

# Creating and Plotting Histograms

- Use any grayscale image
  - <http://scikit-image.org/docs/dev/api/skimage.io.html#skimage.io.imread>
  - Convert RGB to grayscale using following methods:
    - <http://scikit-image.org/docs/dev/api/skimage.color.html#skimage.color.rgb2gray>
  - Convert an image to unsigned byte format, with values in [0, 255] using `img_as_ubyte`
    - <http://scikit-image.org/docs/dev/api/skimage.html>
- Create a 1D array of size 256
  - Initialize with all zeros (`np.zeros`)
- Iterate through the image and for each pixel value, increment the corresponding histogram array element by one.
  - e.g. if pixel value=96, increment `hist[96]` by one
- Use pyplot to plot the histogram.
  - Tutorial:
    - [http://matplotlib.org/users/pyplot\\_tutorial.html](http://matplotlib.org/users/pyplot_tutorial.html)

# Histogram Equalization

- Compute the cumulative histogram
- Use the following formula to replace each pixel value:

$$a' = \text{floor}\left[\frac{K-1}{MN} H(a) + 0.5\right]$$

- $a'$ : new pixel value
- $K$ : no. of intensity levels (typically 256)
- $M, N$ : image dimensions (height and width)
- $H(a)$ : cumulative histogram

# Histogram Matching

- Normalize the histograms
  - Divide each histogram by the sum of all its entries (i.e. total no. of pixels in the image)
- Use Bhattacharya Coefficient

$$BC(p, q) = \sum_{i=0}^{K-1} \sqrt{p(i)q(i)}$$

- p, q are the normalized histograms of the two images
  - summation is to be done over all gray levels
- For a perfect match BC is 1, for a complete mismatch BC is 0
  - higher BC value implies a better match