Image Processing Project #1

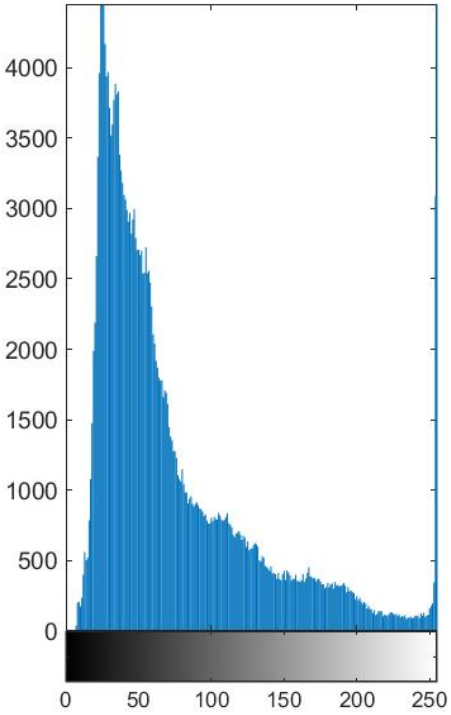
0510894 電機 4D 翁紹恩

I. Source codes (With Matlab)

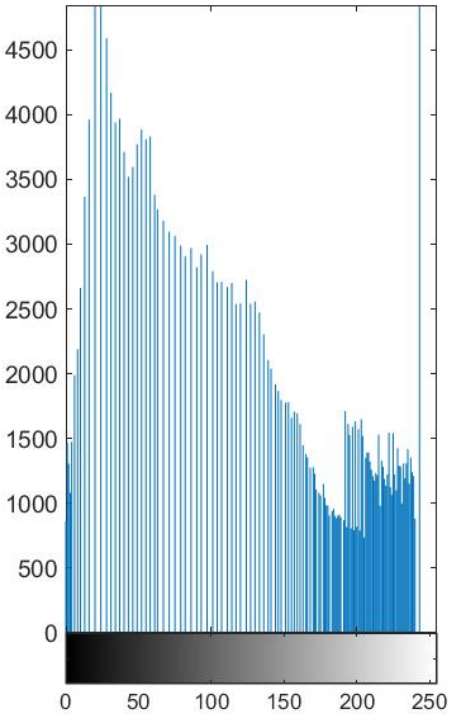
```
%% load image %%
I = imread('camellia (mono) 512x512.tif');
%% plot image and its histogram %%
figure
subplot(1,2,1)
imshow(I)
subplot(1,2,2)
imhist(I)
saveas(gcf,'origin','png');
%% calculate the number of every gray level %%
[nk,rk]=imhist(I);
%% turn the number into probability and calculate its cdf %%
pr=nk/(512*512);
sk=zeros(256,1);
for i =1:256
    if i==1
        sk(i)=pr(i);
    else
        sk(i)=sk(i-1)+pr(i);
    end
end
%% define the expected spec. %%
pz=zeros(256,1);
pz(1:64)=1248/(512*512);
pz(193:256)=1248/(512*512);
pz(65:192)=800/(512*512);
vn=zeros(256,1);
for i =1:256
    if i==1
        vn(i)=pz(i);
    else
        vn(i)=vn(i-1)+pz(i);
    end
end
%% histogram equalization %%
```

```
startfrom=ones(256,1);
% create table of transformation function %
for i =1:256
    if i==1
        for j=1:256
            if vn(j)>=sk(i)
                startfrom(i)=j;
                break;
            end
        end
    else
        for j=startfrom(i-1):256
            if vn(j)>=sk(i)
                startfrom(i)=j;
                break;
            end
        end
    end
end
% Since matlab index starts from 1, and the gray level starts from 0, so minus 1
realgraylevel=startfrom-1;
%% transfrom the image %%
transfer_image=zeros(512,512);
for i=1:512
    for j=1:512
        transfer_image(i,j)=realgraylevel(I(i,j)+1);
    end
end
%%plot the resulted image and its histogram %%
figure
subplot(1,2,1)
imshow(transfer_image,[0 255]);
subplot(1,2,2)
imhist(uint8(transfer_image));
saveas(gcf,'transferred','png');
```

II. Figures of the original image and histograms



III. Figure of the output image and histograms after applying the histogram specification scheme



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

r_k	z_k	r_k	z_k	r_k	z_k
0	0	34	52	68	157
1	0	35	55	69	159
2	0	36	58	70	161
3	0	37	61	71	163
4	0	38	63	72	165
5	0	39	67	73	166
6	0	40	71	74	168
7	0	41	75	75	170
8	0	42	79	76	171
9	0	43	82	77	172
10	0	44	86	78	174
11	0	45	90	79	175
12	1	46	93	80	177
13	1	47	97	81	178
14	1	48	101	82	179
15	2	49	104	83	180
16	2	50	107	84	181
17	3	51	111	85	183
18	4	52	114	86	184
19	6	53	117	87	185
20	8	54	120	88	186
21	10	55	124	89	187
22	13	56	127	90	188
23	16	57	130	91	189
24	20	58	133	92	191
25	24	59	136	93	192
26	28	60	139	94	192
27	31	61	141	95	193
28	34	62	144	96	194
29	37	63	146	97	194
30	40	64	148	98	195
31	43	65	151	99	195
32	46	66	153	100	196
33	49	67	155	101	197

r_k	z_k
102	197
103	198
104	199
105	199
106	200
107	201
108	201
109	202
110	203
111	203
112	204
113	204
114	205
115	206
116	206
117	207
118	207
119	208
120	208
121	209
122	209
123	210
124	210
125	211
126	211
127	212
128	212
129	213
130	213
131	214
132	214
133	215
134	215
135	215
136	216
137	216

r_k	z_k
138	217
139	217
140	217
141	218
142	218
143	218
144	219
145	219
146	219
147	220
148	220
149	220
150	221
151	221
152	221
153	222
154	222
155	222
156	222
157	223
158	223
159	223
160	224
161	224
162	224
163	225
164	225
165	225
166	225
167	226
168	226
169	226
170	227
171	227
172	227
173	228

r_k	z_k
174	228
175	228
176	228
177	229
178	229
179	229
180	229
181	230
182	230
183	230
184	230
185	231
186	231
187	231
188	232
189	232
190	232
191	232
192	233
193	233
194	233
195	233
196	234
197	234
198	234
199	234
200	234
201	235
202	235
203	235
204	235
205	235
206	235
207	235
208	236
209	236

r_k	z_k
210	236
211	236
212	236
213	236
214	236
215	236
216	237
217	237
218	237
219	237
220	237
221	237
222	237
223	237
224	237
225	237
226	237
227	238
228	238
229	238
230	238
231	238
232	238
233	238
234	238
235	238
236	238
237	238
238	238
239	239
240	239
241	239
242	239
243	239
244	239
245	239

r_k	z_k
246	239
247	239
248	239
249	239
250	240
251	240
252	240
253	240
254	243
255	255