# 实验一：数字图像处理相关软件学习

姓名：鲁国锐 学号：17020021031 专业年级：电子信息科学与技术专业2017级 日期：2020年03月13号

**实验目标：**

1. 熟悉Python图像处理的基本操作。
2. 使用Python制作一个gif动态图。
3. 使用Python用很多幅小的相关图片组成一副大的主题图片。

**实验要求：**

1. 使用Python语言，完成以下功能：
2. 打开一幅图片（如自己的照片）
3. 将图片大小修改成640\*480
4. 将修改大小后的图像转成黑白图像
5. 将图像存成gif格式
6. 使用Python语言,用很多幅小的相关图片组成一副大的主题图片，如，右图就是用很多表情笑脸图合成左图的效果。

**实验步骤（可以附加实验代码）：**

**任务一**

**思路：见实验要求1**

import numpy as np  
import matplotlib.pyplot as plt  
from PIL import Image, ImageSequence  
import imageio  
  
###############################################################  
#读取图像，转为灰度图，改变大小，保存  
###############################################################  
kobe = Image.open('kobe1.jpg')  
kobe = kobe.resize( ( int(kobe.size[0]\*0.5), int(kobe.size[1]\*0.5) ) )  
kobe.save('kobe\_small.jpg')  
kobe = kobe.convert('L')  
kobe.save('kobe\_gray.jpg')  
  
###############################################################  
# 制作gif的函数  
###############################################################  
def create\_gif(img\_list, gif\_name, duration):  
 frames = []  
 for img in img\_list:  
 frames.append(img)  
 imageio.mimsave(gif\_name+'.gif', frames, 'GIF', duration=duration)  
 return imageio.mimread('%s.gif' % gif\_name)  
###############################################################  
  
###############################################################  
# 读取多幅图像，逐个转为灰度图，改变大小  
###############################################################  
path\_list = [ 'jayzhou1.jpg', 'jayzhou2.jpg', 'jayzhou3.jpg']  
img\_list = []  
for i in path\_list:  
 img = Image.open(i)  
 img = img.convert('L')  
 img = img.resize((400, 300))  
 img\_list.append(img)  
 print(img.size)  
###############################################################  
# plt.figure("qilixiang")  
# plt.imshow(img)  
# plt.show()  
my\_gif = create\_gif(img\_list, 'jayzhou', 1)

**任务二**

**思路：**

* 1. 打开第一幅图片
  2. 将图片变小
  3. 将修改大小后的图像进行尺度变换，使其像素值在0~255
  4. 读取第二幅图像
  5. 改变大小使其长宽都是小图的整数倍
  6. 以大图每一点的像素值除以255作为权重，使它的像素值大小随着小图做周期性变化，沿长（或宽）的变化周期即为小图长度（或宽度）

import numpy as np  
import matplotlib.pyplot as plt  
from PIL import Image  
import imageio  
  
# 注意这里dim[0]指的是列数，dim[1]指的是行数  
# 转成np数组后会有一次转置  
dim = (8, 8)  
  
smile = Image.open('smile.jpg')  
smile = smile.resize( dim ).convert('RGB')  
smile = np.array(smile)  
  
#########################################################################  
# 对笑脸做尺度变换，使其像素值在0~255之间  
#########################################################################  
min\_smile = [256, 256, 256]  
max\_smile = [0, 0, 0]  
  
for i in range( len(smile) ):  
 for j in range( len(smile[i]) ):  
 for c in range(0, 3):  
 if smile[i][j][c] < min\_smile[c]:  
 min\_smile[c] = smile[i][j][c]  
 elif smile[i][j][c] > max\_smile[c]:  
 max\_smile[c] = smile[i][j][c]  
  
for i in range( len(smile) ):  
 for j in range( len(smile[i]) ):  
 for c in range(0, 3):  
 smile[i][j][c] = 255 / (max\_smile[c] - min\_smile[c]) \* (smile[i][j][c] - min\_smile[c])  
#########################################################################  
  
  
  
elephant = Image.open('01.jpg')  
print(elephant.size)  
elephant = elephant.resize( (elephant.size[0]//dim[0]\*dim[0], elephant.size[1]//dim[1]\*dim[1]) )  
print(elephant.size)  
elephant = elephant.convert('RGB')  
elephant = np.array(elephant)  
  
  
for i in range( len(elephant) ):  
 for j in range( len(elephant[i]) ):  
 for c in range(0, 3):  
 temp = float(elephant[i][j][c])  
 temp = temp \* smile[ i % dim[1] ][ j % dim[0] ][c] / 255.0  
 elephant[i][j][c] = temp  
  
plt.figure('elephant')  
plt.imshow(elephant)  
plt.show()  
  
im = Image.fromarray(elephant)  
print(im.size)  
im.save('res.jpg')

**实验结果：**

****

图 1原始图像

****

图 2改变大小之后的图像

****

图 3灰度化之后的图像

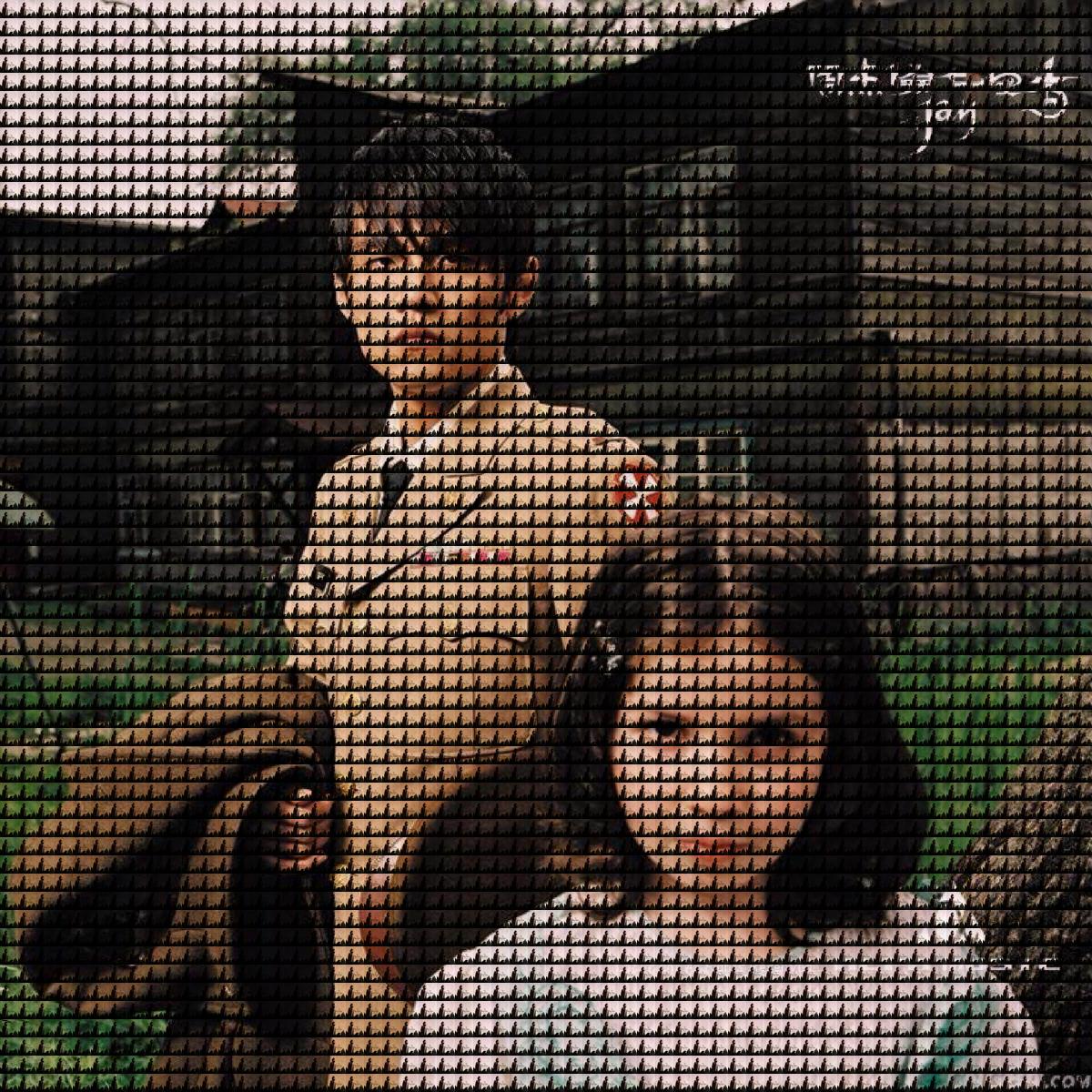


图 4 由《十一月的肖邦》海报拼成的《七里香》海报1



图 5 由《十一月的肖邦》海报拼成的《七里香》海报2



图 6 主题图像中的小图：《十一月的肖邦》海报



图 7 主体图像中的大图：《七里香》海报



图 8 由笑脸拼成的大象

**实验总结：**

本次实验共分两部分。第一部分主要是熟悉Python图像处理的基本操作；第二部分相对比较有挑战性，最终以大图像素作为权重来改变小图像素的方法实现了，其中还用到了尺度变换的方法，相对而言综合性比较强。

同时对于实验二，根据实验结果可以看到，主体图像呈现出的效果与小图明暗局域所占比重关系比较大。用《十一月的肖邦》海报合成的《七里香》海报，比原来要暗不少，原因应是《十一月的肖邦》整体色调偏暗，且深色区域面积比较大。将图像放大后可以看出，每一个小海报中深色区域（如人，建筑等）都大致保持了原来的样子，而浅色区域则是跟大图对应位置的图案比较相近。分析其原因，应当是主题图像每一点的像素值都是大图和小图相乘的结果：小图深色区域像素值小，所以不管原图在该点像素值多大，相乘后得到的结果都很小；而浅色区域像素值大，故此时乘积很大程度上取决于加给它的权重的大小，即大图在该位置的像素值大小。

另外也可以看出，主题图像中，小图越小，合成出来的结果就越接近大图本来的样子，但同时整体上的色调也会受小图更多的影响。如上述实验结果中，将《十一月的肖邦》海报进一步缩小后，合成出来的主题图像在图案上更接近原《七里香》海报，但整体色调也更暗了，即更接近于《十一月的肖邦》的色调。