Log transformation is used to enhance pixel intensities that are otherwise missed due to a wide range of intensity values or lost at the expense of high intensity values. If the intensities in the image range from [0,L-1] then the log transformation at (i, j) is given by where and is maximum magnitude value and I(i, j) is the intensity value of the pixel in the input image at (i, j). If both I(i, j) and are equal to L-1 then t(i, j) = L-1. When I(i, j) = 0, since log(1) = 0 will give t(i, j) = 0. While the end points of the range get mapped to themselves, other input values will be transformed by the above equation. The log can be of any base; however, common log (log base 10) or natural log (log base e) are widely used. The inverse of the above log transformation when the base is e is given by , which does the opposite of the log transformation.

Similar to the power law transformation withthe log transformation also maps a small range of dark or low intensity pixel values in the input image to a wide range of intensities in the output image, while a wide range of bright or high intensity pixel values in the input image get mapped to narrow range of high intensities in the output image. Considering the intensity range is between[0,1]. Figure 5.6 illustrates the log and inverse log transformations.

The python code for log transformation is given below.

import scipy.misc

import numpy, math

from scipy.misc.pilutil import Image

#opening the image and converting it to grayscale

a = Image.open('01.png').convert('L')

# a is converted to an ndarray

b = scipy.misc.fromimage(a)

# b is converted to type float

b1 = b.astype(float)

# maximum value in b1 is determined

b2 = numpy.max(b1)

# performing the log transformation

c = (255.0\*numpy.log(1+b1))/numpy.log(1+b2)

# c is converted to type int

c1 = c.astype(int)

# c1 is converted from ndarray to Image

d = scipy.misc.toimage(c1)

#saving d as logtransform\_output.png

d.save('output.png')