

---

# Intensity Transformation

Team 11: Austin Vu, Zhao Li, Tymothee Jones,  
Duong Nguyen, Jason Honea, Amanjit Singh,  
Zhuoran Cao

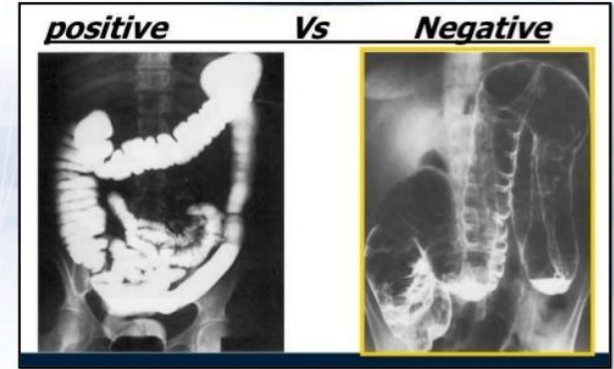
---

# Our Objectives for this Topic

- Create a GUI that can properly implement the algorithms for intensity transformation.
- Make sure each subtopics are functioning smoothly with “no” bugs in GUI and with all essential features applied.
- Leave an impression about the importance of intensity transformation and how it plays in real-world applications.
- Add a feature on GUI that’ll suggest a method based on the image to make it clearer. By analyzing the histogram of the image and offer a method that fits the user’s liking.

# Motivations for Intensity Trans.

- Medical industry:
  - Altering pixels in MRI scans, X-Rays, etc. to give a clearer image for medical analysis using image negative, histogram equalization, and contrast stretch.
- Devices such as phone, TV, or monitor:
  - Gamma correction modifies the brightness due to luminance not displayed correctly using gamma transformation.



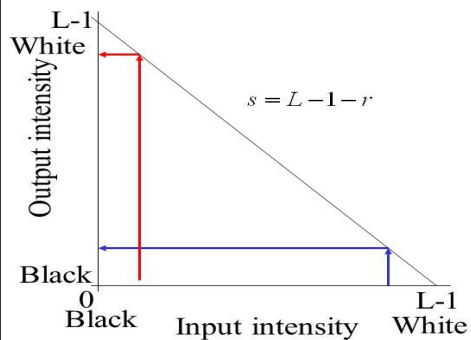


# Subtopics

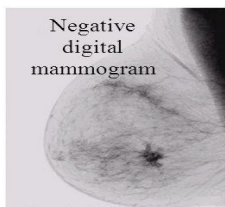
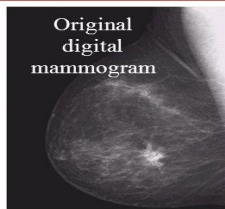
Features we've implemented into our project.

- Image Negative
- Histogram Equalization.
- Histogram Matching
- Histogram Shaping
- Contrast Stretch
- Gamma and Log Trans.
- GUI Demo

## Image Negative



$L$  = the number of gray levels



(Images from Rafael C. Gonzalez and Richard S. Woods, Digital Image Processing, 2nd Edition)

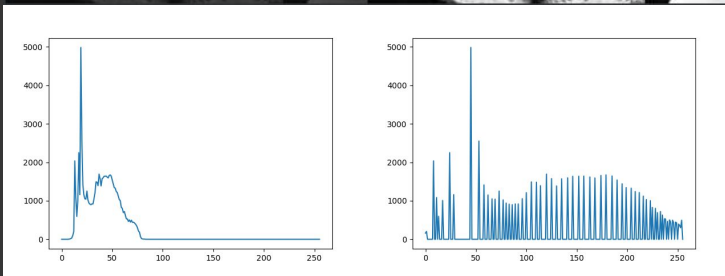
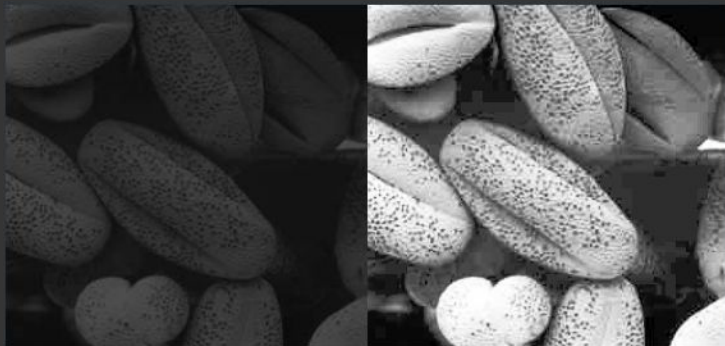
# Image Negative

## Image Negative

The lower the input pixel value is, the higher the output pixel value will be. In general, the darker the pixel, the brighter it is when negative.

```
3
4
5 def imgNegative(image):
6     # Inputs image and returns a negative
7     NegImage = np.zeros(image.shape, np.uint8)
8     #cv2.imshow(image)
9     #cv2.waitKey(0)
10
11     for i in range(0, image.shape[0]):
12         for j in range(0, image.shape[1]):
13             # S (output pixel) = L (size) - 1 - r (input pixel)
14             NegImage[i, j] = image.shape[0] - 1 - image[i, j]
15
16     return NegImage
```

# Histogram Equalization



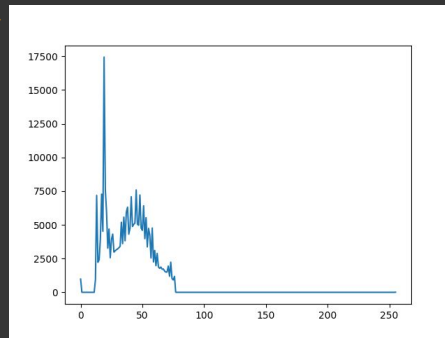
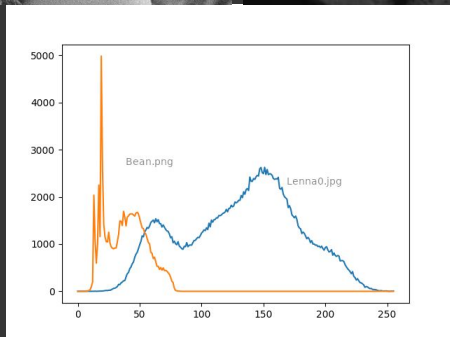
Increases contrast by flattening the histogram.

# Histogram Matching

Source

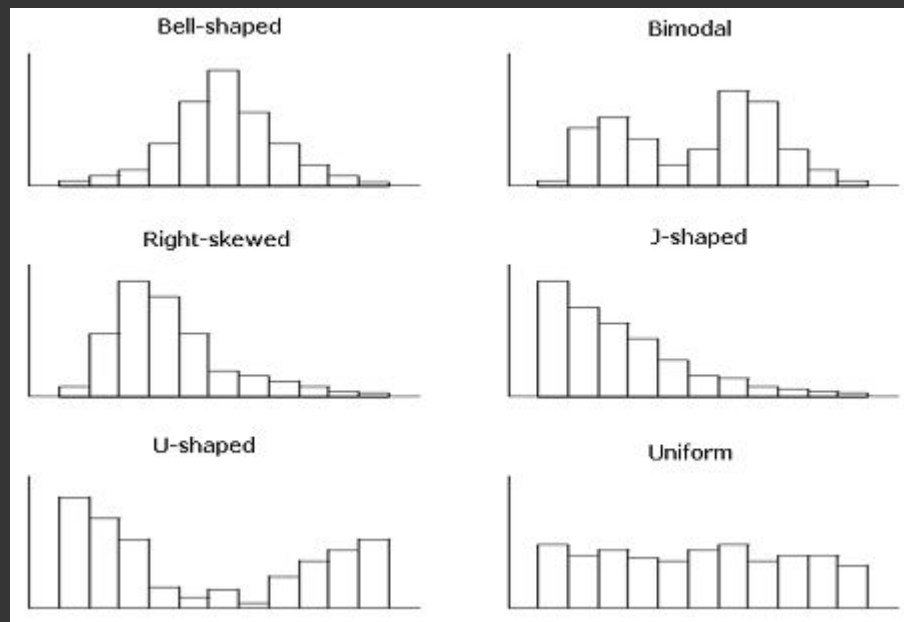


Target



Matches the histogram of the source image to that of the target image.

# Histogram Shaping



## Histogram Shaping

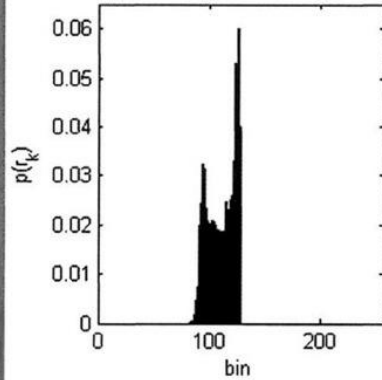
Modifying an image by altering its histogram to match another histogram shape. Different methods of shaping results in different results of the digital image.



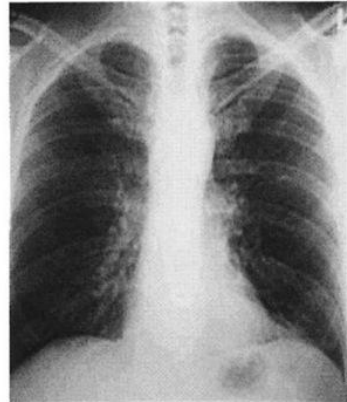
# Contrast Stretch



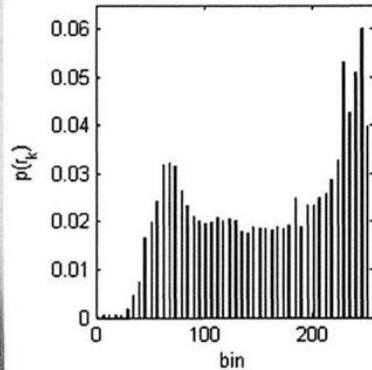
(a)



(b)



(a)



(b)

## Contrast Stretch

Also called "Normalization," enhances image contrasts by stretching the range of intensity values. Works best on skewed histogram or histogram that's not well-distributed..

# Log Transformation

ORIGINAL



C = 1



C = 2



C = 5

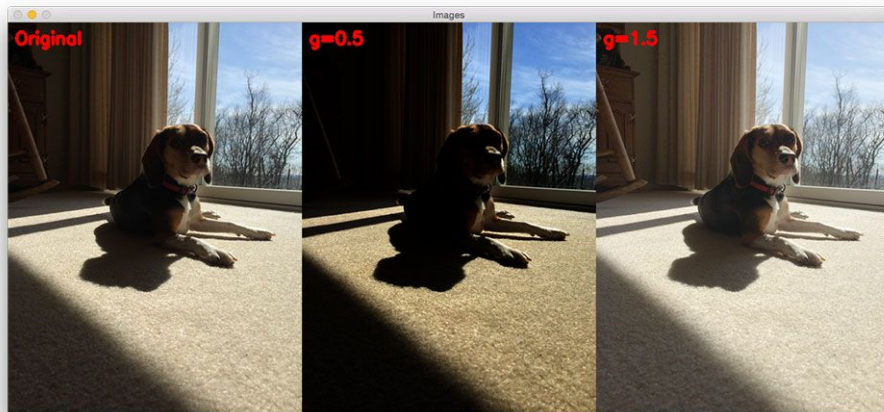


```
1  from __future__ import print_function
2
3  class LogTransform:
4      def log_transform(self, image, c):
5          self.rows = image.shape[0] #number of rows
6          self.columns = image.shape[1] #number of columns
7
8          self.resulted_img = np.zeros(shape = image.shape)
9
10         for i in range(self.rows):
11             for j in range(self.columns):
12                 self.resulted_img[i,j] = c*np.log(image[i,j] + 1)
13
14
15         return self.resulted_img
```

## Log Transformation

Increase the detail (or contrast) of lower intensity values. Higher the c value, brighter the image.

# Gamma Transformation



```
1 import numpy as np
2
3 def powerLaw(image, gamma):
4
5     c = 255
6     new_image = np.array(c*(image/c)**gamma , dtype=np.uint8)
7     return new_image
8
```

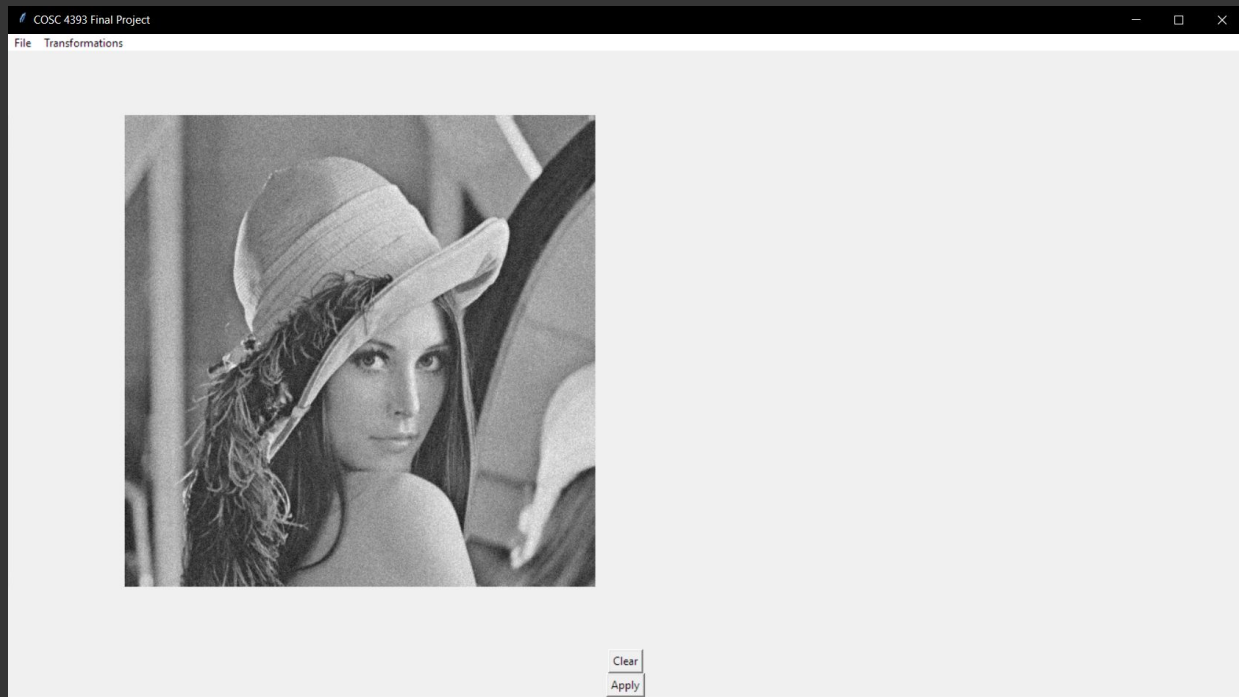
## Gamma Trans.

Used to correct the power-law response phenomena found in display and capture devices. Gamma  $> 1$  produces a darker image, while gamma  $< 1$  produces brighter images

# GUI Demo

## Further work:

- Implement histogram matching
- Log transform
- PL gamma input
- “Suggestions” onto GUI
- Save feature
- Fix remaining bugs
- Clean up GUI



# What Needs To Be Done

		In Progress	Completed	Tasks
1	Few algorithms that needs to be applied to the GUI	✗	✓	Creation of all algorithms for each subtopics
2	Implementation of a feature that suggests an intensity trans.	✗	✓	Ideas for the GUI and what we want to have for the GUI.
3	Completion of the report that records our findings	✗	✓	Outline and information for the report such as challenges and process of creating the algorithm



**THANK YOU**

**Q&A TIME**