

实验一：数字图像处理相关软件学习

姓名：胡帅 学号：17020031016 专业年级：计算机专业 2017 级 日期：2020.3.15

实验目标：

Matlab 是干啥的？

BMP 图像的结构是什么样的？

用 VC 怎么写图像处理的程序？

实验要求：

1. 通过使用学习 Matlab 的联机帮助。写个简单的 Matlab 程序（或者 Python），完成以下功能：

- 打开一幅图片（如自己的照片）
- 将图片大小修改成 640*480
- 将修改大小后的图像转成黑白图像
- 将图像存成 gif 格式

2. 不知道大家还记不记得奥运开幕式中那一幅幅小的笑脸照片组成大大的笑脸环节，其实日常生活中也有很多这样类似的创意，用很多幅小的相关图片组成一副大的主题图片，



右图就是用很多表情笑脸图合成左图的效果。考虑一下如果自己写程序的话，怎么来实现。

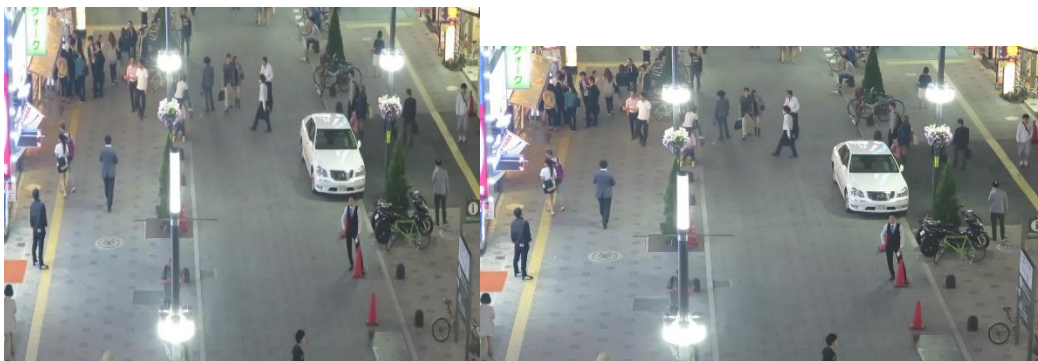
实验步骤：

1. 打开一幅图片

```
1. import cv2
2. import numpy as np
3.
4. img = cv2.imread("./1.jpg")
5. cv2.imshow("Image", img)
6. cv2.waitKey(0)
```

2. 将图片大小修改成 640*480

```
1. import cv2
2.
3. im1 = cv2.imread('./original.jpg')
4. cv2.imshow('image1', im1)
5. cv2.waitKey(0)
6.
7. im2 = cv2.resize(im1, (640,480), interpolation=cv2.INTER_CUBIC)
8. cv2.imshow('image2', im2)
9. cv2.waitKey(0)
10. cv2.imwrite('resize.jpg', im2)
```



3. 将修改大小后的图像转成黑白图像

```
1. import cv2
2. import numpy as np
3.
4. if __name__ == "__main__":
5.     img_path = "resize.jpg"
6.     img = cv2.imread(img_path)
7.     #获取图片的宽和高
8.     width,height = img.shape[:2][::-1]
9.     #将图片缩小便于显示观看
10.    img_resize = cv2.resize(img,
11.        (int(width*0.5),int(height*0.5)),interpolation=cv2.INTER_CUBIC)
12.    cv2.imshow("img",img_resize)
13.    print("img_reisze shape:{}".format(np.shape(img_resize)))
14.
15.    #将图片转为灰度图
16.    img_gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
17.    cv2.imshow("img_gray",img_gray)
18.    print("img_gray shape:{}".format(np.shape(img_gray)))
19.    cv2.waitKey()
20.    cv2.imwrite('resize_gray.jpg', img_gray)
```



4. 将图像存成 gif 格式

```
1. import imageio
2.
3. def create_gif(image_list, gif_name):
4.
5.     frames = []
6.     for image_name in image_list:
7.         frames.append(imageio.imread(image_name))
8.     # Save them as frames into a gif
9.     imageio.mimsave(gif_name, frames, 'GIF', duration = 0.1)
10.
11.     return
12.
13. def main():
14.     image_list = ['original.jpg', 'resize_gray.jpg', 'resize.jpg']
15.     gif_name = 'created_gif.gif'
16.     create_gif(image_list, gif_name)
17.
18. if __name__ == "__main__":
19.     main()
```

生成的动图见附件，word 无法演示 gif

5. 千图成像

千图成像：用 N 张图片拼凑成一张图片。

实现原理：先将所要成像的图片转化成马赛克图片，然后从图库中用对应颜色的图片替换相应色块。

图库中的图片处理：标记图库中每张图片的混合颜色，用于替换目标色块，并记录每张图片的特征用于成像，增加成像质量。

```

1. function zoomphoto( )
2. %UNTITLED 此处显示有关此函数的摘要
3. % 此处显示详细说明
4. img=imread('1.jpg');
5. imshow(img);
6. r=img(:,:,1);
7. g=img(:,:,2);
8. b=img(:,:,3);
9. [h w1]=size(img);
10. w=w1/3;
11. imgr=zeros(h,w);
12. imgg=zeros(h,w);
13. imgb=zeros(h,w);
14. psize=10;
15. ww=floor(w/psize)*psize;
16. hh=floor(h/psize)*psize;
17. file_path = '.\images\';% 图像文件夹路径
18. img_path_list = dir(strcat(file_path,'*.jpg'));%获取该文件夹中所有 jpg 格式的图像
19. img_num = length(img_path_list);%获取图像总数量
20. imgmatr=[psize,psize*img_num];
21. imgmatg=[psize,psize*img_num];
22. imgmatb=[psize,psize*img_num];
23. ave=[4,img_num];
24. if img_num > 0 %有满足条件的图像
25.     for j = 1:img_num %逐一读取图像
26.         image_name = img_path_list(j).name;% 图像名
27.         image = imread(strcat(file_path,image_name));
28.         imagezoom=imresize(image,[psize,psize]);
29.         imgmatr(1:psize,j*psize-psize+1:j*psize)=imagezoom(:,1);
30.         imgmatg(1:psize,j*psize-psize+1:j*psize)=imagezoom(:,2);
31.         imgmatb(1:psize,j*psize-psize+1:j*psize)=imagezoom(:,3);
32.         rr=imagezoom(:,1);
33.         gg=imagezoom(:,2);
34.         bb=imagezoom(:,3);
35.         ave(j,1)=mean(rr(:));
36.         ave(j,2)=mean(gg(:));
37.         ave(j,3)=mean(bb(:));
38.         ave(j,4)=0;
39.
40.     end
41. end
42. for x=1:psize:hh
43.     for y=1:psize:ww
44.         imgr(x:x+psize-1,y:y+psize-1)=mean(mean(r(x:x+psize-1,y:y+psize-1)));

```

```

45.         imgg(x:x+psize-1,y:y+psize-1)=mean(mean(g(x:x+psize-1,y:y+psize-1)));
46.         imgb(x:x+psize-1,y:y+psize-1)=mean(mean(b(x:x+psize-1,y:y+psize-1)));
47.     end
48.
49.     imgr(x:x+psize-1,ww+1:w)=mean(mean(r(x:x+psize-1,ww+1:w)));
50.     imgg(x:x+psize-1,ww+1:w)=mean(mean(g(x:x+psize-1,ww+1:w)));
51.     imgb(x:x+psize-1,ww+1:w)=mean(mean(b(x:x+psize-1,ww+1:w)));
52.
53. end
54.
55. for y=1:psize:ww
56.     imgr(hh+1:h,y:y+psize-1)=mean(mean(r(hh+1:h,y:y+psize-1)));
57.     imgg(hh+1:h,y:y+psize-1)=mean(mean(g(hh+1:h,y:y+psize-1)));
58.     imgb(hh+1:h,y:y+psize-1)=mean(mean(b(hh+1:h,y:y+psize-1)));
59.
60. end
61.
62. imgr(hh+1:h,ww+1:w)=mean(mean(r(hh+1:h,ww+1:w)));
63. imgg(hh+1:h,ww+1:w)=mean(mean(g(hh+1:h,ww+1:w)));
64. imgb(hh+1:h,ww+1:w)=mean(mean(b(hh+1:h,ww+1:w)));
65. newr=imgr;
66. newg=imgg;
67. newb=imgb;
68. for x=1:psize:hh
69.     for y=1:psize:ww
70.         min=(imgr(x,y)-ave(1,1))^2+(imgg(x,y)-ave(1,2))^2+(imgb(x,y)-ave(1,3))^2;
71.         minz=0;
72.         for z=1:img_num
73.             distance=(imgr(x,y)-ave(z,1))^2+(imgg(x,y)-ave(z,2))^2+(imgb(x,y)-
                ave(z,3))^2;
74.             if distance<min&&ave(z,4)<10
75.                 min=distance;
76.                 minz=z-1;
77.             end
78.         end
79.         ave(minz+1,4)=ave(minz+1,4)+1;
80.         newr(x:x+psize-1,y:y+psize-1)=imgmatr(1:psize,minz*psize+1:minz*psize+psize);
81.         newg(x:x+psize-1,y:y+psize-1)=imgmatg(1:psize,minz*psize+1:minz*psize+psize);
82.         newb(x:x+psize-1,y:y+psize-1)=imgmatb(1:psize,minz*psize+1:minz*psize+psize);
83.     end
84. end
85. figure;
86. newr=uint8(newr);
87. newg=uint8(newg);

```

```
88. newb=uint8(newb);  
89. imshow(cat(3,newr,newg,newb));  
90.  
91. end
```

