

```

1 # Write a Python program to perform the following:
2 # • Apply histogram equalization on the given image for contrast enhancement.
3 # • Detect edges in the given image.
4
5 import numpy as np
6 from PIL import Image, ImageFilter
7 img_filename = 'input_image.jpg'
8 save_filename = 'output_image.jpg'
9 #####
10 # READ IMAGE FROM FILE
11 #####
12 #load file as pillow Image
13 img = Image.open(img_filename)
14 # convert to grayscale
15 imggray = img.convert(mode='L')
16 #convert to NumPy array
17 img_array = np.asarray(imggray)
18 #####
19 # PERFORM HISTOGRAM EQUALIZATION
20 #####
21 #####STEP 1: Normalized cumulative histogram#####
22 #flatten image array and calculate histogram via binning
23 histogram_array = np.bincount(img_array.flatten(), minlength=256)
24 #normalize
25 num_pixels = np.sum(histogram_array)
26 histogram_array = histogram_array/num_pixels
27 #normalized cumulative histogram
28 chistogram_array = np.cumsum(histogram_array)
29 #####STEP 2: Pixel mapping lookup table#####
30 transform_map = np.floor(255 * chistogram_array).astype(np.uint8)
31 #####STEP 3: Transformation#####
32 # flatten image array into 1D list
33 img_list = list(img_array.flatten())
34 # transform pixel values to equalize
35 eq_img_list = [transform_map[p] for p in img_list]
36 # reshape and write back into img_array
37 eq_img_array = np.reshape(np.asarray(eq_img_list), img_array.shape)
38 #####
39 # WRITE EQUALIZED IMAGE TO FILE
40 #####
41 #convert NumPy array to pillow Image and write to file
42 eq_img = Image.fromarray(eq_img_array, mode='L')
43 # Detecting Edges on the Image using the argument ImageFilter.FIND_EDGES
44 image = imggray.filter(ImageFilter.FIND_EDGES)
45 # Saving the Image Under the name
46 image.save(save_filename)

```