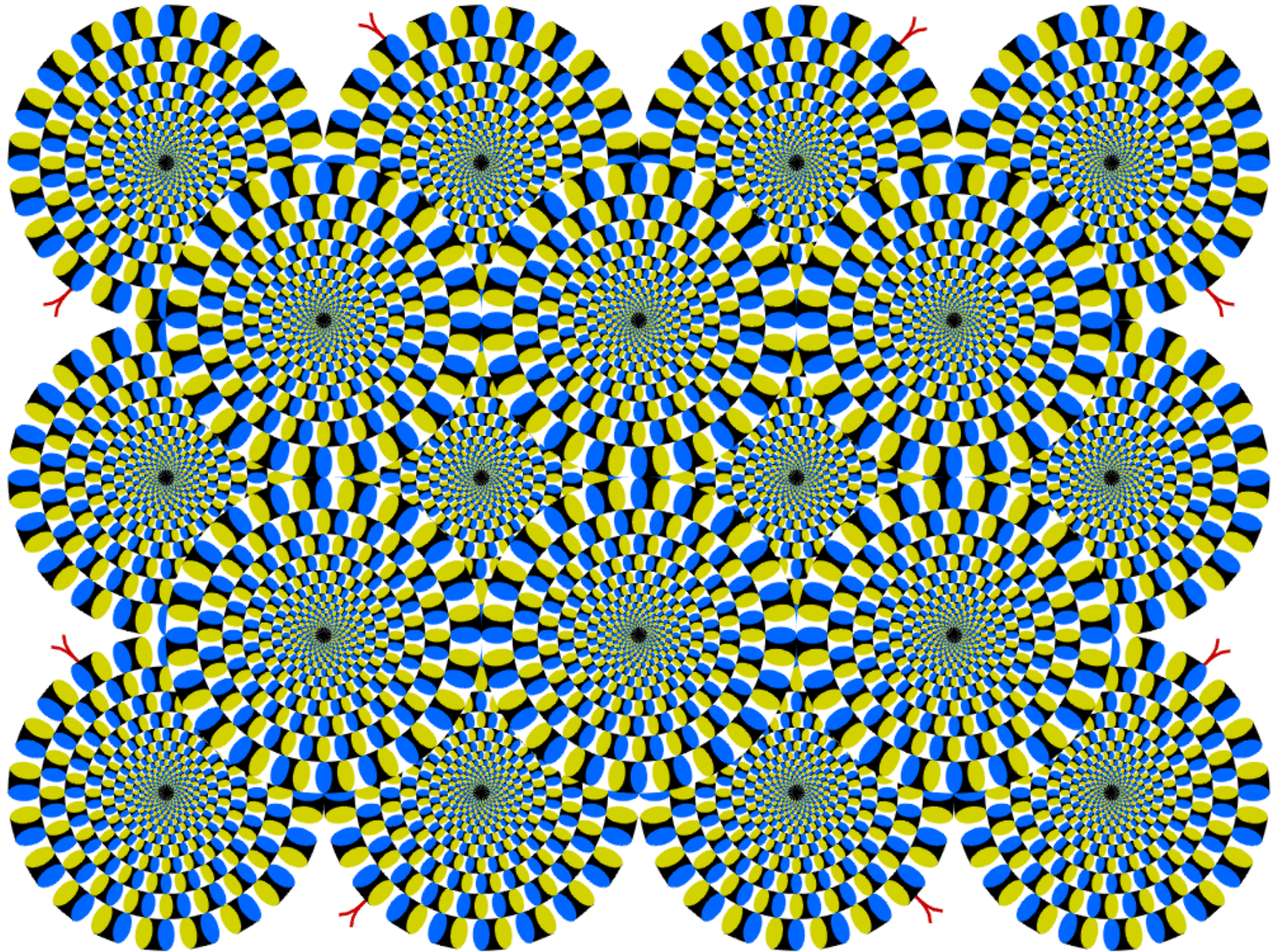
- These slides borrow and adapt materials developed by others, including
  - Michael Brown
  - Kyros Kutulakos
  - David Lindell
  - Gordon Wetzstein
  - Marc Levoy
  - Fredo Durand
  - Paul Debevec
  - Ramesh Raskar

# Slide credits

- A lot of inspiration and quite a few examples for these slides were taken directly from:
  - Kayvon Fatahalian (15-769, Fall 2016).
  - Michael Brown (CVPR 2016 Tutorial on understanding the image processing pipeline).
  - Marc Levoy (Stanford CS 178, Spring 2014).

# Special thanks to Ioannis Gkioulekas

- Many of the slides are taken with his permission from the computational photography course that he has developed at CMU



on lenses



# Image Enhancement

- Make an image more suitable for a **particular application** than the original image
- Types of techniques
  - Point processing
  - Spatial processing
  - Frequency domain processing

E.g., Human perception



# Image Enhancement

- Make an image more suitable for a **particular application** than the original image
- Types of techniques
  - Point processing
  - Spatial processing (**pixel neighbourhoods**)
  - Frequency domain processing

E.g., Human perception

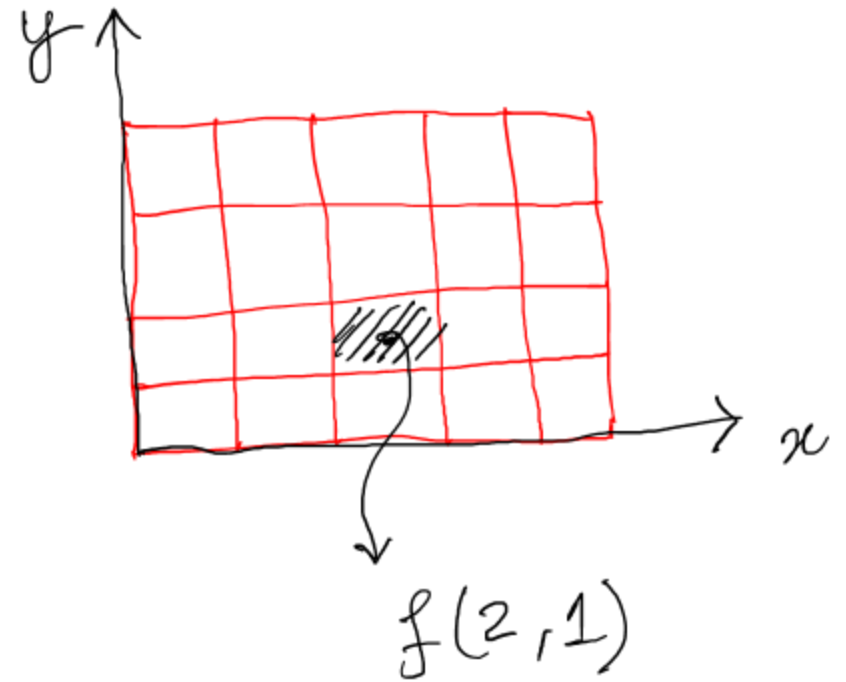


# Image Enhancement

- Make an image more suitable for a **particular application** than the original image
- Types of techniques
  - **Point processing** ← **Today's Focus**
  - Spatial processing (pixel neighbourhoods)
  - Frequency domain processing

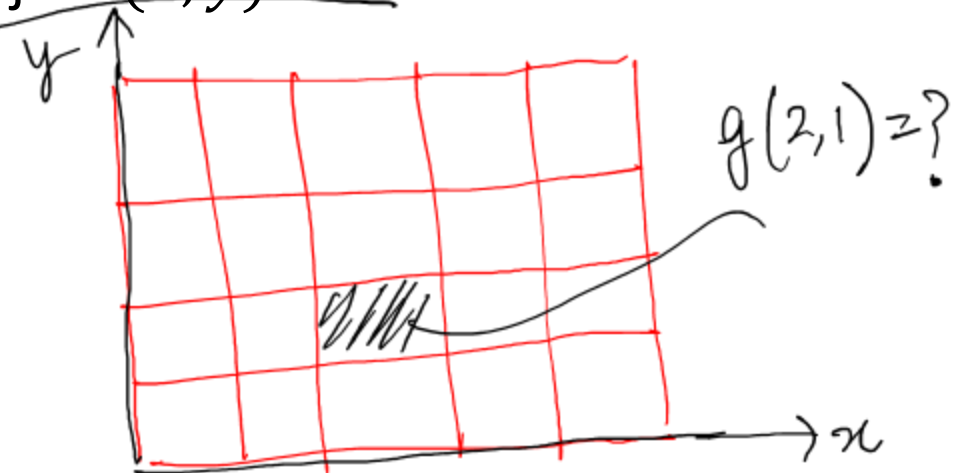
# Image Enhancement

- Input image:  $f(x, y)$
- Output image:  $g(x, y)$
- $T$  is an operator on  $f$  or a set of  $f$ 
  - $T$  is defined over some neighbourhood of  $(x, y)$
  - $T$  can operate over a set of images
  - For point processing the neighbourhood of  $(x, y)$  is just  $(x, y)$  itself



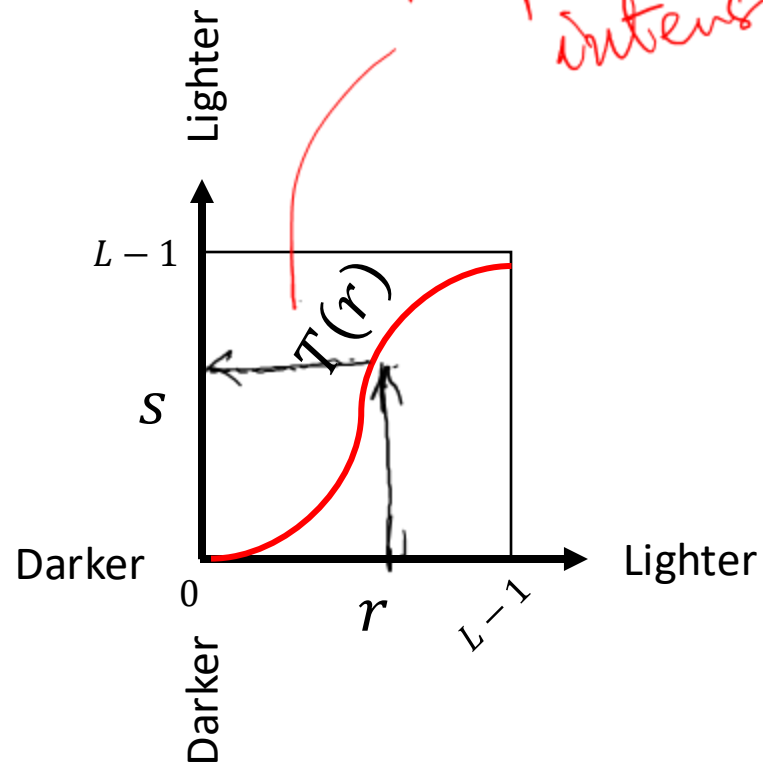
---

$$T(f(2,1)) \rightarrow g(2,1)$$



# Point Processing Example

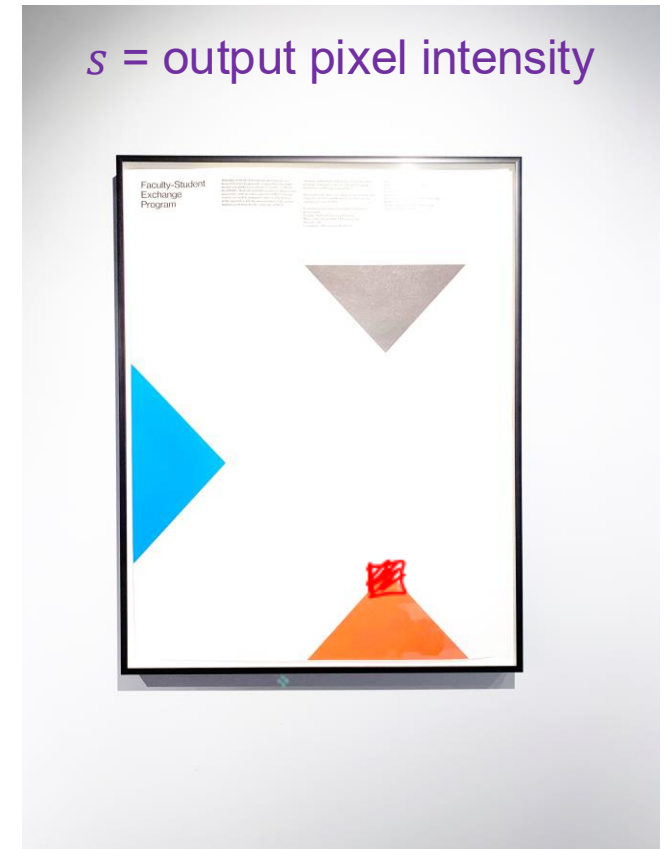
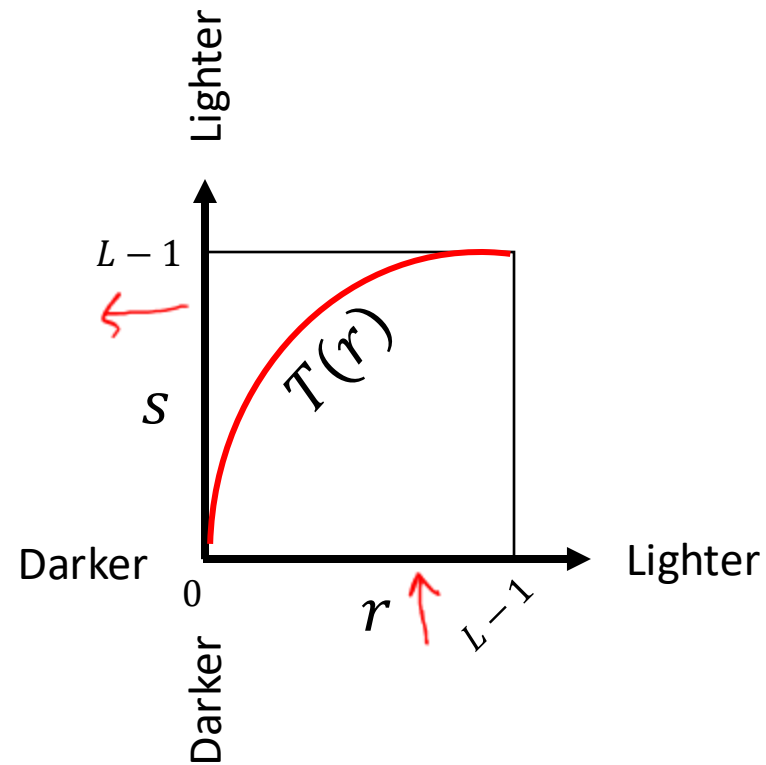
$r$  = input pixel intensity



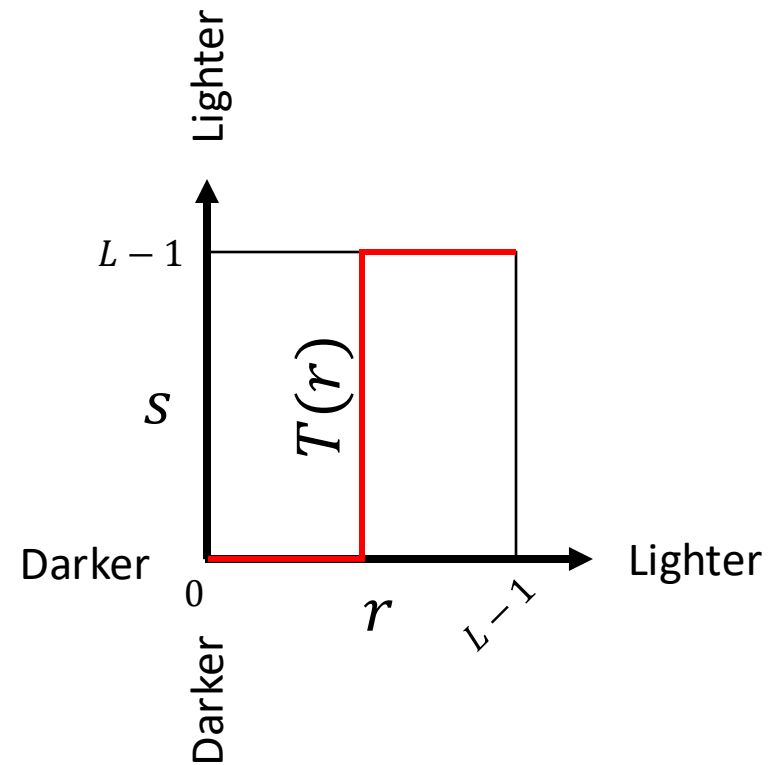
$s$  = output pixel intensity



# Point Processing Example



# Point Processing Example



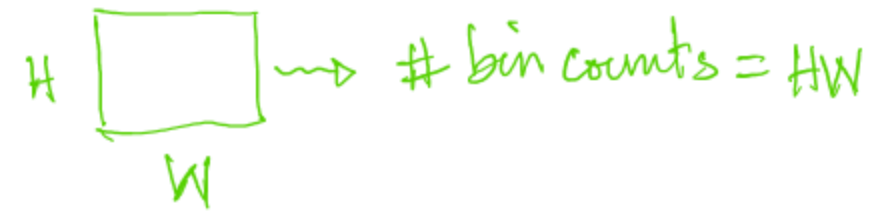
# Image Histogram

- Consider an  $H \times W$  image with  $L$  gray levels.
- Record the count  $n_k$  of pixels at each gray level  $r_k$  where  $k \in [0, L - 1]$
- The probability of a pixel at gray level  $r_k$  is

bin count for bin  $k$

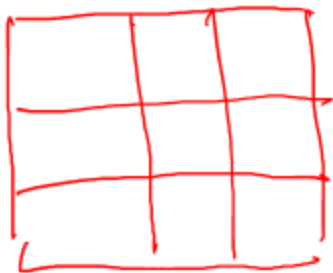
$$p(r_k) = \frac{n_k}{H \cdot W}$$

total pixels



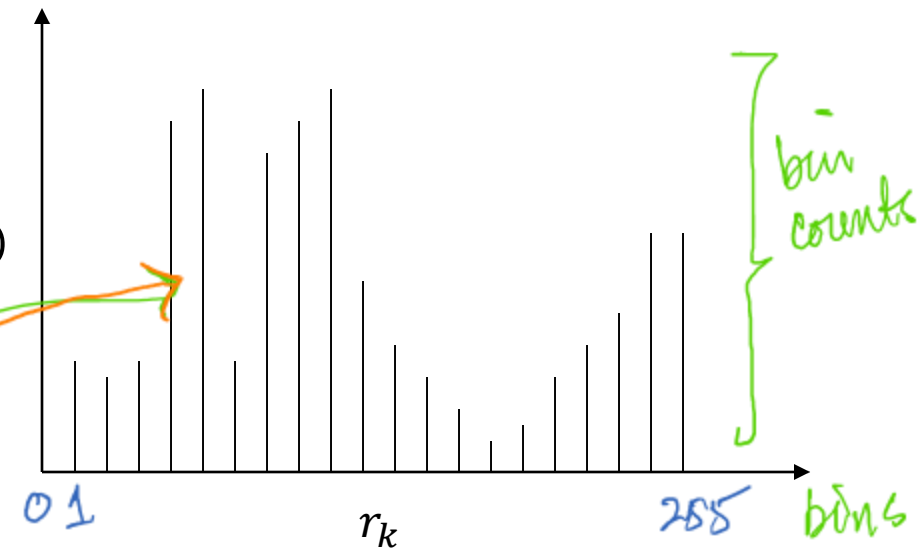
$r =$  input intensity

$\leftarrow$  8-bit ;  $2^8 = 256$

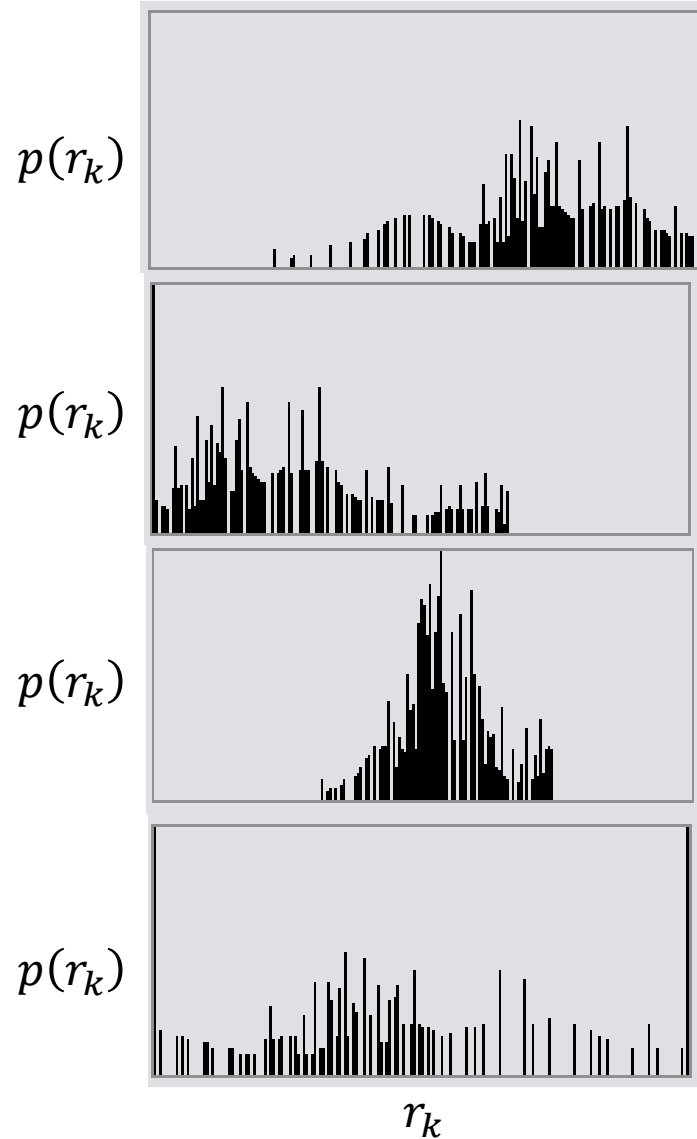


prob. dist.

$p(r_k)$



# Image Histogram



A



C



B

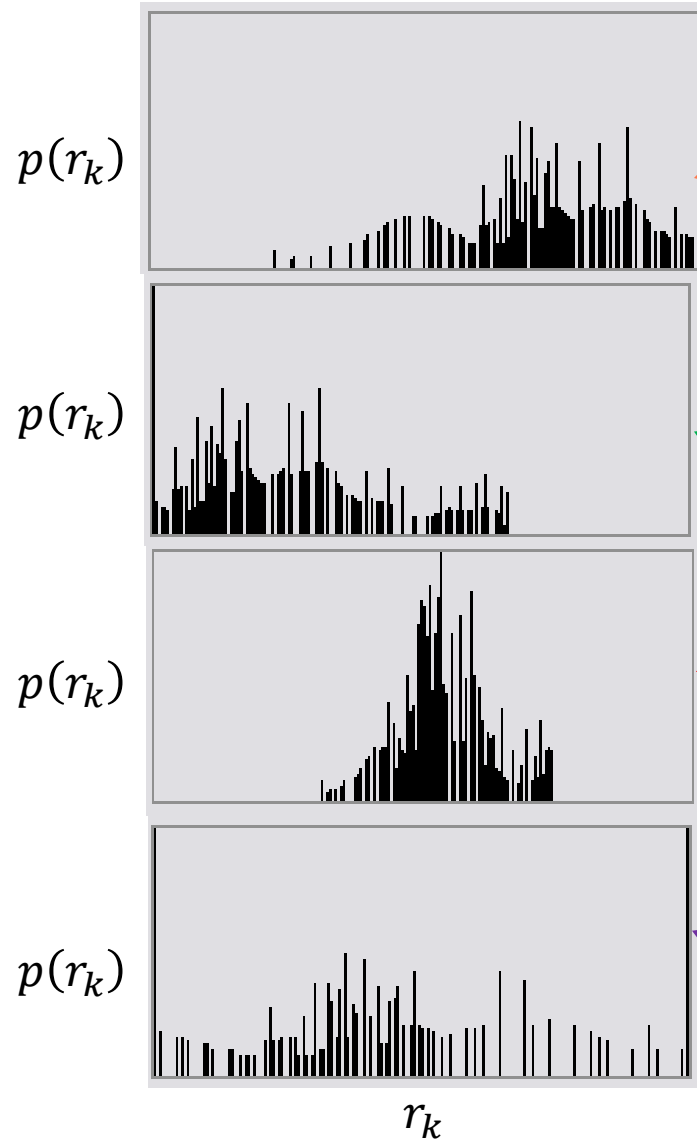


D

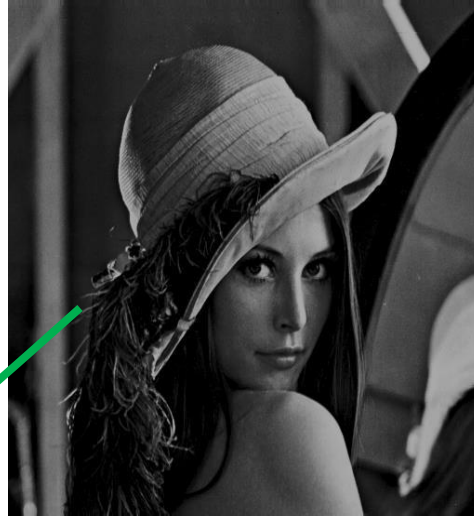


Match the image to its probably histogram

# Image Histogram



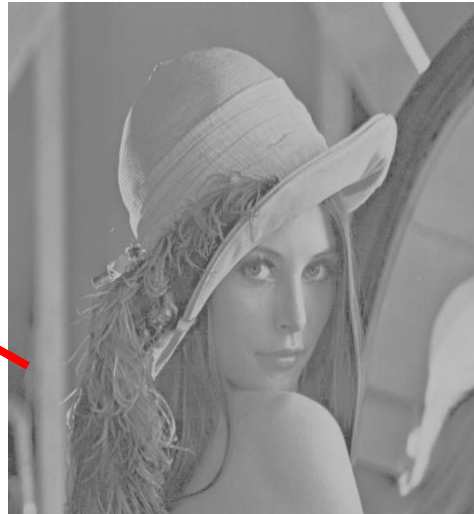
A



C



B



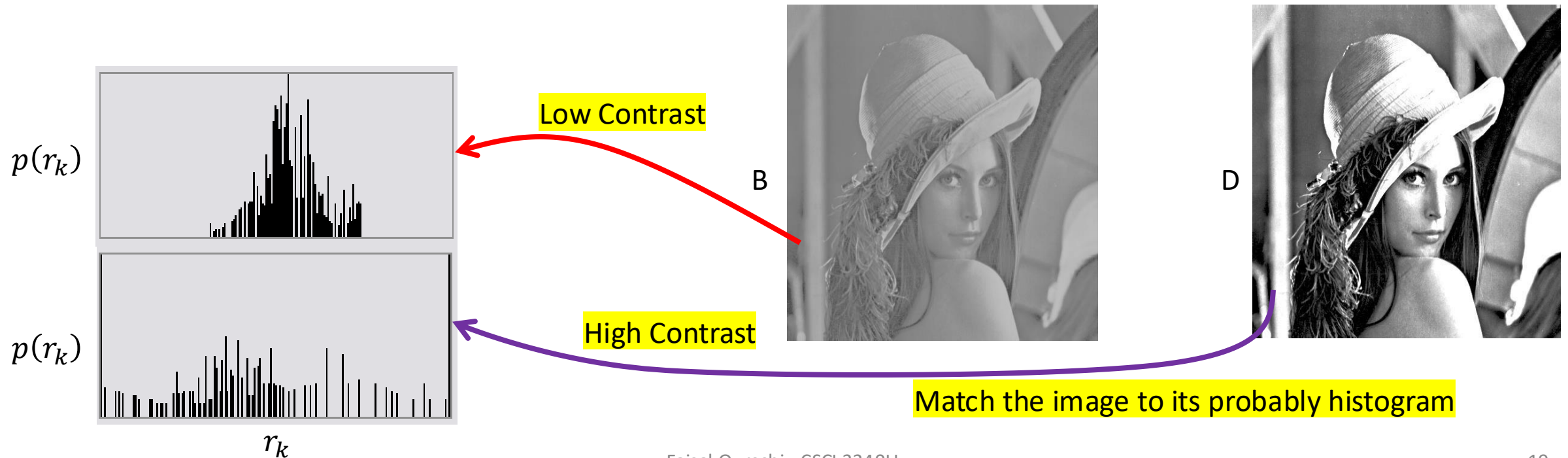
D



Match the image to its probably histogram

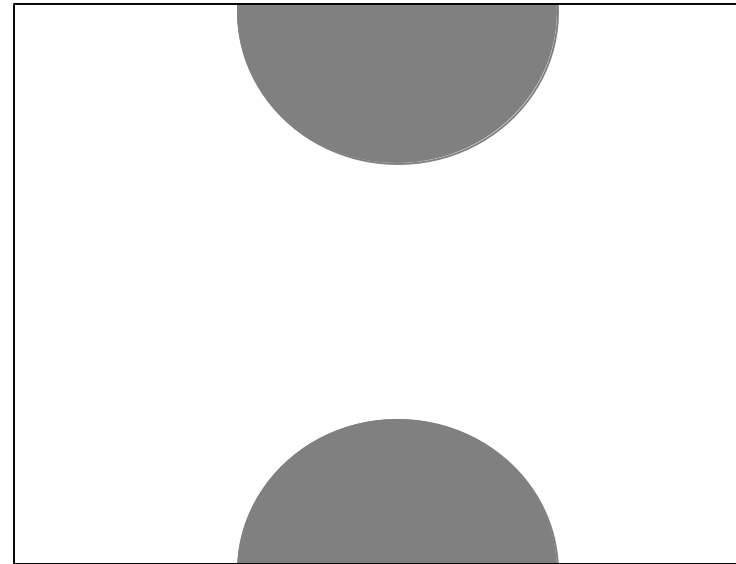
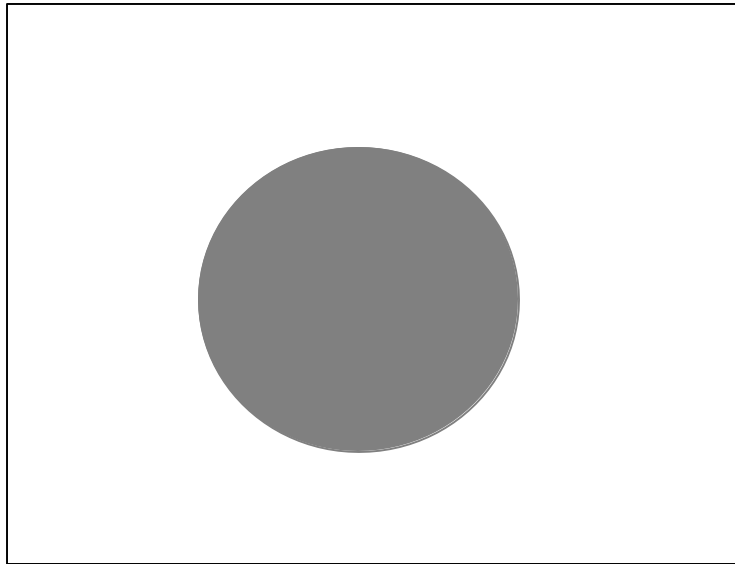
# Image Histogram

- Contrast is defined as the ratio of the **maximum intensity** to **minimum intensity**



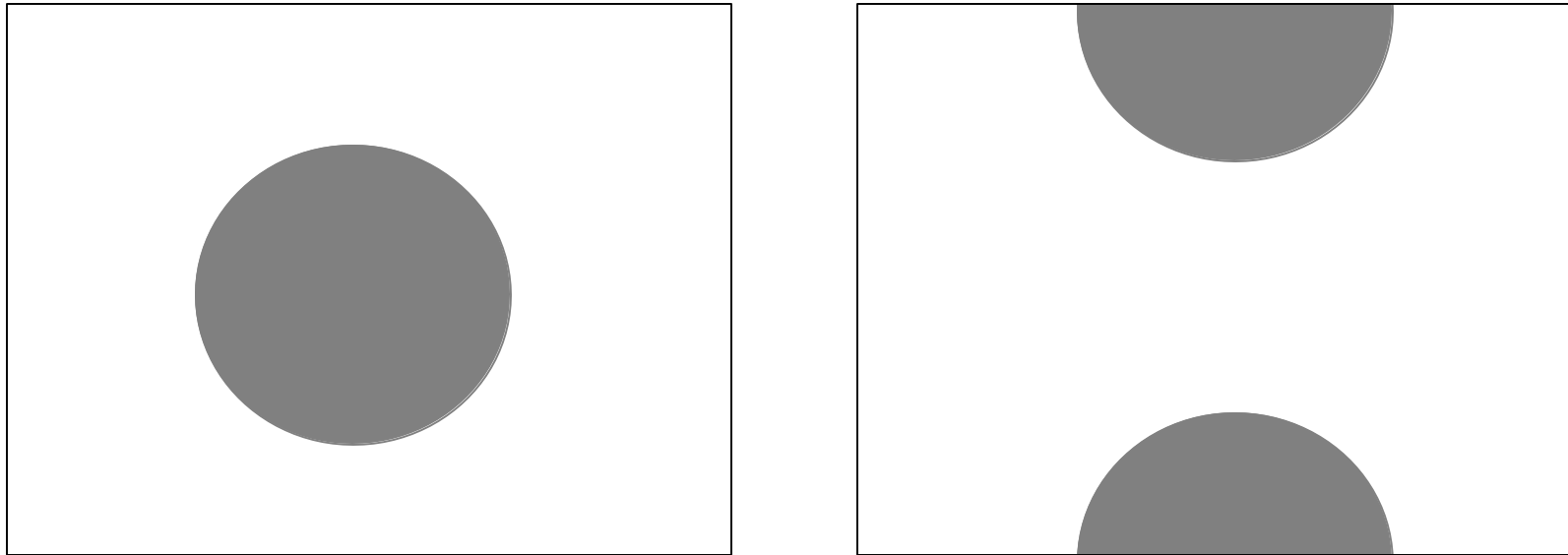
# Image Histogram

- Do the following two images have the same or different histograms?



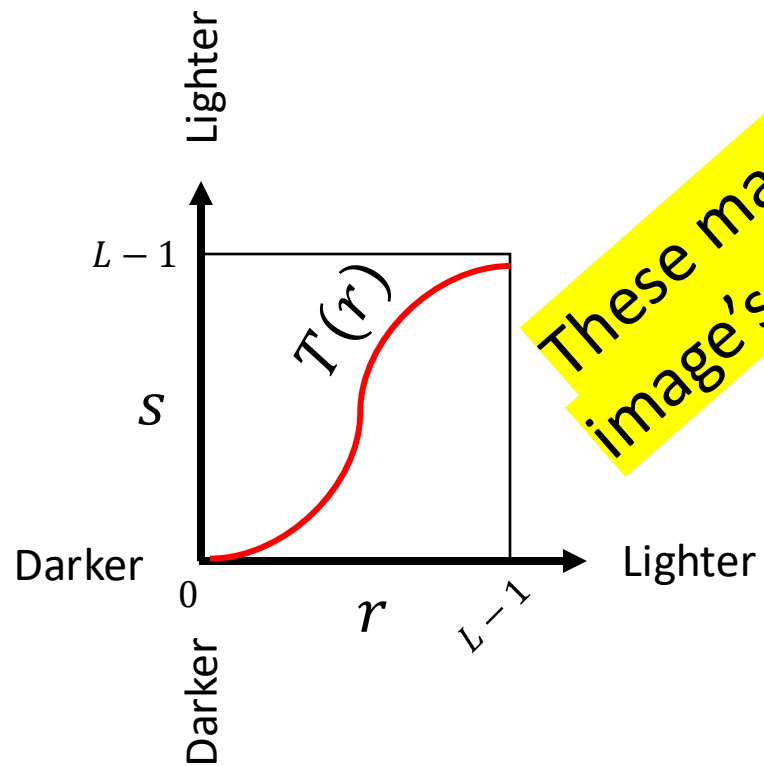
# Image Histogram

- Do the following two images have the same or different histograms?



Same. Histograms are just counts.  
These are not spatially aware.

# Adjustment Curves



These manipulate  
image's histograms

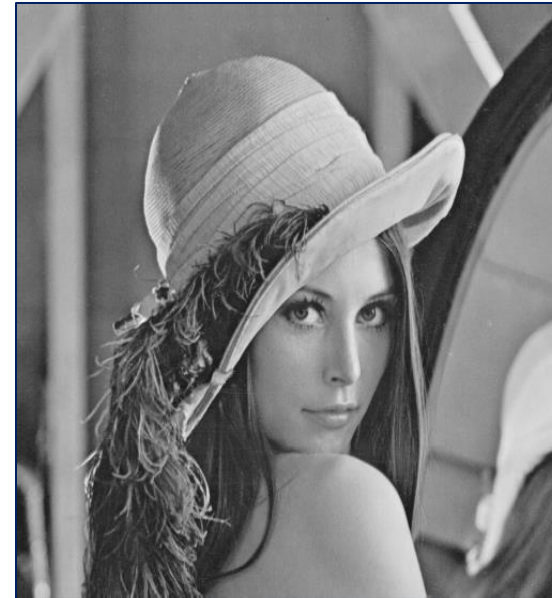


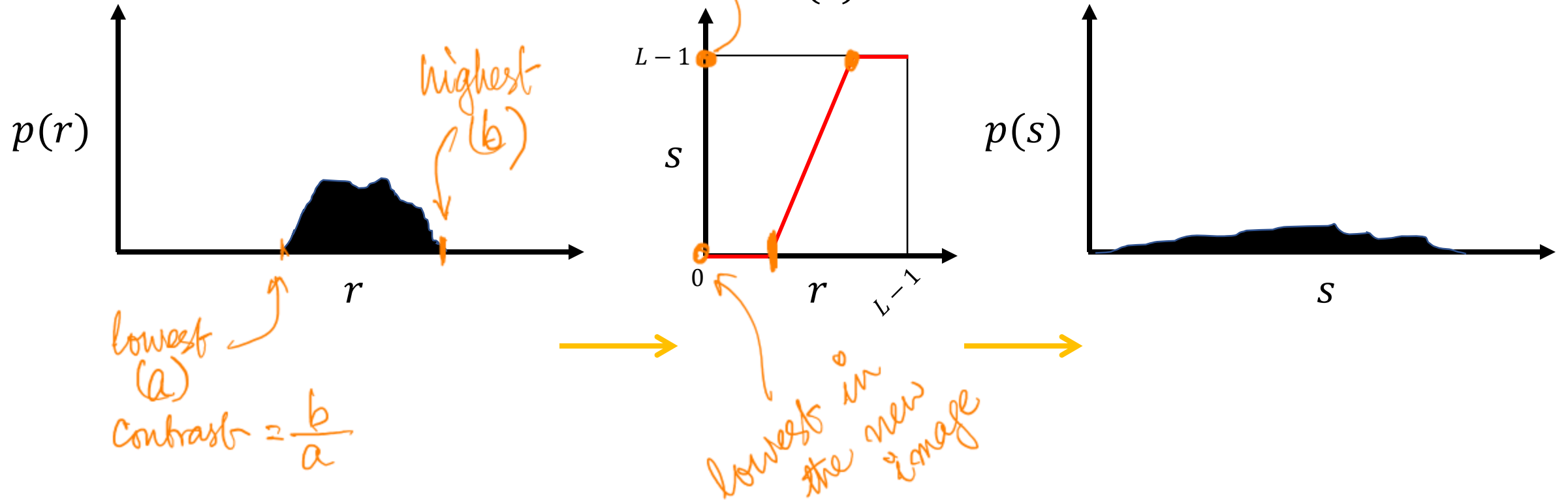
Image using pixels  $r$



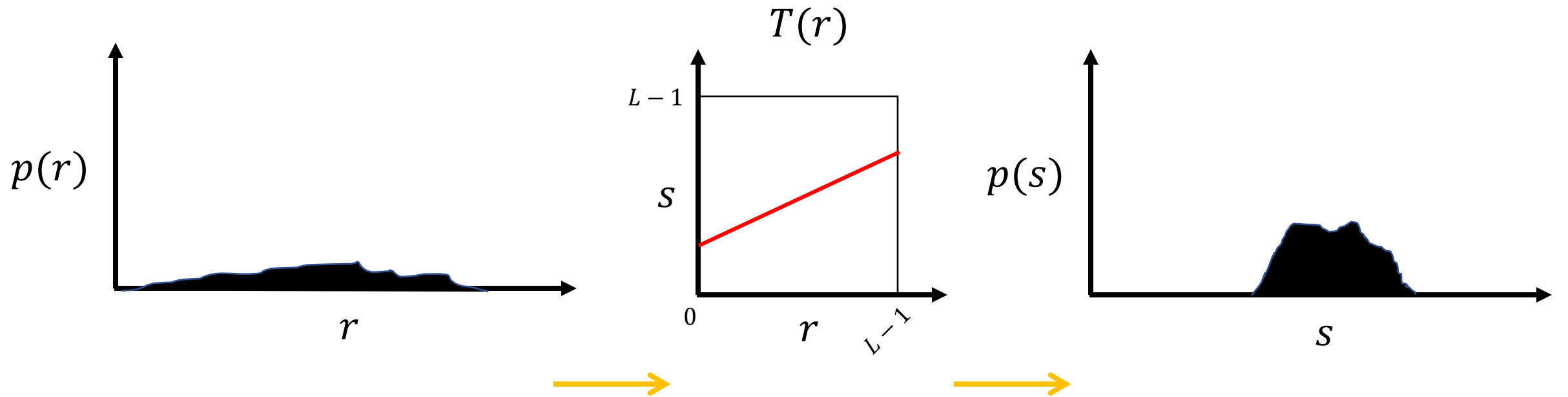
Image using pixels  $s$

$r$  = input pixel intensity  
 $s$  = output pixel intensity

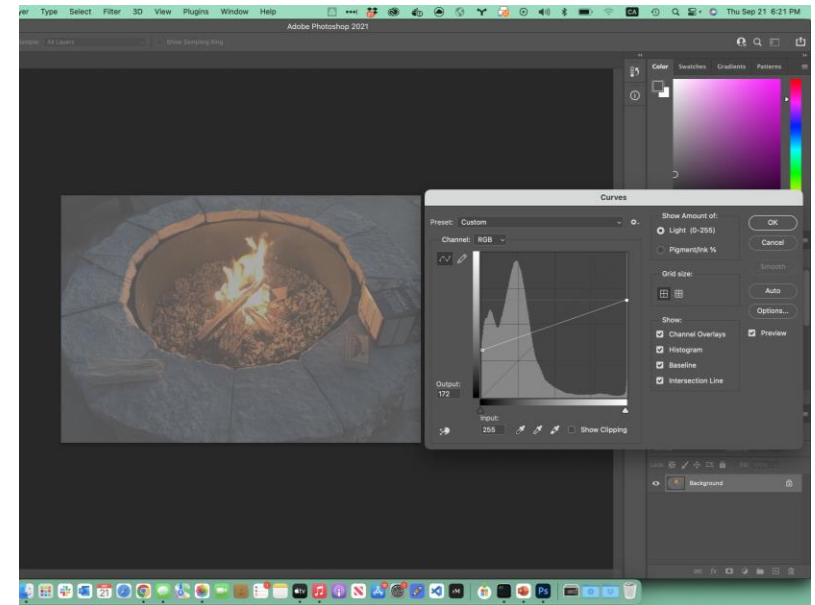
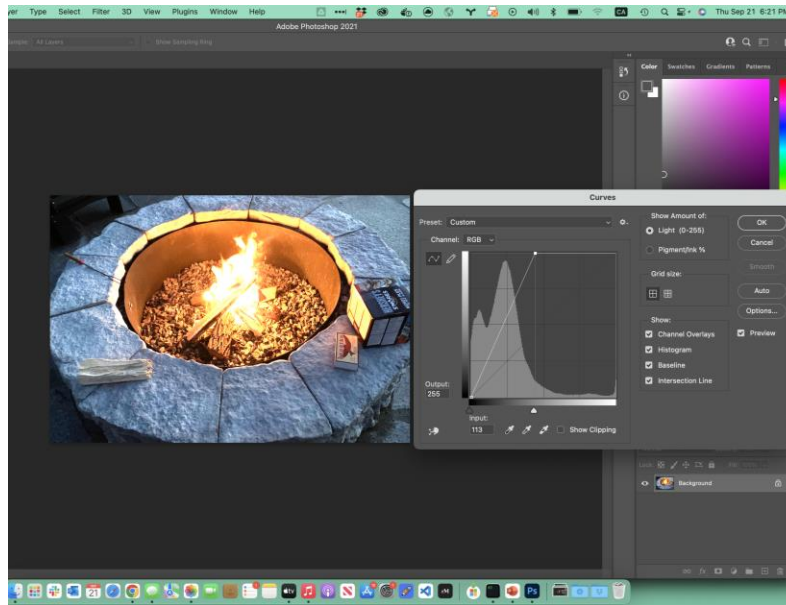
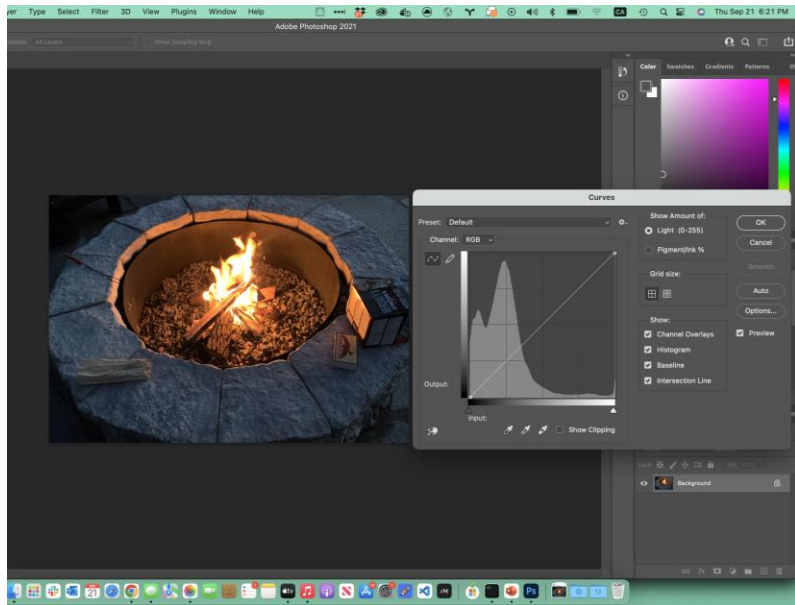
# Increase Contrast



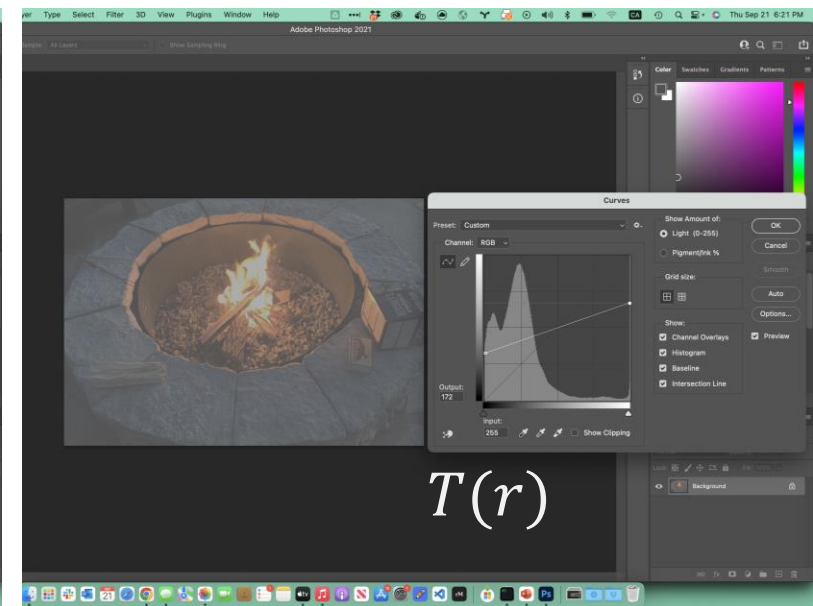
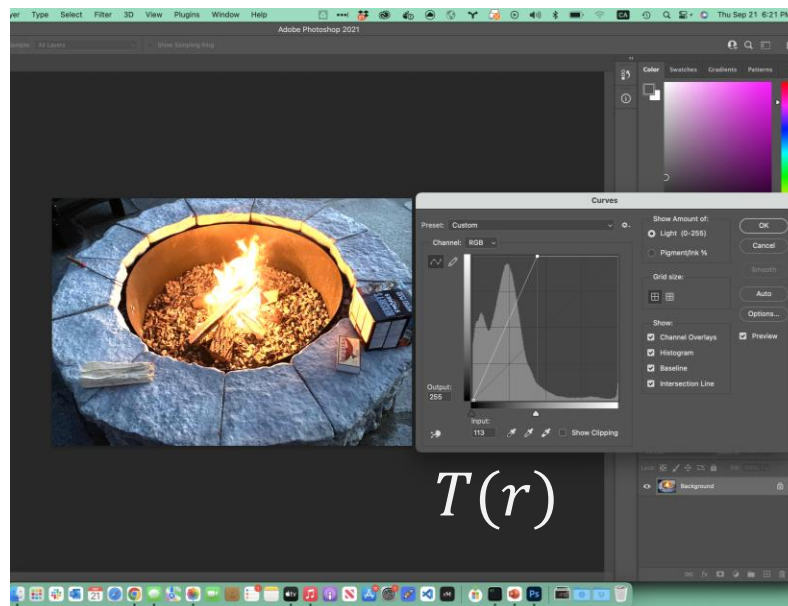
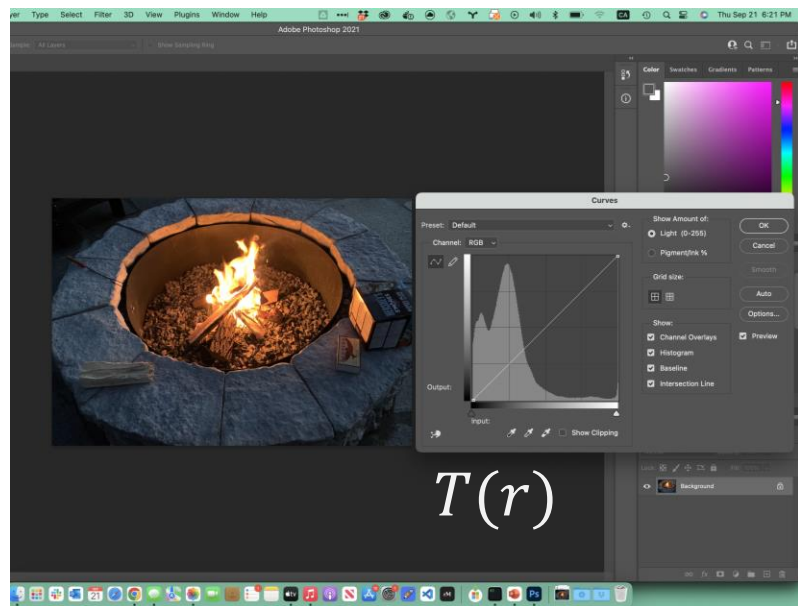
# Decrease Contrast



# Photoshop: Image > Adjustment > Curves

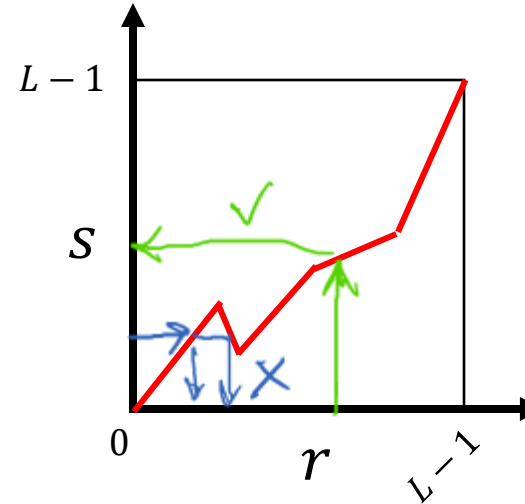


# Photoshop: Image > Adjustment > Curves

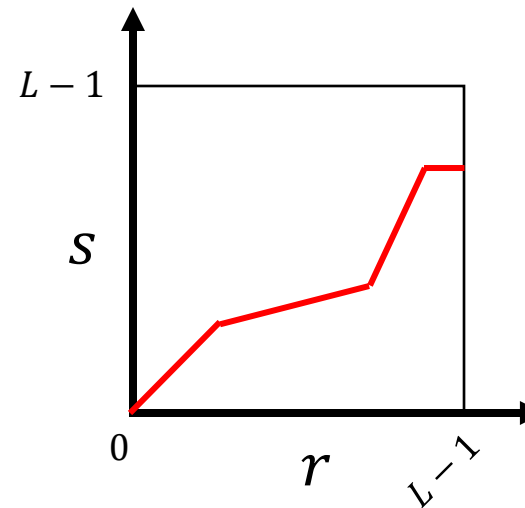


# Properties of $T(r)$

- Non-monotonic
  - Does not preserve gray level ordering
  - Looks unnatural
  - Does not have an inverse
- Monotonic
  - Preserves gray level ordering
  - Looks natural
  - Inverse exists



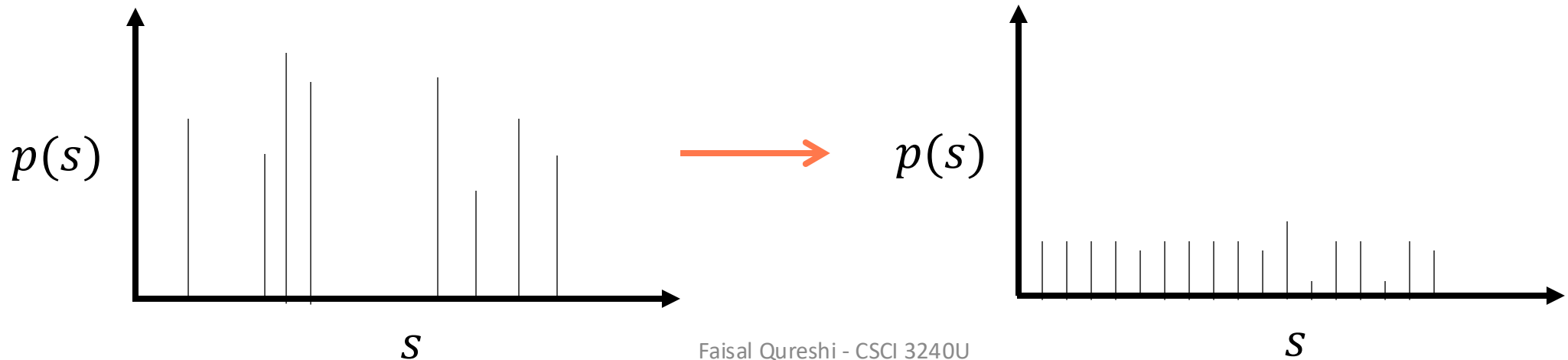
Non-monotonic



Monotonically increasing

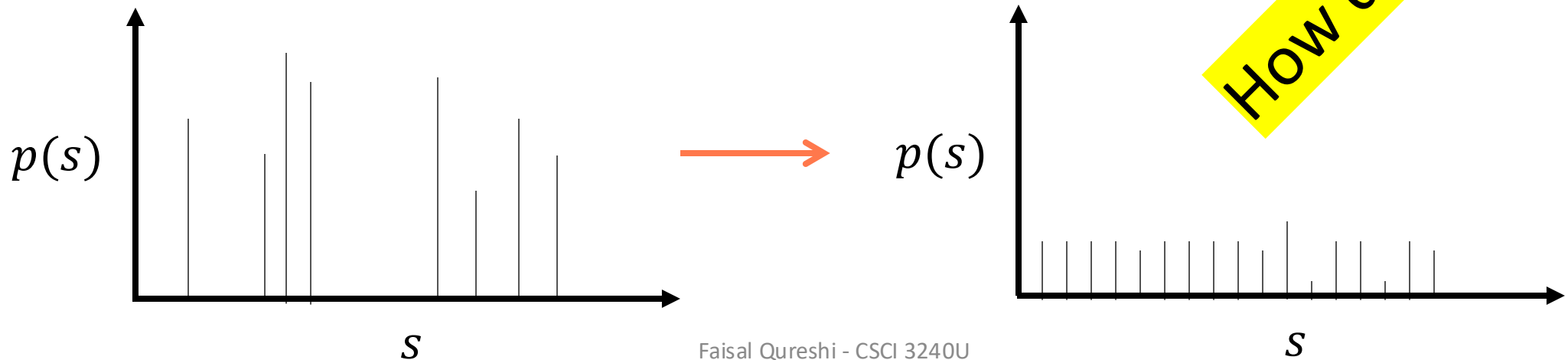
# Histogram Equalization

- Construct an image with equally many pixels at each gray level
  - Image is perceptually pleasant (nice to look at)
  - Pixel resources are maximized, so to speak
- Such an image will have an equal histogram
- Counts of pixels at each level  $r_k$  will be the “same”
  - The counts will be *roughly* equal to  $(H \cdot W)/L$



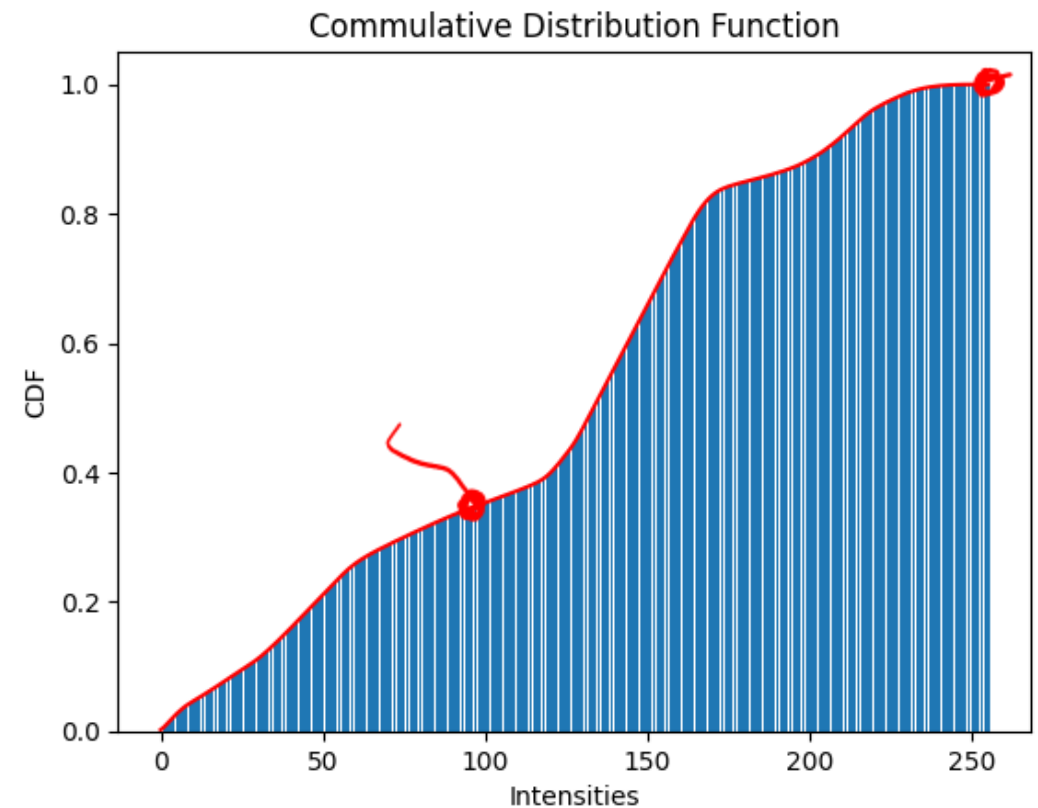
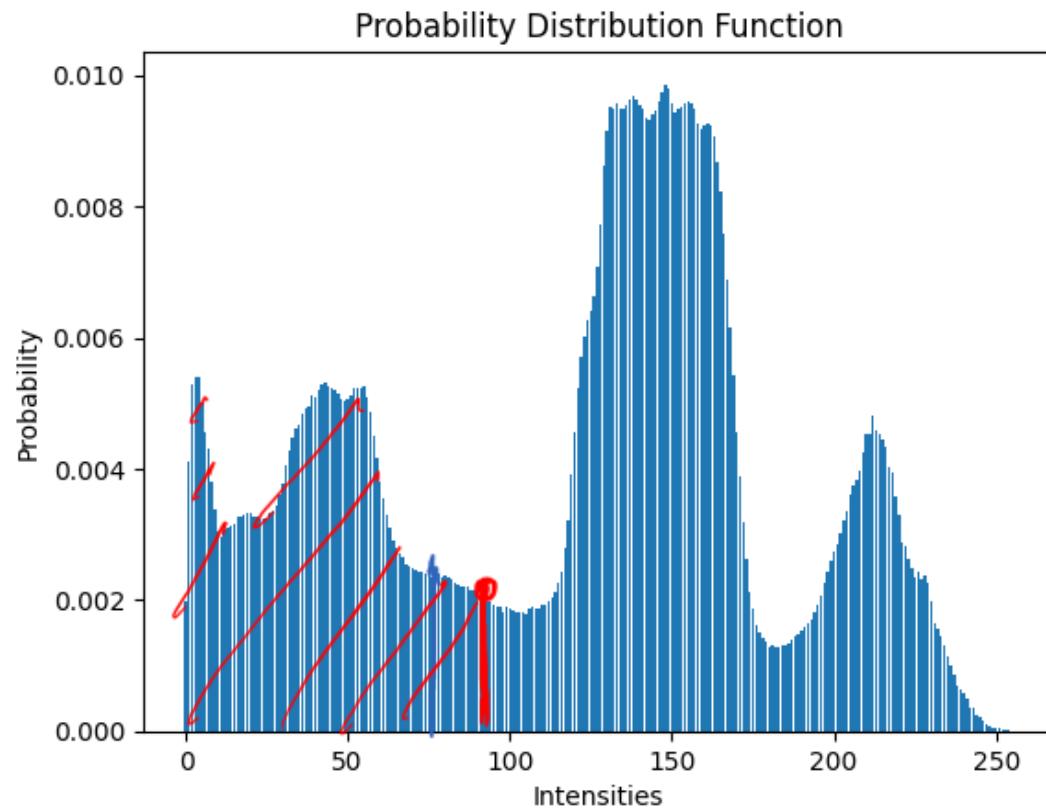
# Histogram Equalization

- Construct an image with equally many pixels at each gray level
  - Image is perceptually pleasant (nice to look at)
  - Pixel resources are maximized, so to speak
- Such an image will have an equal histogram
- Counts of pixels at each level  $r_k$  will be the “same”
  - The counts will be *roughly* equal to  $(H \cdot W)/L$



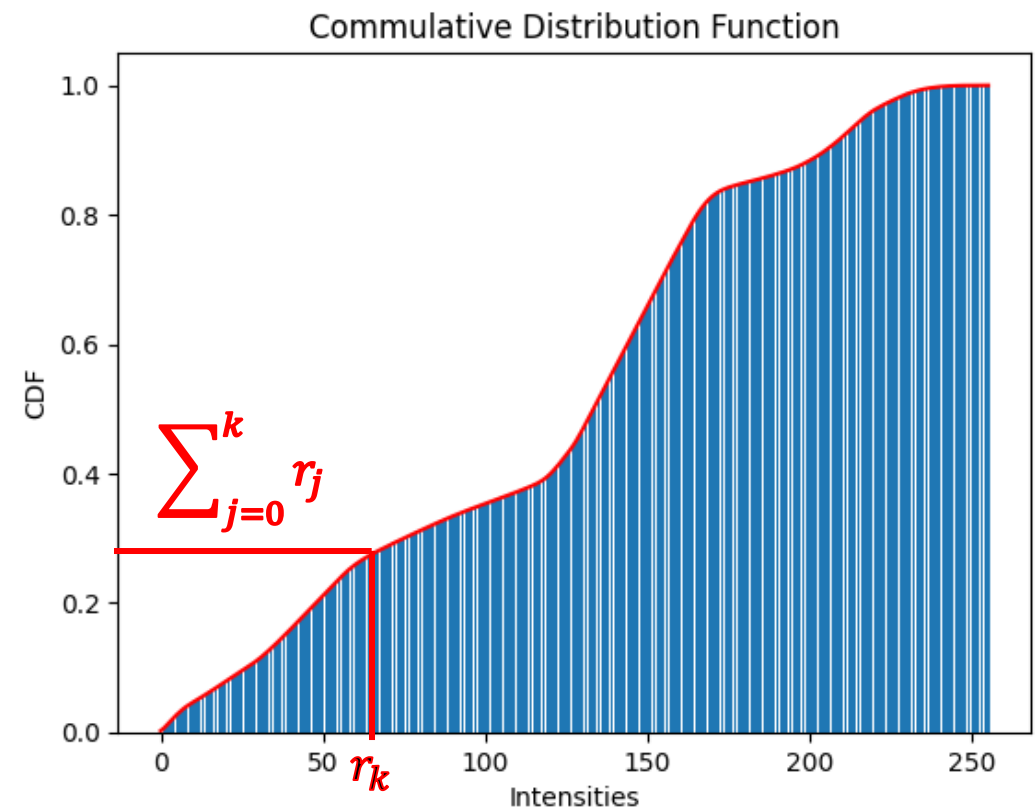
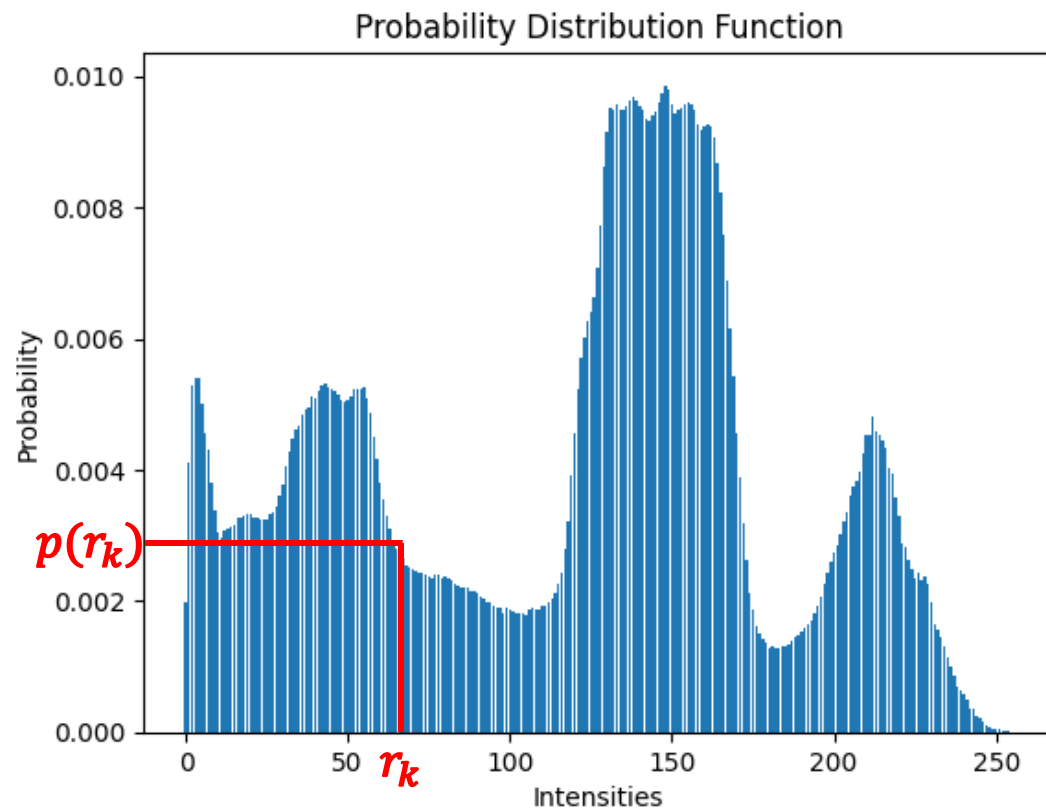
# Histogram Equalization

- Use **cumulative distribution function** to construct the necessary  $T(r)$



# Histogram Equalization

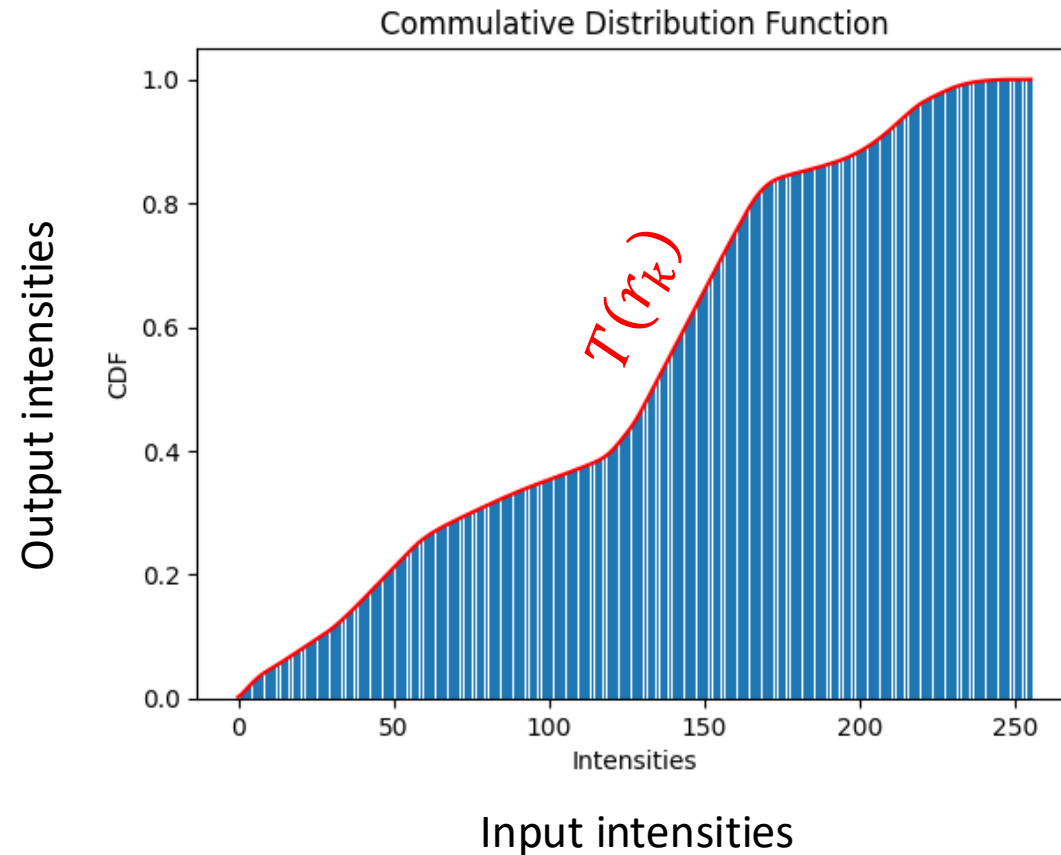
- Use **cumulative distribution function** to construct the necessary  $T(r)$



# Histogram Equalization

- Use **cumulative distribution function** to construct the necessary  $T(r)$

$$\begin{aligned} s_k &= T(r_k) \\ &= \frac{1}{H \cdot W} \sum_{j=1}^k n_j \\ &= \sum_{j=1}^k p(r_j) \end{aligned}$$

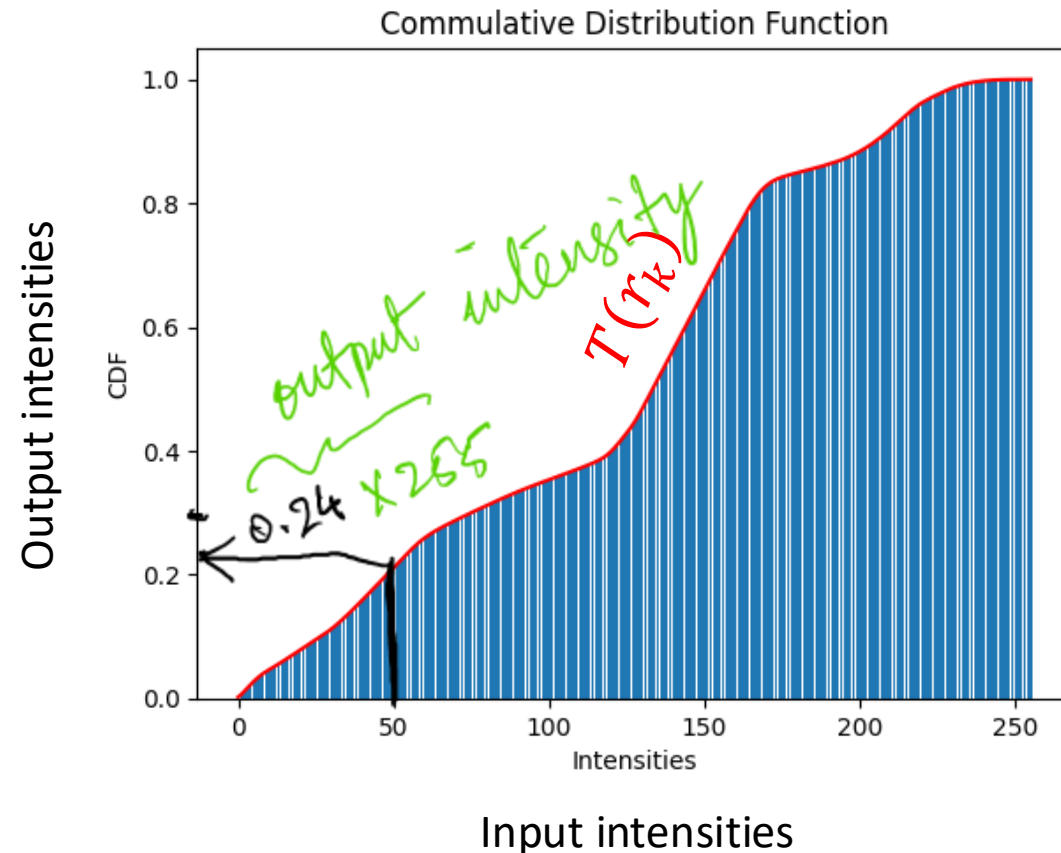


# Histogram Equalization

- Use **cumulative distribution function** to construct the necessary  $T(r)$

$$\begin{aligned} s_k &= T(r_k) \\ &= \frac{1}{H \cdot W} \sum_{j=1}^k n_j \\ &= \sum_{j=1}^k p(r_j) \end{aligned}$$

Input intensities fall between 0 and L-1, whereas output intensities are between 0 and 1?



# Histogram Equalization

- Use **cumulative distribution function** to construct the necessary  $T(r)$

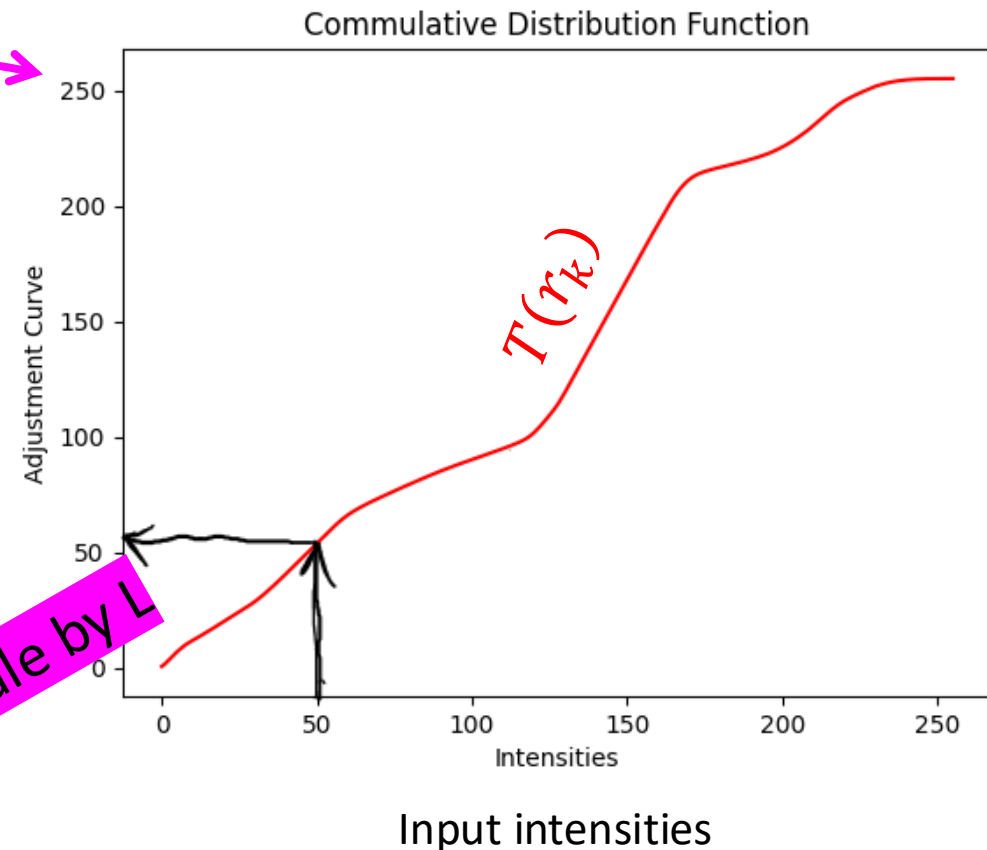
$$\begin{aligned} s_k &= T(r_k) \\ &= \frac{1}{H \cdot W} \sum_{j=1}^k n_j \\ &= \sum_{j=1}^k p(r_j) \end{aligned}$$

Input intensities fall between 0 and L-1, whereas output intensities are between 0 and 1?

Scale by L



Output intensities



# Histogram Equalization

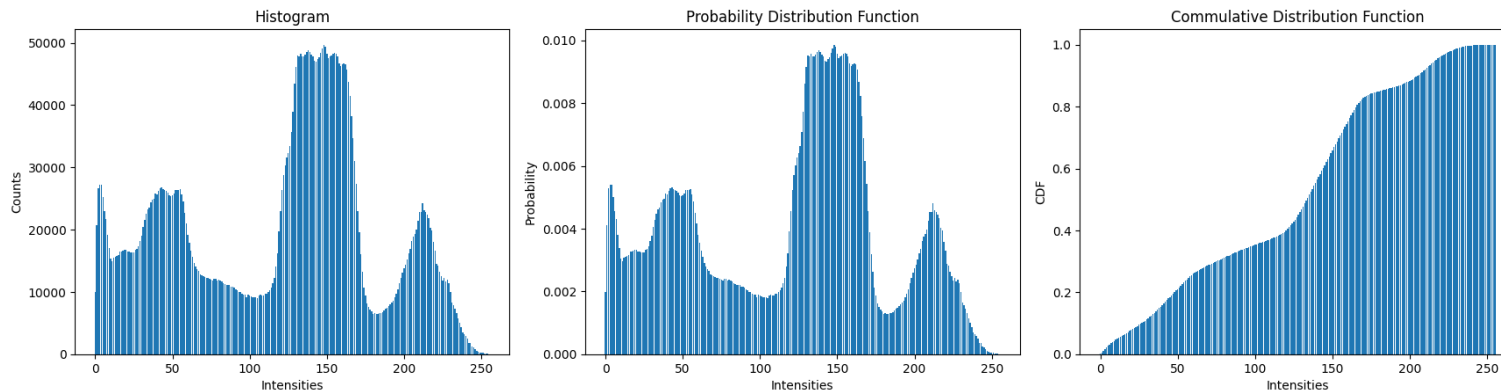
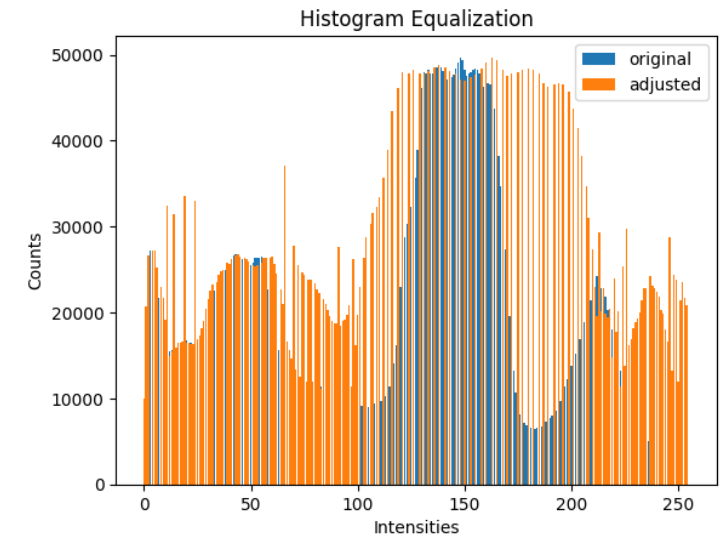
1944 x 2592 - uint8



1944 x 2592 - uint8



## Histogram before and after adjustment



# Histogram Equalization

- Can significantly improve image appearance
- Automatic
  - Derived fully from the input image
- Often used as a pre-processing step
  - Accounts for lighting variations (somewhat)
  - Accounts for camera/device characteristics (somewhat)
  - Helps with image comparison
- It is possible to “recover” the original since  $r = T^{-1}(s)$  exists (at least in theory)
  - Assuming a reasonable distribution of gray scales in the original image
  - This won't work if the original image was black-and-white

# Summary

- Point processes for image enhancement
- Adjustment curves
- Histogram equalization

# Something to Think About

- How would you use what you have learned in this lecture to develop tools to enhance the appearance of color images?