

```

1
2 import cv2
3 import numpy as np
4 #import random
5 import sys
6 import argparse
7 #import json
8 #from json_tricks import dump, dumps, load, loads, strip_comments
9 import glob, os
10 import scipy.misc as misc
11 from sklearn import decomposition
12 import matplotlib.pyplot as plt
13 from mpl_toolkits.mplot3d import Axes3D
14 import time
15 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
16 from metric_learn import NCA
17 from sklearn.manifold import TSNE
18
19 from dim_module import *
20
21 def ParseArguments():
22     parser = argparse.ArgumentParser(description="Project ")
23     parser.add_argument('--image', default="", required=True, help='Color image (default: %(default)s)')
24     args = parser.parse_args()
25
26     return args.image
27
28
29 # main program
30 image_file = ParseArguments()
31
32 image = misc.imread(image_file ,mode="RGB")
33
34
35 image_misc_bw = misc.imread(image_file ,mode="L")
36
37
38 image_height=image.shape[0]
39 image_width=image.shape[1]
40
41 image2=image.reshape(-1,3)
42
43
44
45 print("Calculating PCA...", end="", flush=True)
46 start_time = time.time()
47 pca = decomposition.PCA(n_components=3)
48 pca.fit(image2)
49 points_pca_reduced = pca.transform(image2)
50 print("\t\t took %s seconds " % round((time.time() - start_time),5))
51
52 print("points_pca_reduced.shape = ", points_pca_reduced.shape)
53
54 #quit()
55
56
57 image_bw_pc1 = points_pca_reduced[:,0].reshape(image_height,image_width)
58 image_bw_pc2 = points_pca_reduced[:,1].reshape(image_height,image_width)
59 image_bw_pc3 = points_pca_reduced[:,2].reshape(image_height,image_width)
60
61 # ~ image_rgb_pca=np.zeros(image.shape)
62
63 # ~ image_rgb_pca[:, :,0] = 255*(image_bw_pc1 - np.min(image_bw_pc1))/np.ptp(image_bw_pc1).astype(int)
64 # ~ image_rgb_pca[:, :,1] = 255*(image_bw_pc2 - np.min(image_bw_pc2))/np.ptp(image_bw_pc2).astype(int)
65 # ~ image_rgb_pca[:, :,2] = 255*(image_bw_pc3 - np.min(image_bw_pc3))/np.ptp(image_bw_pc3).astype(int)
66
67 # ~ #image_rgb_pca[:, :,1] = image_bw_pc2
68 # ~ #image_rgb_pca[:, :,2] = image_bw_pc1
69
70
71
72 axes1, center1, eigenvals1, explained_var1 = perform_PCA(image2);
73
74
75
76 fig_orig = plt.figure(1)
77 ax_orig = fig_orig.add_subplot(111)
78 ax_orig.imshow(image )
79
80

```

```

81 fig_misc_bw = plt.figure(2)
82 ax_misc_bw = fig_misc_bw.add_subplot(111)
83 ax_misc_bw.imshow(image_misc_bw, cmap='gray')
84
85
86
87 fig_pca = plt.figure(3)
88 ax_pca = fig_pca.add_subplot(111)
89 ax_pca.imshow(image_bw_pc1, cmap='gray')
90
91
92 fig_pca2 = plt.figure(4)
93 ax_pca2 = fig_pca2.add_subplot(111)
94 ax_pca2.imshow(image_bw_pc2, cmap='gray')
95
96
97 fig_pca3 = plt.figure(5)
98 ax_pca3 = fig_pca3.add_subplot(111)
99 ax_pca3.imshow(image_bw_pc3, cmap='gray')
100
101
102
103
104 # ~ fig_rgb_pca = plt.figure(6)
105 # ~ ax_rgb_pca = fig_rgb_pca.add_subplot(111)
106 # ~ ax_rgb_pca.imshow(image_rgb_pca )
107
108
109
110 fig_orig3d = plt.figure(7)
111 ax_orig3d = fig_orig3d.add_subplot(111, projection='3d')
112 tmp=np.arange(image2.shape[0])
113 np.random.shuffle(tmp);
114
115 if(tmp.shape[0]>2000):
116     image2s=image2[tmp[:2000],:]
117 else:
118     image2s=image2[tmp,:]
119 ax_orig3d.scatter(image2s[:,0], image2s[:,1], image2s[:,2],s=1)
120
121 #draw mean
122 image2_mean=np.mean(image2,axis=0)
123 ax_orig3d.scatter(image2_mean[0], image2_mean[1], image2_mean[2],s=15, color='red')
124
125 #draw axes
126 X, Y, Z, U, V, W = axes_quiver_form(axes1, center1, explained_var1);# ps1_w1,ps1_w2, ps1_w3);
127 ax_orig3d.quiver(X,Y,Z,U,V,W,(5,2,1),arrow_length_ratio=0.1, color='red')
128
129
130 ax_orig.set_title("Original image")
131 ax_pca.set_title("BW: PC 1")
132 ax_pca2.set_title("BW: PC 2")
133 ax_pca3.set_title("BW: PC 3")
134
135 # ~ ax_rgb_pca.set_title("(R,G,B) = (PC1, PC2, PC3)")
136
137 ax_misc_bw.set_title("BW: scipy.misc")
138
139
140
141
142
143
144
145 ax_orig.legend()
146 ax_pca.legend()
147 ax_pca2.legend()
148 ax_pca3.legend()
149 ax_misc_bw.legend()
150
151
152 plt.show()
153
154
155
156
157
158
159
160

```

```
161 |
162 |
163 |
164 |
165 | quit()
166 |
```