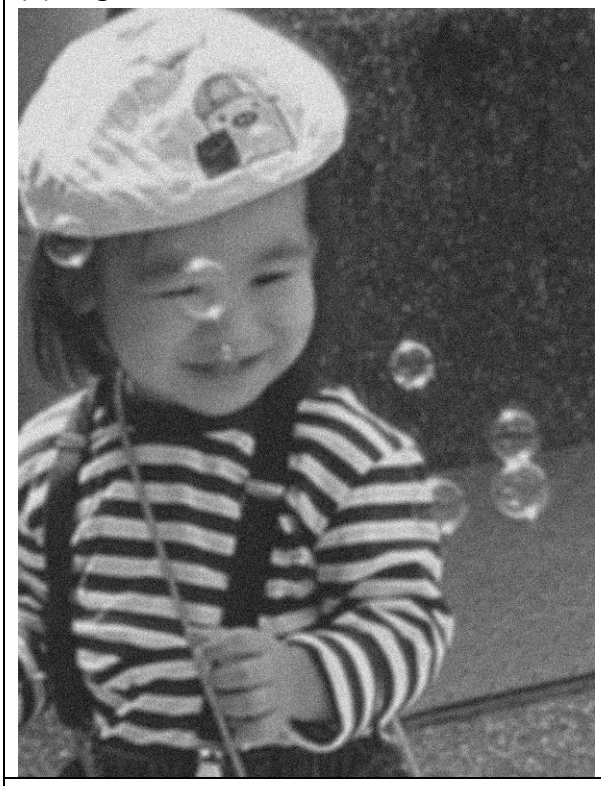
Project1

studentID: 311512040 name:林胤宏

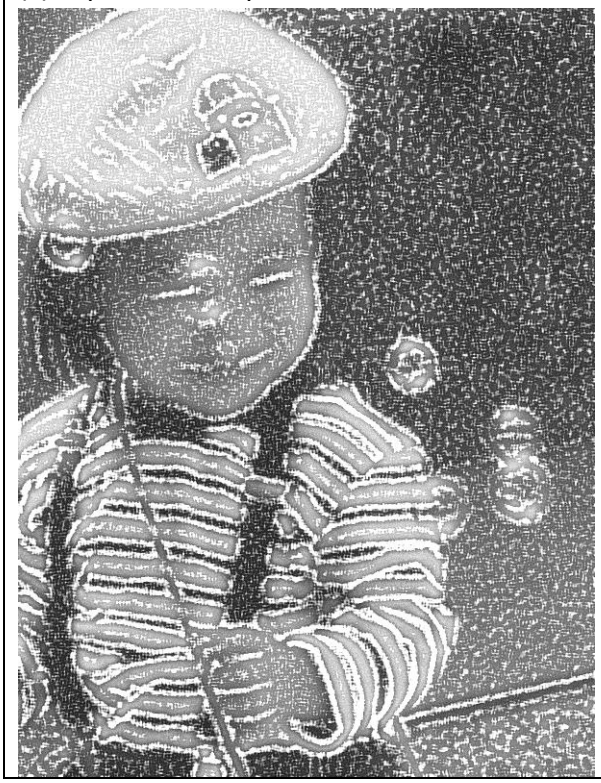
1. Kid blurred-noisy.tiff

(a) original



(b) Laplacian

(c) Laplacian-sharpened

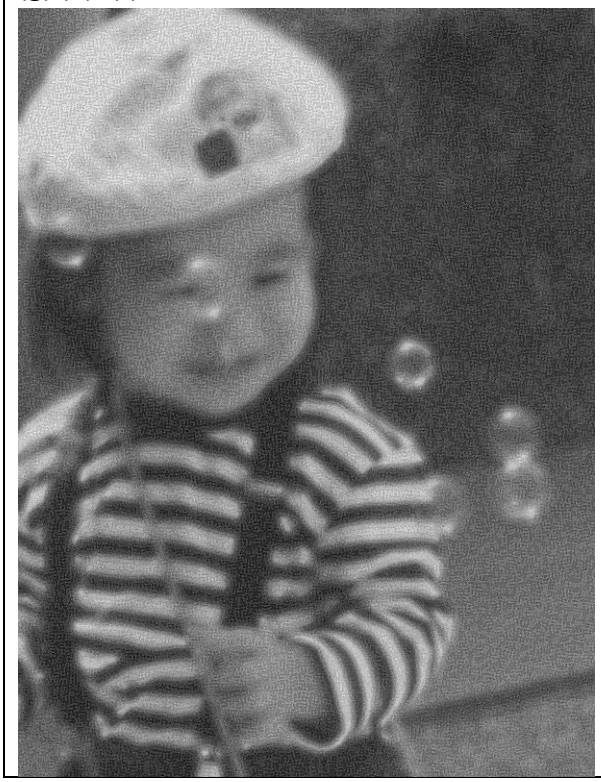


(d) Sobel-gradient

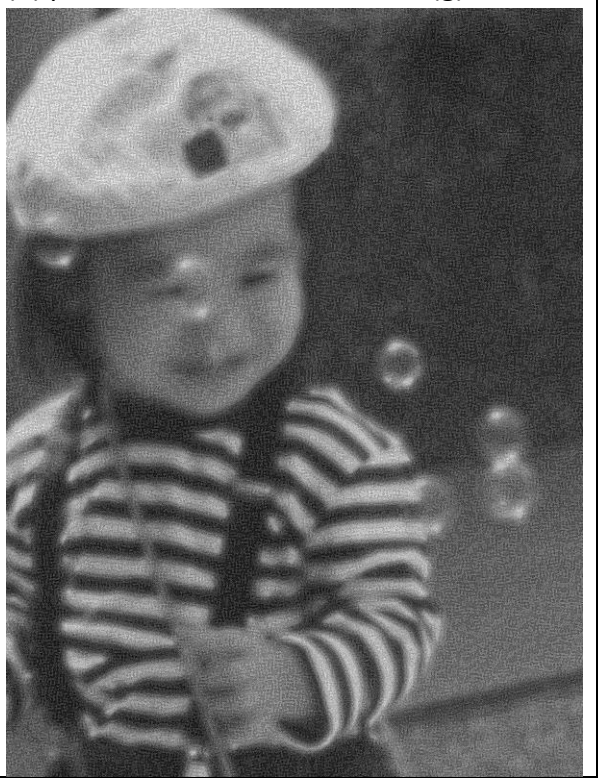
(e) smoothed gradient

(f) (e) • (b)

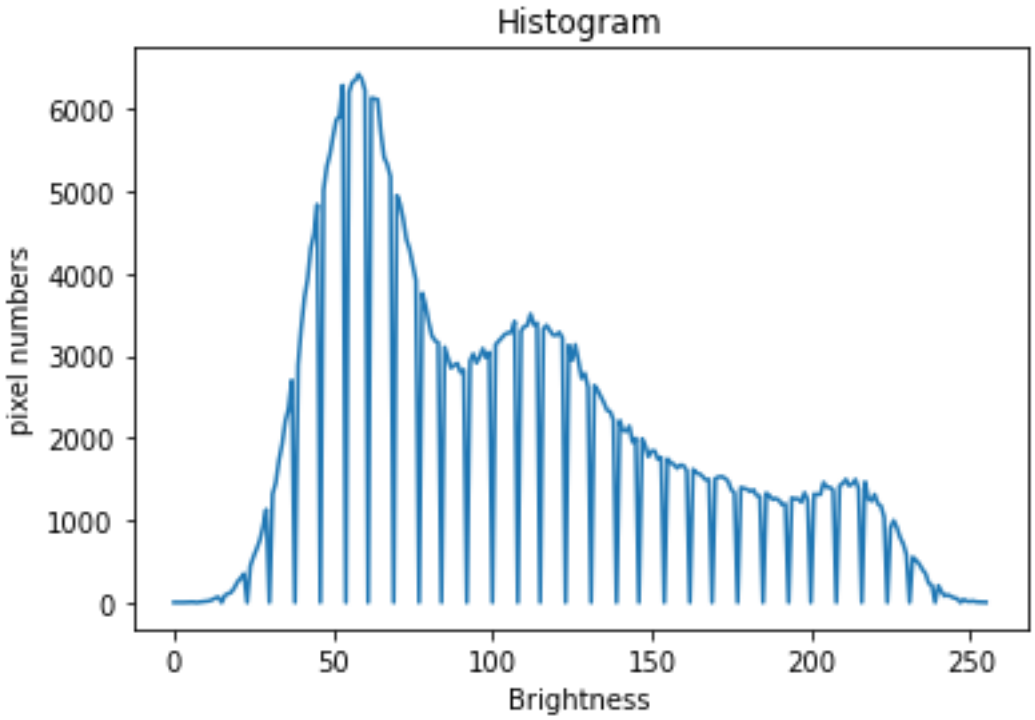
(g) (a)+(f)



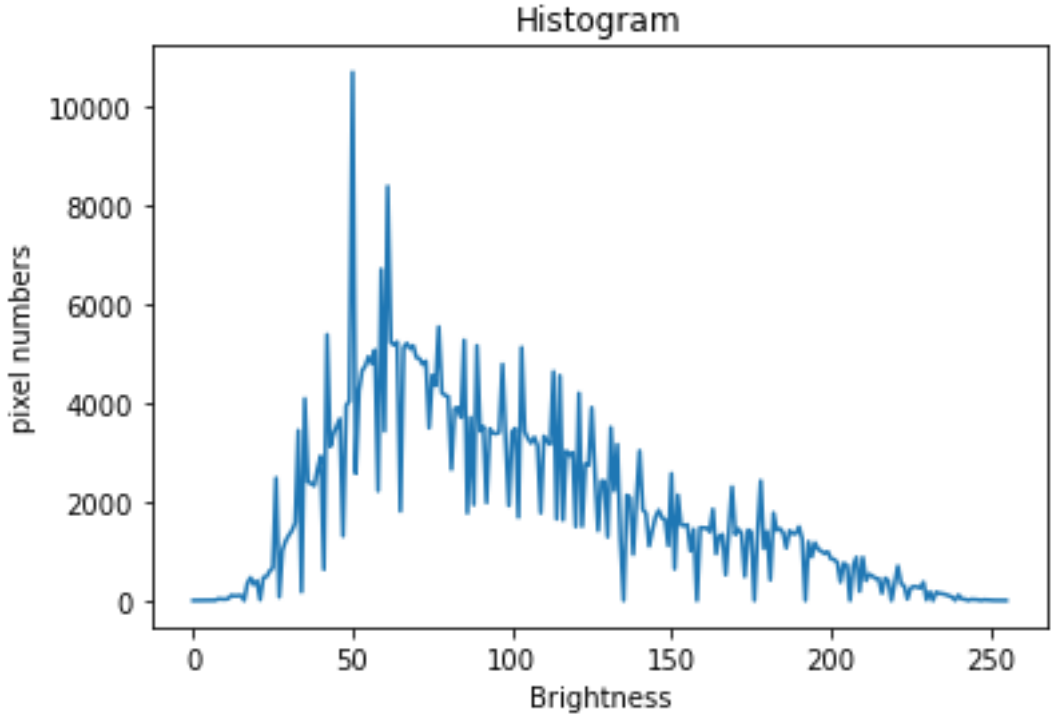
(h) power-law transformation of (g)



Original histograms



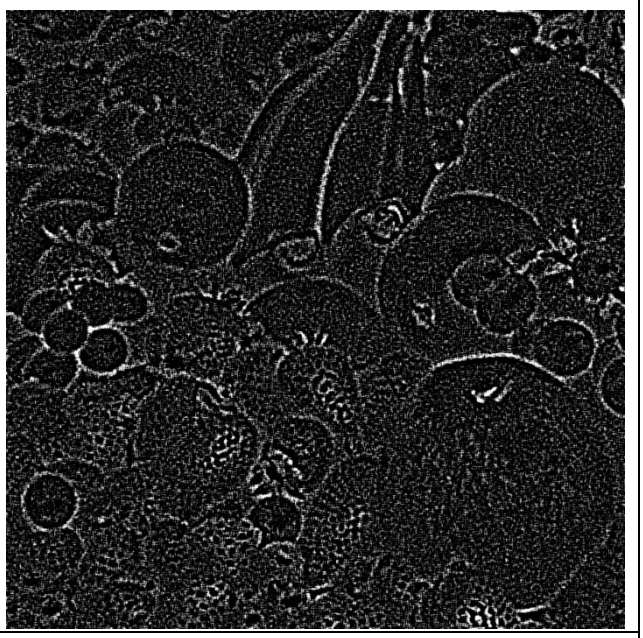
Output histograms



2. Fruit blurred-noisy.tiff

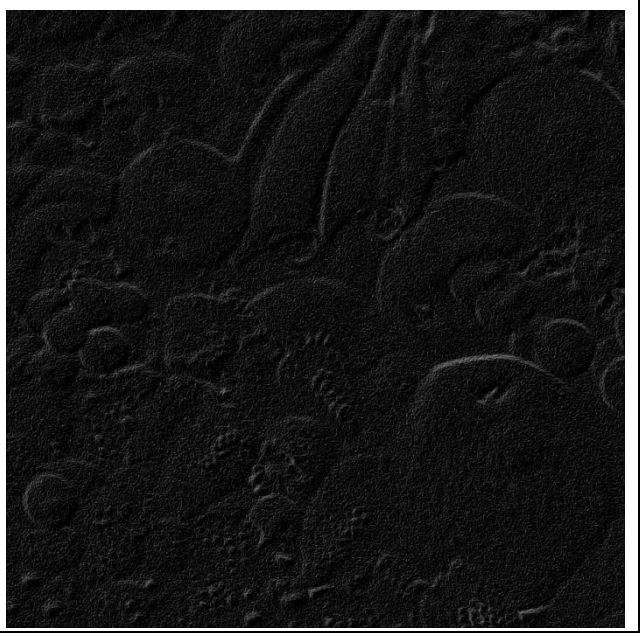
(a) original

(b) Laplacian

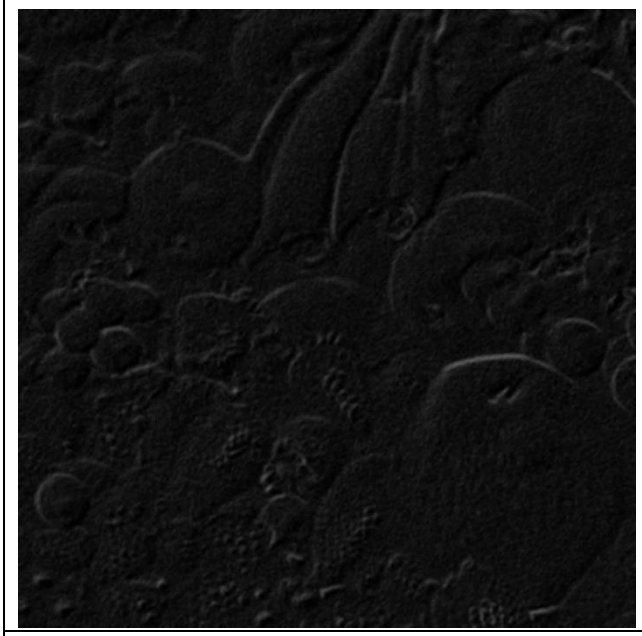


(c) Laplacian-sharpened

(d) Sobel-gradient

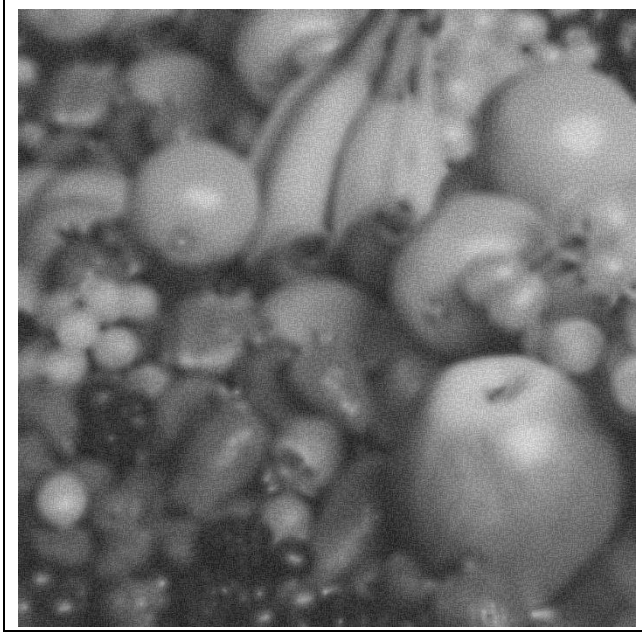


(e) smoothed gradient



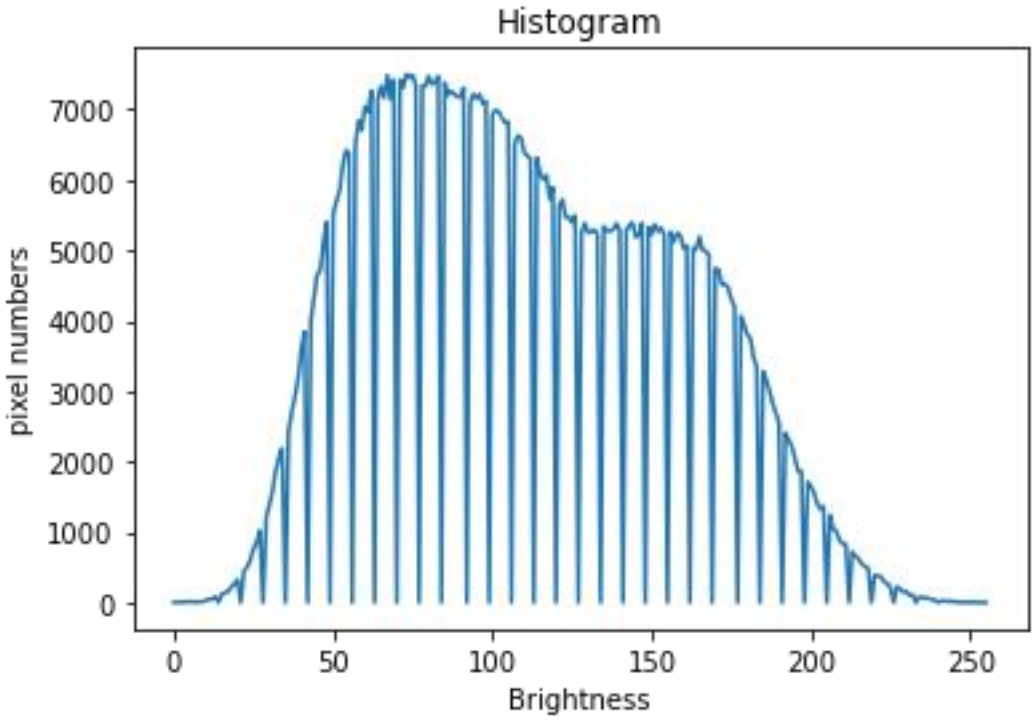
(f) (e) • (b)

(g) (a)+(f)

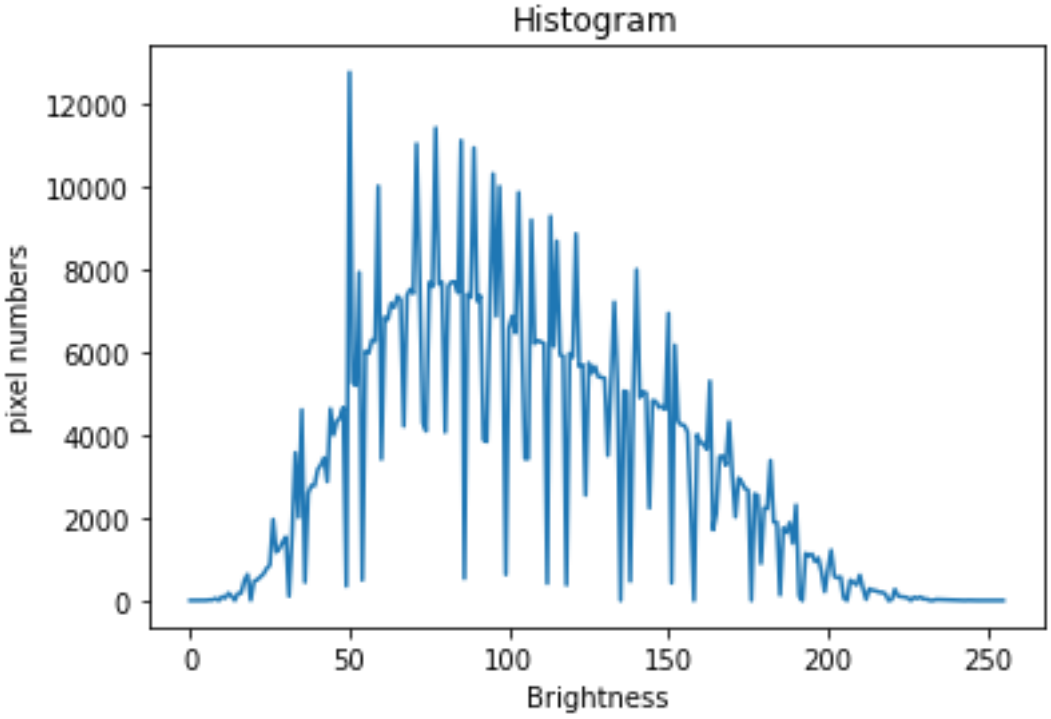


(h) power-law transformation of (g)

Original histograms



Output histograms



cc

r	*Kid Ori*	*Kid Out*	*Fruit Ori*	*Fruit Out*
0	1	0	2	1
1	0	0	3	1
2	0	3	1	8
3	1	2	2	2
4	1	4	6	6
5	4	4	8	3
6	6	3	12	14
7	0	5	0	13
8	5	35	12	51
9	13	27	11	0
10	14	26	30	93
11	19	36	45	60
12	32	106	46	177
13	54	90	81	108
14	65	96	0	2
15	0	103	112	174
16	85	2	138	180
17	107	345	156	481
18	124	462	218	629
19	179	323	265	7
20	252	398	328	464
21	295	15	0	498
22	349	467	452	571
23	0	481	500	641
24	444	596	610	775
25	557	662	758	854
26	652	2485	850	1964
27	759	70	1021	1175
28	922	1016	0	1255
29	1130	1206	1208	1412
30	0	1322	1387	1536
31	1320	1418	1594	104
32	1458	1574	1879	1649
33	1731	3450	2061	3578
34	1926	175	2186	2010
35	2205	4093	0	4617
36	2354	2405	2437	442
37	2700	2375	2703	2596
38	0	2327	2923	2781

39	2886	2658	3180	2778
40	3344	2928	3600	3166
41	3706	628	3843	3283
42	3933	5391	0	3453
43	4310	3120	4002	2877
44	4478	3377	4326	4640
45	4838	3530	4664	4009
46	0	3683	4711	4342
47	5013	1306	5134	4419
48	5295	3952	5399	4670
49	5447	4045	0	348
50	5670	10712	5510	12786
51	5867	2577	5692	5275
52	5910	4247	5840	5204
53	6286	4665	6200	7941
54	0	4756	6416	492
55	6205	4945	6391	6038
56	6322	4807	0	5986
57	6351	5079	6501	6306
58	6422	2222	6850	6273
59	6359	6717	6705	10030
60	6225	3435	7037	3419
61	0	8398	6965	6856
62	6133	5237	7267	6799
63	6123	5178	0	7189
64	6112	5242	7151	7089
65	5709	1812	7330	7368
66	5411	5135	7170	7280
67	5325	5214	7484	4219
68	5176	5104	7147	7377
69	0	5168	7424	7533
70	4948	4926	0	7431
71	4844	4903	7423	11053
72	4663	4785	7313	7644
73	4414	4851	7503	4304
74	4296	3498	7450	4102
75	4102	4561	7496	7697
76	3914	4356	7336	7593
77	0	5548	0	11439
78	3761	4208	7337	7655
79	3608	4160	7338	7704

80	3411	4135	7474	4074
81	3242	2659	7359	7609
82	3182	3891	7391	7711
83	3159	3913	7473	7707
84	0	3712	0	7449
85	3104	5279	7386	11135
86	2980	1773	7205	533
87	2851	3714	7261	7413
88	2885	1940	7224	7337
89	2906	5167	7184	10955
90	2798	3437	7176	7246
91	2828	3525	7306	7374
92	0	1983	0	3886
93	2935	3481	7084	3845
94	3022	3394	7219	7014
95	2912	3381	7160	10328
96	2982	3410	7218	6880
97	3086	4790	7079	10021
98	2976	3257	7121	6835
99	3041	1923	0	625
100	0	3416	6930	6590
101	3134	3487	6992	6882
102	3178	1688	6971	6472
103	3224	5133	6885	9878
104	3262	3419	6812	6446
105	3278	3285	6823	3431
106	3285	3189	0	3432
107	3415	3310	6511	9210
108	0	3142	6627	6220
109	3299	1773	6599	6292
110	3355	3329	6387	6257
111	3371	3221	6347	6222
112	3506	3168	6298	414
113	3375	4642	0	9302
114	3398	1651	6311	6151
115	0	4559	6097	8699
116	3323	1631	6004	5943
117	3369	3034	6065	5915
118	3310	2937	5732	376
119	3252	2998	5895	5978
120	3248	1497	0	5883

121	3284	4204	5639	8875
122	3214	1504	5725	5658
123	0	2768	5473	5710
124	3126	2740	5468	2547
125	2945	3911	5396	5749
126	3134	2572	5495	5503
127	2940	1417	0	5663
128	2726	2421	5258	5413
129	2786	2418	5387	5401
130	2617	1288	5252	5377
131	0	3510	5260	3507
132	2636	2223	5294	5227
133	2583	3164	5232	7223
134	2509	1234	0	5279
135	2430	0	5339	0
136	2336	2133	5281	5068
137	2321	2082	5274	5050
138	2219	935	5300	466
139	0	2052	5384	5049
140	2211	3037	5282	8009
141	2092	1845	0	4911
142	2081	1770	5275	5048
143	2143	1104	5358	5024
144	1951	1424	5394	2237
145	1996	1693	5189	4837
146	0	1829	5216	4820
147	1989	1674	5389	4671
148	1905	1643	0	4700
149	1773	1111	5332	4626
150	1843	2574	5227	6950
151	1840	639	5371	422
152	1739	2138	5270	6180
153	1764	1527	5319	4341
154	0	1522	5212	4249
155	1745	1536	0	4239
156	1698	1006	5260	4072
157	1679	1444	5124	2224
158	1632	0	5245	0
159	1671	1478	5217	4017
160	1664	1470	5033	3817
161	1604	1471	5068	3800

162	0	1395	0	3661
163	1617	1851	5032	5312
164	1576	947	5014	1712
165	1565	1303	5196	2101
166	1531	1338	5005	3479
167	1490	520	4991	3494
168	1496	1408	4931	3271
169	0	2304	0	4326
170	1506	1337	4743	3062
171	1525	1451	4729	2022
172	1530	1351	4514	2961
173	1511	486	4524	2857
174	1475	1430	4475	2709
175	1372	1390	4307	2668
176	1333	0	4205	0
177	0	1365	0	2577
178	1397	2429	4065	2541
179	1388	1055	3947	895
180	1376	1399	3808	2242
181	1347	416	3757	2225
182	1369	1775	3479	3389
183	1298	1419	3352	1898
184	1291	1458	0	1888
185	0	1375	3280	132
186	1329	1064	3161	1755
187	1297	1416	3004	1646
188	1252	1357	2826	1882
189	1264	1353	2694	1385
190	1243	1493	2551	2310
191	1188	1244	0	113
192	1199	0	2407	0
193	0	1193	2307	1140
194	1271	885	2221	1085
195	1258	1170	2046	1123
196	1261	1039	1862	935
197	1218	1000	1865	1029
198	1334	951	0	695
199	1247	990	1712	213
200	0	843	1642	750
201	1319	832	1562	1219
202	1310	774	1399	582

203	1315	374	1332	547
204	1459	756	1364	570
205	1401	718	0	55
206	1399	0	1238	0
207	1366	698	1048	491
208	0	874	1013	438
209	1418	186	885	376
210	1445	874	831	613
211	1497	403	831	293
212	1420	530	0	28
213	1441	499	718	282
214	1494	442	658	244
215	1382	445	621	236
216	0	147	541	202
217	1465	455	518	197
218	1268	399	481	162
219	1234	0	0	0
220	1313	366	383	18
221	1201	698	383	269
222	1173	337	360	105
223	1038	297	296	95
224	0	30	266	81
225	908	250	214	66
226	996	285	0	1
227	912	274	222	79
228	791	252	193	49
229	713	370	149	86
230	583	12	156	34
231	0	177	101	38
232	551	0	108	0
233	515	168	0	3
234	469	143	76	25
235	403	136	73	22
236	351	119	57	13
237	232	92	61	15
238	209	70	42	9
239	0	4	33	10
240	205	110	0	11
241	131	30	23	8
242	80	27	22	4
243	92	0	14	0

244	75	22	21	5
245	55	21	7	2
246	45	15	8	1
247	0	0	0	0
248	36	14	5	1
249	14	9	2	2
250	13	8	2	0
251	17	2	1	1
252	7	1	4	1
253	5	0	1	0
254	0	0	0	0
255	2	0	1	0

3.code

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from openpyxl import load_workbook
```

```
#(a)original
img = cv2.imread("./fruit blurred-noisy.tif", 0)
# normalize float versions
norm_img1 = cv2.normalize(img, None, alpha=0,beta=1, norm_type=cv2.NORM_MINMAX,
dtype=cv2.CV_64F)
# scale to uint8
norm_img1 = (255*norm_img1).astype(np.uint8)
plt.imshow(norm_img1,cmap='gray')
plt.show()

norm_img = Image.fromarray(norm_img1)
norm_img.save('output/(a)original.tif',dpi=(200.0,200.0))
```

```
#(b)Laplacian
#blur = cv2.medianBlur(norm_img1, 3)
blur = cv2.boxFilter(norm_img1,-1,(5,5),normalize=True)
gray_lap = cv2.Laplacian(blur,-1, ksize=5)
Laplacian = cv2.convertScaleAbs(gray_lap)
plt.imshow(Laplacian,cmap='gray')
plt.show()
laplacian = Image.fromarray(Laplacian)
laplacian.save('output/(b)Laplacian.tif',dpi=(200.0,200.0))
```

```
#(c)Laplacian-sharpened
Laplacian_sharpened = cv2.add(Laplacian, norm_img1)
plt.imshow(Laplacian_sharpened,cmap='gray')
plt.show()

laplacian_sharpened = Image.fromarray(Laplacian_sharpened)
laplacian_sharpened.save('output/(c)Laplacian-sharpened.tif',dpi=(200.0,200.0))
```

```
#(d)Sobel-gradient
x = cv2.Sobel(norm_img1, -1, 1, 0, ksize=3)
y = cv2.Sobel(norm_img1, -1, 0, 1, ksize=3)
absX = cv2.convertScaleAbs(x)    # 轉換為影像原本儲存的格式 uint8
absY = cv2.convertScaleAbs(y)

Sobel_gradient = cv2.addWeighted(absX, 0.5, absY,0.5,0)

plt.subplot(121)
plt.imshow(absX,cmap='gray')
plt.subplot(122)
plt.imshow(absY,cmap='gray')
plt.show()
plt.imshow(Sobel_gradient,cmap='gray')
plt.show()

sobel_gradient = Image.fromarray(Sobel_gradient)
sobel_gradient.save('output/(d)Sobel-gradient.tif',dpi=(200.0,200.0))
```

```
# (e)smoothed gradient

smooth_gradient = cv2.boxFilter(Sobel_gradient, -1, (5,5), normalize=True)
plt.imshow(smooth_gradient,cmap='gray')
plt.show()

Smooth_gradient = Image.fromarray(smooth_gradient)
Smooth_gradient.save('output/(e)smoothed gradient.tif',dpi=(200.0,200.0))
```

```
# (f)extracted feature
ext = cv2.multiply(smooth_gradient ,Laplacian)
#ext = smooth_gradient * Laplacian
plt.imshow(ext,cmap='gray')
plt.show()

Ext = Image.fromarray(ext)
Ext.save('output/(f)extracted feature.tif',dpi=(200.0,200.0))
```

```
# (g)
c = 0.8
g_img  = cv2.addWeighted(norm_img1, c, ext, 1-c, 0)
plt.imshow(g_img,cmap='gray')
plt.show()

G_img = Image.fromarray(g_img)
G_img.save('output/(g)=(a)+(f).tif',dpi=(200.0,200.0))
```

```
#(h)final
gamma = 1.1
h_img = np.array(255*(g_img/255)**gamma,dtype='uint8')
plt.imshow(h_img,cmap='gray')
plt.show()

H_img = Image.fromarray(h_img)
H_img.save('output/(h)final.tif',dpi=(200.0,200.0))
```

```
#histogram
hist = cv2.calcHist([norm_img1], [0], None, [256], [0, 256]) #cv2.calcHist(影像, 通道, 遮罩, 區間數量, 數值範圍)
hist2 = cv2.calcHist([h_img], [0], None, [256], [0, 256])
print(hist.shape)

plt.title("Histogram")
plt.xlabel("Brightness")
plt.ylabel("pixel numbers")
plt.plot(hist)
plt.show()

plt.title("Histogram")
plt.xlabel("Brightness")
plt.ylabel("pixel numbers")
```

```
plt.plot(hist2)
plt.show()
```

```
#將資料寫入 excel
wb = load_workbook("Histograms.xlsx")
print(wb.sheetnames)
sheet = wb.worksheets[0]  #抓出列數
print("row: {}, column: {}".format(sheet.max_row, sheet.max_column))

for i in range(2, sheet.max_row+1):
    sheet.cell(row = i, column = 4, value = hist[i-2][0])
    sheet.cell(row = i, column = 5, value = hist2[i-2][0])

wb.save("Histograms.xlsx")
```