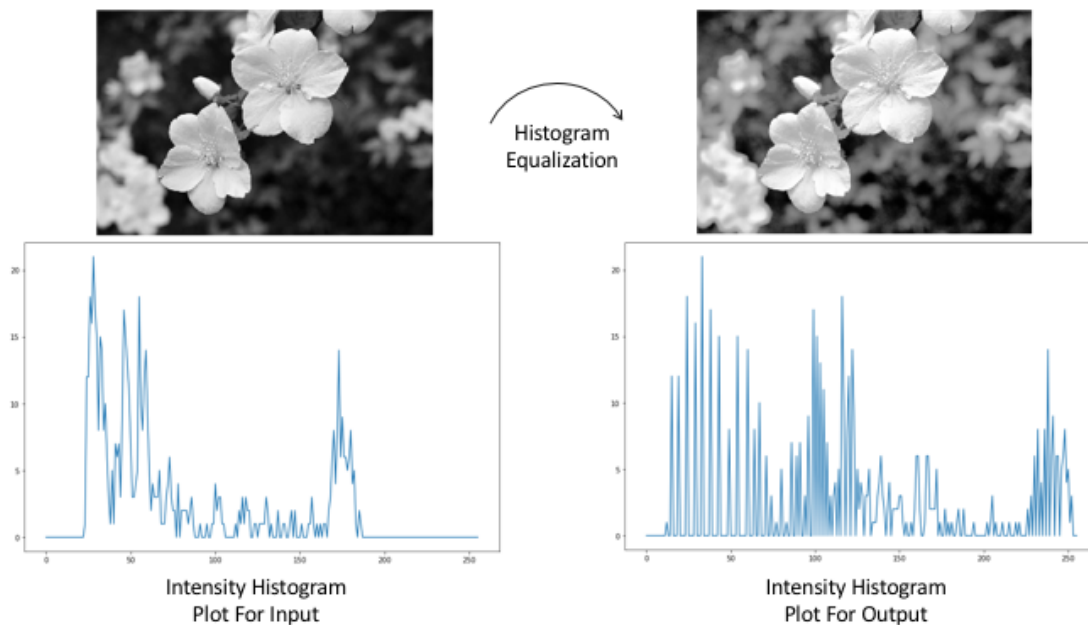


Histogram equalization

The basic point operations, to change the brightness and contrast, help in improving photo quality but require manual tuning. Using histogram equalization technique, these can be found algorithmically and create a better-looking photo. Intuitively, this method tries to set the brightest pixels to white and the darker pixels to black. The remaining pixel values are similarly rescaled. This rescaling is performed by transforming original intensity distribution to capture all intensity distribution. An example of this equalization is as following:



The preceding image is an example of histogram equalization. On the right is the output and, as you can see, the contrast is increased significantly. The input histogram is shown in the bottom figure on the left and it can be observed that not all the colors are observed in the image. After applying equalization, resulting histogram plot is as shown on the right bottom figure. To visualize the results of equalization in the image, the input and results are stacked together in following figure:



Code for the preceding photos is as follows:

```
def plot_gray(input_image, output_image):  
    """  
    Converts an image from BGR to RGB and plots  
    """  
    # change color channels order for matplotlib  
    fig, ax = plt.subplots(nrows=1, ncols=2)  
  
    ax[0].imshow(input_image, cmap='gray')  
    ax[0].set_title('Input Image')  
    ax[0].axis('off')  
  
    ax[1].imshow(output_image, cmap='gray')  
    ax[1].set_title('Histogram Equalized ')  
    ax[1].axis('off')  
  
    plt.savefig('../figures/03_histogram_equalized.png')  
  
    plt.show()  
  
def main():  
    # read an image  
    img = cv2.imread('../figures/flower.png')  
  
    # grayscale image is used for equalization  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
  
    # following function performs equalization on input image  
    equ = cv2.equalizeHist(gray)  
  
    # for visualizing input and output side by side  
    plot_gray(gray, equ)
```

```
if __name__ == '__main__':  
    main()
```