DIGITAL IMAGE PROCESSING

Project #1: Apply histogram-specification method

Name: 黎文雄 (Le Van Hung)

Student ID: 0860831

Requirement

Apply histogram-specification method to the image, assume desired histogram:

$$\begin{cases} n_{Zk} = 1248 \quad if \;\; 0 \le k \le 63 \\ n_{Zk} = 800 \quad if \;\; 64 \le k \le 191 \\ n_{Zk} = 1248 \quad if \;\; 192 \le k \le 255 \end{cases}$$

Determine the transformation function z = T(r) and the output image after applying the transformation function.

Implementation

I apply histogram-specification method to the image by using step by step below:

Step1: from original image, calculate n_k of original image

Step2: from calculated n_k, calculate p_r,s_k

$$p_r(r_j) = \frac{n_j}{n} \;\; with \;\; j = 0,\ldots,\mathrm{L}-1 \;\; and \;\; n = \sum_{j=0}^{L-1} n_j$$

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \;\; with \; 0 \le k \le L-1$$

Step3: from desired n_k, calculate p_z,v_n

$$\begin{cases} n_{Zk} = 1248 \quad if \quad 0 \le k \le 63 \\ n_{Zk} = 800 \quad if \quad 64 \le k \le 191 \\ n_{Zk} = 1248 \quad if \quad 192 \le k \le 255 \end{cases}$$

$$p_z(\mathrm{z}_j) = \frac{n_{zj}}{n} \; with \; j = 0,\ldots,\mathrm{L}-1 \; and \; n = \sum_{j=0}^{L-1} n_{zj}$$

$$v_n = \sum_{j=0}^{k} p_z(\mathrm{z}_j) \; with \; 0 \le k \le L-1$$

Step4: from s_k,v_n we can have T(.) to fully satisfy the specification

Algorithm flowchart

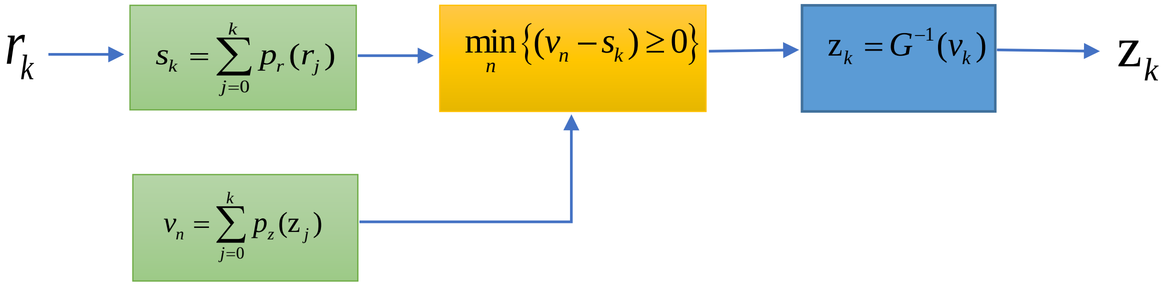

r_k
$s_k = \sum_{j=0}^{k} p_r(r_j)$
$\min_n \{(v_n - s_k) \ge 0\}$
$\mathrm{z}_k = G^{-1}(v_k)$
z_k
$v_n = \sum_{j=0}^{k} p_z(\mathrm{z}_j)$

Fig1: Algorithm of histogram-specification method

The result

Original picture(left) and picture after apply histogram-specification (right)

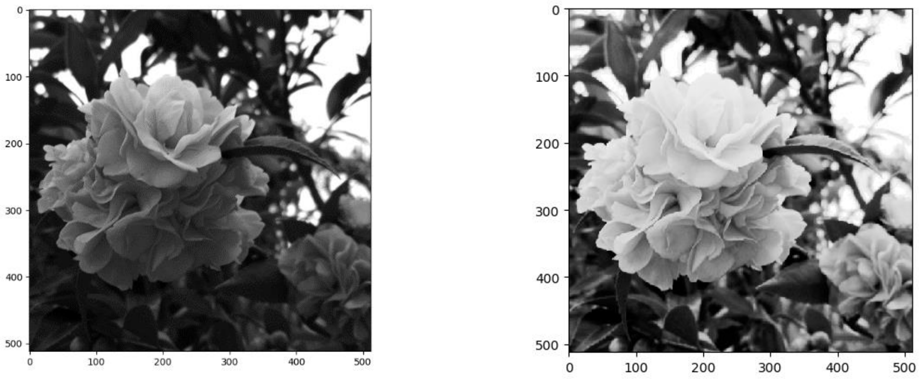


Fig2: comparison of two pictures

Original histogram and histogram of picture after apply histogram-specification

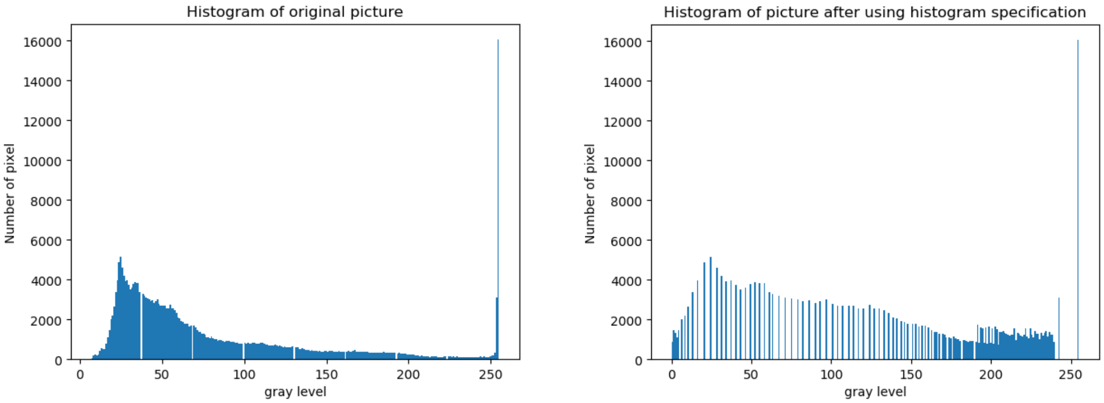


Fig3: comparison of two histograms

Transformation function z = T(r):

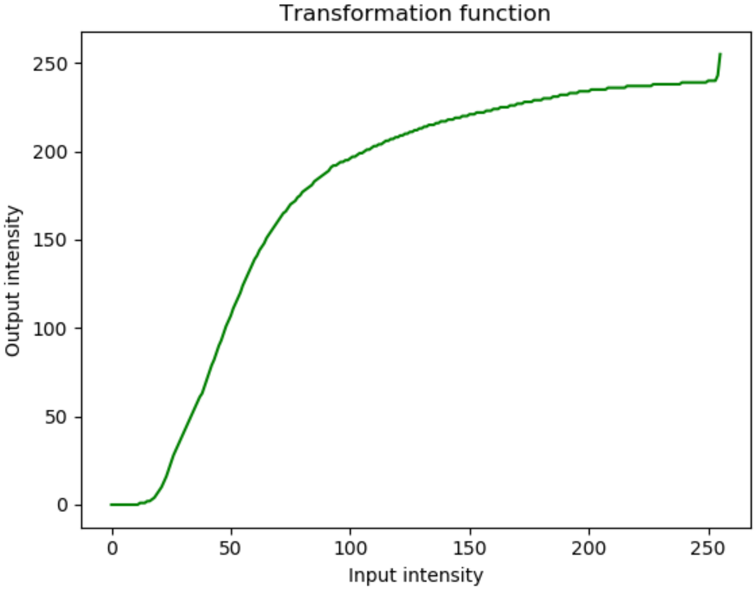


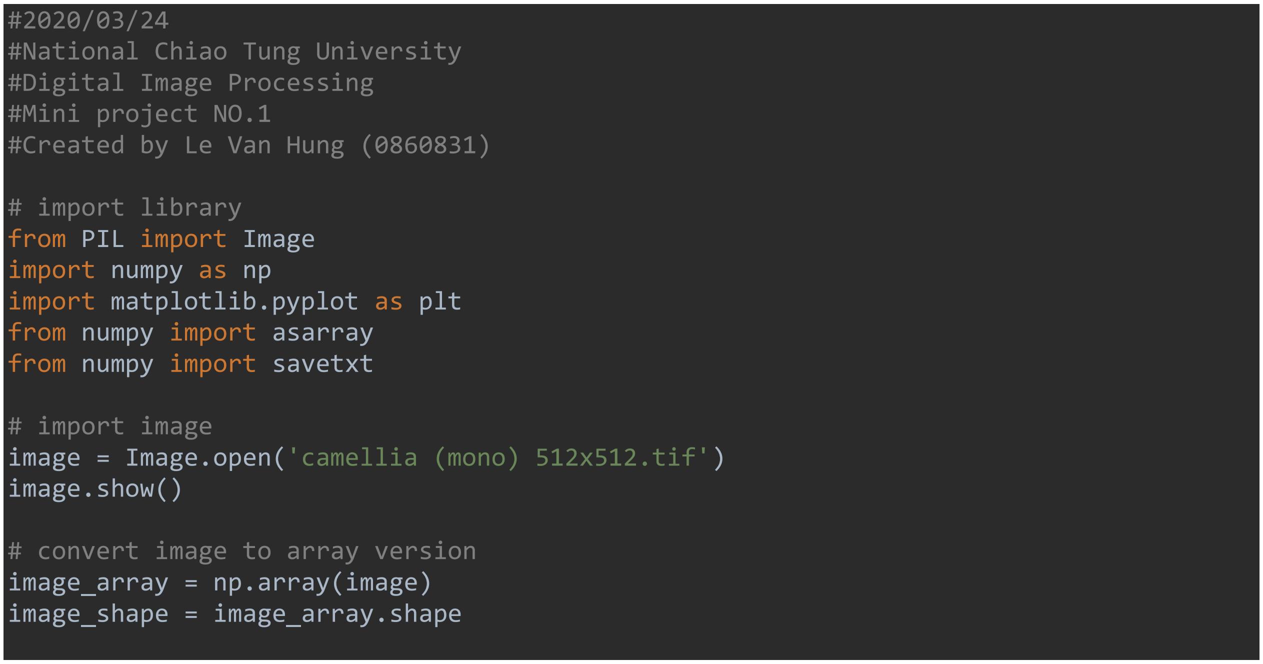
Fig4: Transformation function

Table of transformation function show mapping from input gray level r to output gray level z

r	z	r	z	r	z	r	z	r	z	r	z	r	z	r	z
0	0	33	49	66	153	99	195	132	214	165	225	198	234	231	238
1	0	34	52	67	155	100	196	133	215	166	225	199	234	232	238
2	0	35	55	68	157	101	197	134	215	167	226	200	234	233	238
3	0	36	58	69	159	102	197	135	215	168	226	201	235	234	238
4	0	37	61	70	161	103	198	136	216	169	226	202	235	235	238
5	0	38	63	71	163	104	199	137	216	170	227	203	235	236	238
6	0	39	67	72	165	105	199	138	217	171	227	204	235	237	238
7	0	40	71	73	166	106	200	139	217	172	227	205	235	238	238
8	0	41	75	74	168	107	201	140	217	173	228	206	235	239	239
9	0	42	79	75	170	108	201	141	218	174	228	207	235	240	239
10	0	43	82	76	171	109	202	142	218	175	228	208	236	241	239
11	0	44	86	77	172	110	203	143	218	176	228	209	236	242	239
12	1	45	90	78	174	111	203	144	219	177	229	210	236	243	239
13	1	46	93	79	175	112	204	145	219	178	229	211	236	244	239

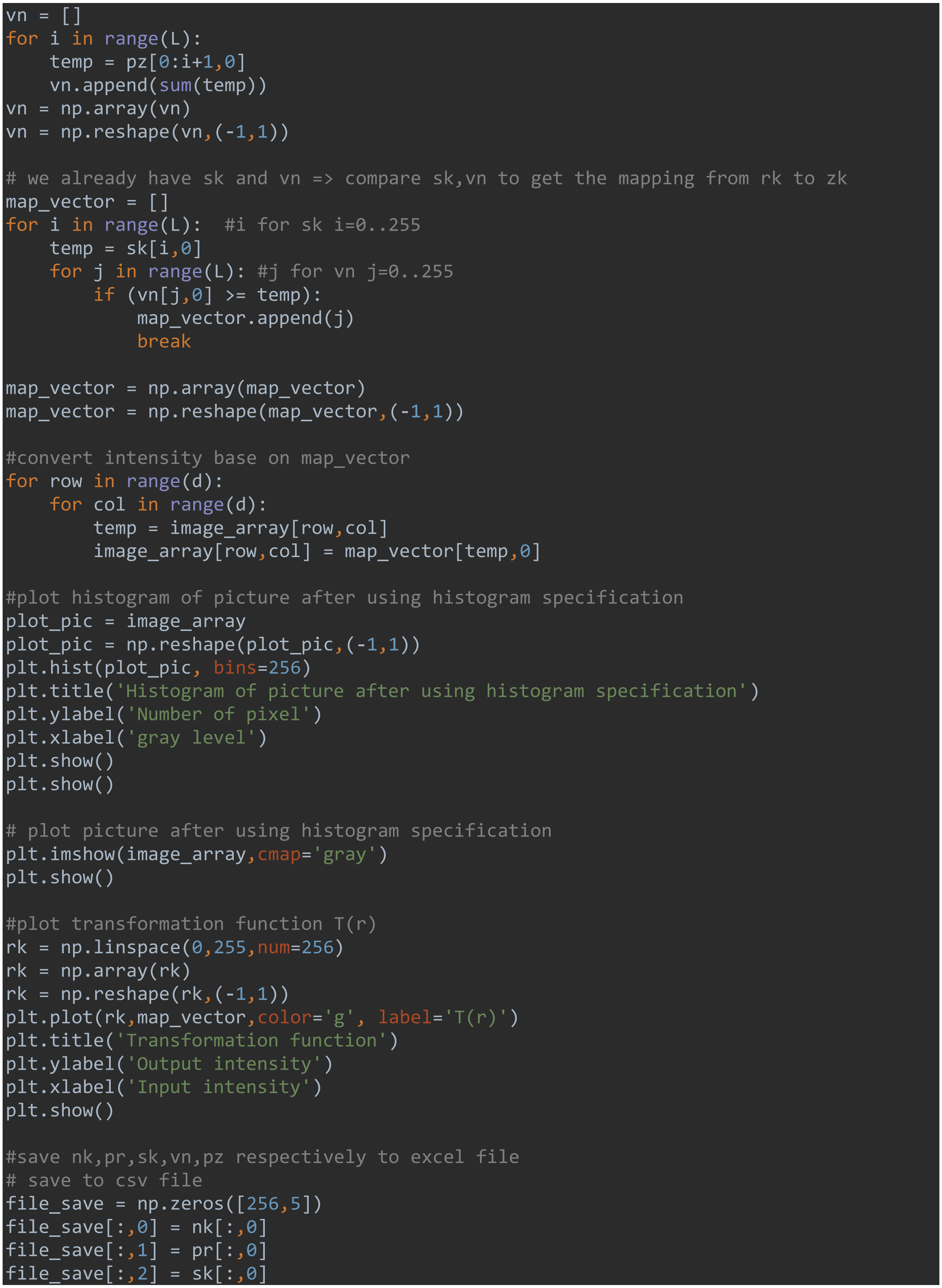
14	1	47	97	80	177	113	204	146	219	179	229	212	236	245	239
15	2	48	101	81	178	114	205	147	220	180	229	213	236	246	239
16	2	49	104	82	179	115	206	148	220	181	230	214	236	247	239
17	3	50	107	83	180	116	206	149	220	182	230	215	236	248	239
18	4	51	111	84	181	117	207	150	221	183	230	216	237	249	239
19	6	52	114	85	183	118	207	151	221	184	230	217	237	250	240
20	8	53	117	86	184	119	208	152	221	185	231	218	237	251	240
21	10	54	120	87	185	120	208	153	222	186	231	219	237	252	240
22	13	55	124	88	186	121	209	154	222	187	231	220	237	253	240
23	16	56	127	89	187	122	209	155	222	188	232	221	237	254	243
24	20	57	130	90	188	123	210	156	222	189	232	222	237	255	255
25	24	58	133	91	189	124	210	157	223	190	232	223	237		
26	28	59	136	92	191	125	211	158	223	191	232	224	237		
27	31	60	139	93	192	126	211	159	223	192	233	225	237		
28	34	61	141	94	192	127	212	160	224	193	233	226	237		
29	37	62	144	95	193	128	212	161	224	194	233	227	238		
30	40	63	146	96	194	129	213	162	224	195	233	228	238		
31	43	64	148	97	194	130	213	163	225	196	234	229	238		
32	46	65	151	98	195	131	214	164	225	197	234	230	238		

Source codes: (created by sing python)

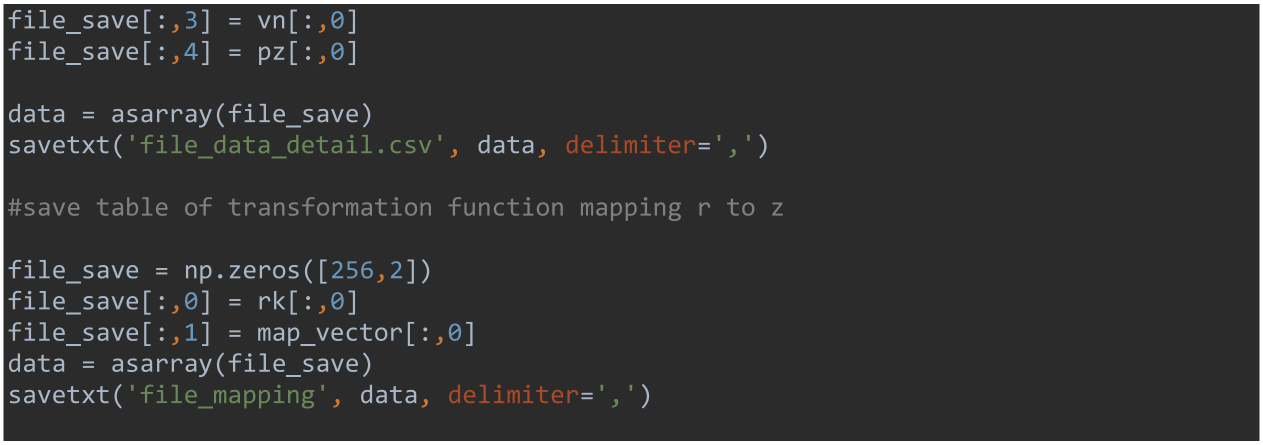


```
#2020/03/24
#National Chiao Tung University
#Digital Image Processing
#Mini project NO.1
#Created by Le Van Hung (0860831)

# import library
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from numpy import asarray
from numpy import savetxt

# import image
image = Image.open('camellia (mono) 512x512.tif')
image.show()

# convert image to array version
image_array = np.array(image)
image_shape = image_array.shape
```

```python
#plot original picture
plt.imshow(image_array,cmap='gray')
plt.show()

#plot histogram of original picture
plot_pic = image_array
plot_pic = np.reshape(plot_pic,(-1,1))
plt.hist(plot_pic, bins=256)
plt.title('Histogram of original picture')
plt.ylabel('Number of pixel')
plt.xlabel('gray level')
plt.show()


# create nk vector including the number of rk respectively (rk is gray level)
nk = []
L = 256
d = image_shape[0]
for k in range(L):
    temp = 0
    for row in range(d):
        for col in range(d):
            if(image_array[row,col] == k):
                temp = temp+1
    nk.append(temp)
nk = np.array(nk)
nk = np.reshape(nk,(-1,1))

# calculate pr vector
pr = nk / (d*d)
pr = np.reshape(pr,(-1,1))

# calculate sk vector
sk = []
for i in range(L):
    temp = pr[0:i+1,0]
    sk.append(sum(temp))
sk = np.array(sk)
sk = np.reshape(sk,(-1,1))

# calculate pz
nz1 = 1248
nz2 = 800
seg_0_63 = nz1/(d*d)
seg_64_191 = nz2/(d*d)
seg_192_255 = nz1/(d*d)

# calculate pz
pz = np.zeros_like(sk)
pz[0:64,0] = seg_0_63
pz[64:192,0] = seg_64_191
pz[192:256,0] = seg_192_255
sum_pz = sum(pz)

# calculate vn
```



```
vn = []
for i in range(L):
    temp = pz[0:i+1,0]
    vn.append(sum(temp))
vn = np.array(vn)
vn = np.reshape(vn,(-1,1))

# we already have sk and vn => compare sk,vn to get the mapping from rk to zk
map_vector = []
for i in range(L):  #i for sk i=0..255
    temp = sk[i,0]
    for j in range(L): #j for vn j=0..255
        if (vn[j,0] >= temp):
            map_vector.append(j)
            break

map_vector = np.array(map_vector)
map_vector = np.reshape(map_vector,(-1,1))

#convert intensity base on map_vector
for row in range(d):
    for col in range(d):
        temp = image_array[row,col]
        image_array[row,col] = map_vector[temp,0]

#plot histogram of picture after using histogram specification
plot_pic = image_array
plot_pic = np.reshape(plot_pic,(-1,1))
plt.hist(plot_pic, bins=256)
plt.title('Histogram of picture after using histogram specification')
plt.ylabel('Number of pixel')
plt.xlabel('gray level')
plt.show()
plt.show()

# plot picture after using histogram specification
plt.imshow(image_array,cmap='gray')
plt.show()

#plot transformation function T(r)
rk = np.linspace(0,255,num=256)
rk = np.array(rk)
rk = np.reshape(rk,(-1,1))
plt.plot(rk,map_vector,color='g', label='T(r)')
plt.title('Transformation function')
plt.ylabel('Output intensity')
plt.xlabel('Input intensity')
plt.show()

#save nk,pr,sk,vn,pz respectively to excel file
# save to csv file
file_save = np.zeros([256,5])
file_save[:,0] = nk[:,0]
file_save[:,1] = pr[:,0]
file_save[:,2] = sk[:,0]
```



```
file_save[:,3] = vn[:,0]
file_save[:,4] = pz[:,0]

data = asarray(file_save)
savetxt('file_data_detail.csv', data, delimiter=',')

#save table of transformation function mapping r to z

file_save = np.zeros([256,2])
file_save[:,0] = rk[:,0]
file_save[:,1] = map_vector[:,0]
data = asarray(file_save)
savetxt('file_mapping', data, delimiter=',')
```

NOTE*

My report consists of 4 files:

1. 0860831_DIP_Project_1 : file report
2. histogram_specification.py : file python code
3. fixxed_image.png : the image after using histogram-specification method
4. file_data_detail.csv : file about the detail that consists of vectors n_k, p_r, s_k, v_n, p_z