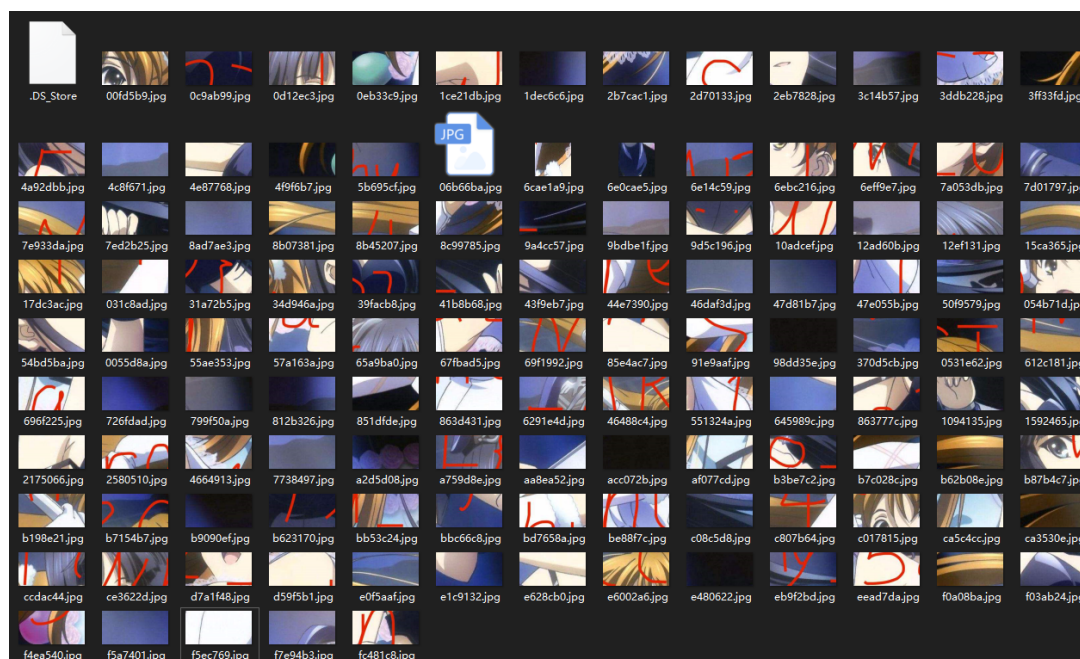


# [MISC]拼图题

打算不再盲目刷题,而是总结一些东西,最近打比赛时遇到拼图题多了,想想以前也有遇到过,就对工具和情况做一个总结

## [MRCTF2020]不眠之夜



这里首先可以使用linux下的montage命令

- 1 安装方法:
- 2 -bash: montage: command not found
- 3 #Debian
- 5 apt-get install graphicsmagick-imagemagick-compat
- 6 #Ubuntu
- 8 apt-get install graphicsmagick-imagemagick-compat
- 10 #Alpine
- 11 apk add imagemagick6
- 12 #Arch Linux
- 14 pacman -S imagemagick6
- 16 #Kali Linux
- 17 apt-get install graphicsmagick-imagemagick-compat
- 18 #Raspbian
- 20 apt-get install graphicsmagick-imagemagick-compat
- 22 #Docker
- 23 docker run cmd.cat/montage montage

montage 命令通过组合几个单独的图像来创建合成图像。图像平铺在合成图像上,并带有边框,框架,图像名称等。该 montage 程序是 ImageMagick 的工具套件。通过组合几个单独的图像,使用它来创建合成图像。

```

1  montage 命令实例:
2  montage 将图像平铺到网格中, 自动调整大于网格单元格大小的图像的大小:
3  montage image1.png image2.jpg imageN.png montage.jpg
5  montage 将图像平铺到网格中, 然后根据最大图像自动计算网格单元大小:
6  montage image1.png image2.jpg imageN.png -geometry +0+0 montage.jpg
8  montage 设置网格单元的大小, 并在平铺之前调整图像的大小以适合它:
9  montage image1.png image2.jpg imageN.png -geometry 640x480+0+0 montage.jpg
10 montage 限制网格中的行和列数, 从而导致输入图像溢出到多个输出蒙太奇中:
12 montage image1.png image2.jpg imageN.png -geometry +0+0 -tile 2x3
    montage_%d.jpg
13 montage 在平铺之前调整图像大小并裁剪图像以完全填充其网格单元:
15 montage image1.png image2.jpg imageN.png -geometry +0+0 -resize 640x480^ -
    gravity center

```

在这道题里就可以先用montage命令把图片拼起来

```

1  montage *.jpg -tile 10x12 -geometry 200x100+0+0 output.jpg
2  -tile 定义行列数,中间是x不是*
4  -geometry 首选每个图和边框尺寸,无边框就默认+0+0

```

在执行命令时发现有一张图片是干扰项 (06b66ba.jpg)



再通过另一个工具gaps进行图片排序还原,如果安装失败自行百度解决方法

```

1  安装方法:(python2.7)
2  git clone https://github.com/nemanja-m/gaps.git
3  cd gaps
4  pip install -r requirements.txt
5  sudo apt-get install python-tk
6  pip install -e .

```

gaps是一种用于拼图基于遗传算法和OpenCV的求解器,能对生成图进行修正

```

1  gaps --image=output.jpg --generations=40 --population=120 --size=100
2  --image 指明修正图
4  --generations 指明遗传算法的代数
5  --population 指明裁切图片数
6  --size 指明生成图的像素大小
7  --verbose 在每一代之后显示最佳解决方案
8  --save 将拼图解决方案另存为图像

```



## 将图片裁切成拼图

先使用create\_puzzle工具将图片随机打乱

```
1 create_puzzle ./keyboard.jpg --size=48 --destination=puzzle.jpg
2 将images/pillars.jpg图片进行裁切
4 --size 指定裁切出来的图片像素大小
5 --destination 指定裁切出来的图片名称
```



想了想还是利用下面的pillow脚本来裁切

```
1 from PIL import Image
2 # name_list可随便定义一组随机数,这里就直接套数独的list了
3
4 name_list = [[11, 27, 32, 44, 58, 63, 76, 85, 99],
5              [13, 25, 38, 49, 56, 61, 77, 82, 94],
6              [14, 29, 36, 47, 52, 65, 71, 88, 93],
7              [12, 24, 37, 48, 51, 69, 75, 83, 96],
8              [19, 23, 35, 46, 57, 64, 72, 81, 98],
9              [18, 26, 31, 45, 53, 62, 79, 84, 97],
10             [17, 22, 34, 43, 55, 66, 78, 89, 91],
11             [16, 21, 39, 42, 54, 68, 73, 87, 95],
12             [15, 28, 33, 41, 59, 67, 74, 86, 92]]
13
14 img = Image.open("./image.png") # 读取要操作的图片
15
16 width, height = img.size         # 获取图片的宽度和高度
17
18 crop_width = width // 9          # 截切后图片的宽度
19
20 crop_height = height // 9        # 截切后图片的高度
```

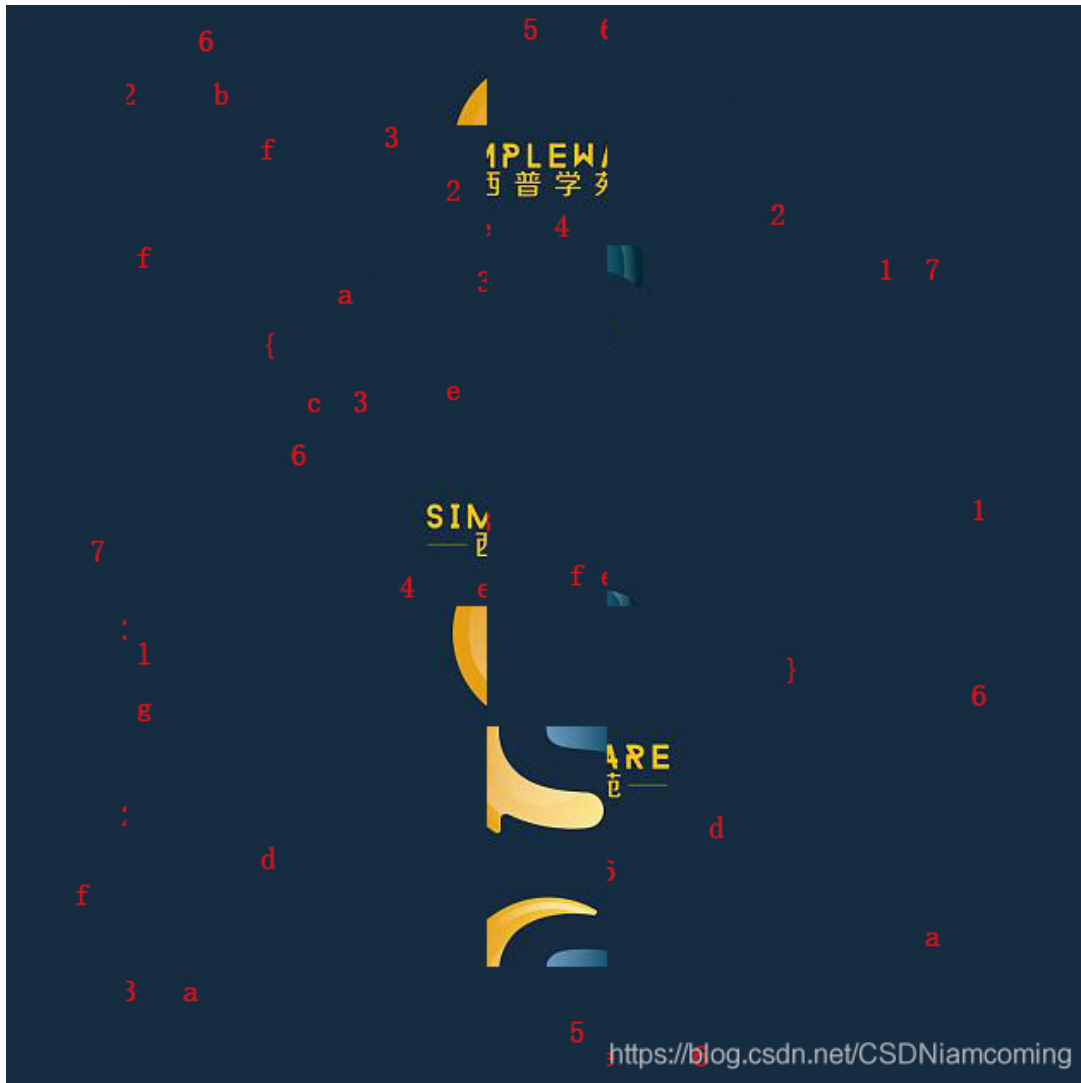
```

20 # 利用循环将原图片截成 9*9 个图片，并保存
22 for i in range(9):
23     for j in range(9):
24         box = (crop_width * j, crop_height * i, crop_width * (j + 1),
25               crop_height * (i + 1)) # 截切下来的图片在原图片的位置 (left, up, right,
26                                     down)
27         new_img = img.crop(box)
28         name = "./image/" + str(name_list[i][j]) + ".png"
29         new_img.save(name)

```

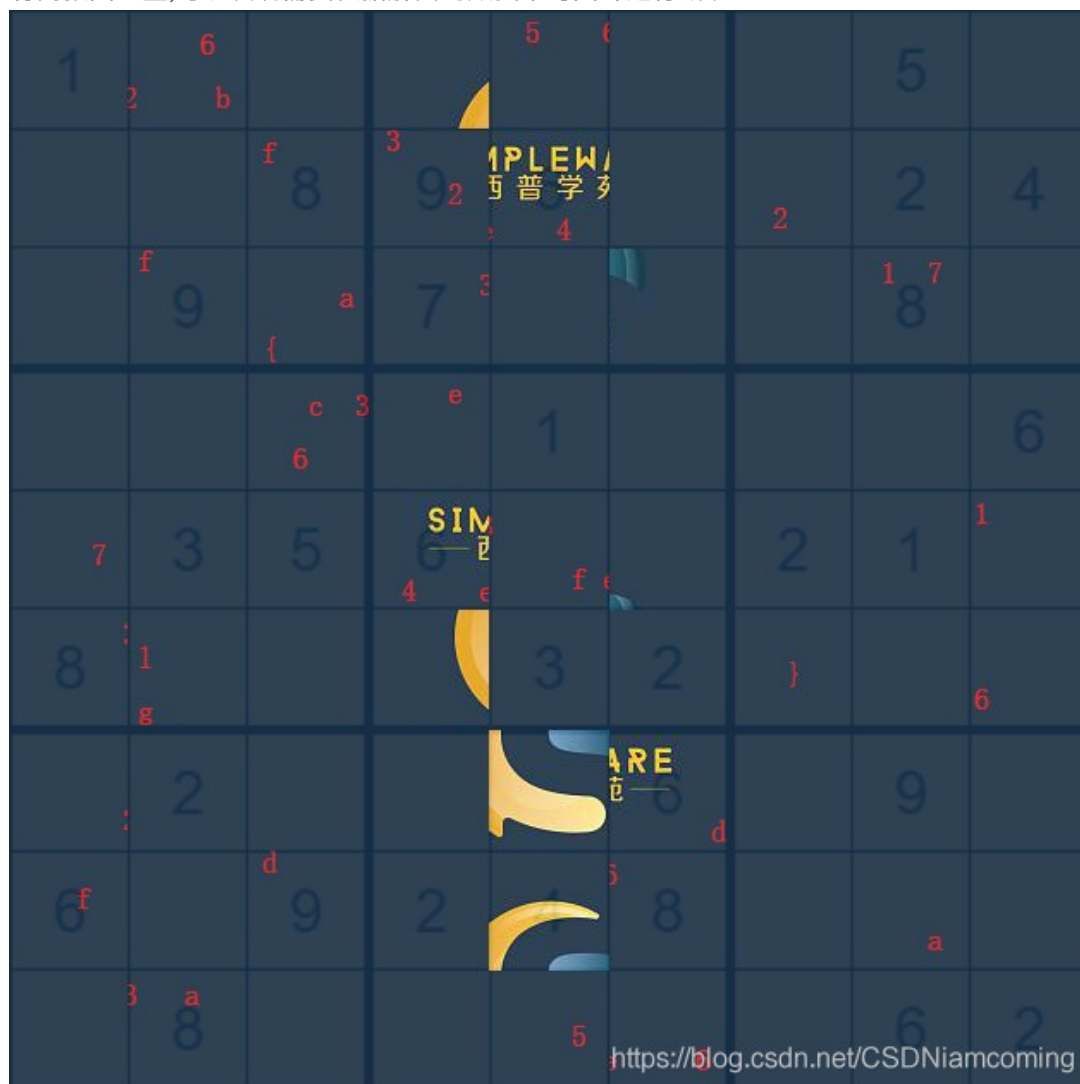
## pillow模块切割、合并图片

看到别人做的一道题,得到的是这两张图片



1							5	
		8	9	6			2	4
	9		7				8	
				1				6
	3	5	6			2	1	
8				3	2			
	2				6		9	
6		9	2	4	8			
	8						6	2

将两张图重叠,可以发现需要根据解出的数独来对图片进行裁切重组



根据解数独脚本结合pillow模块还原图片

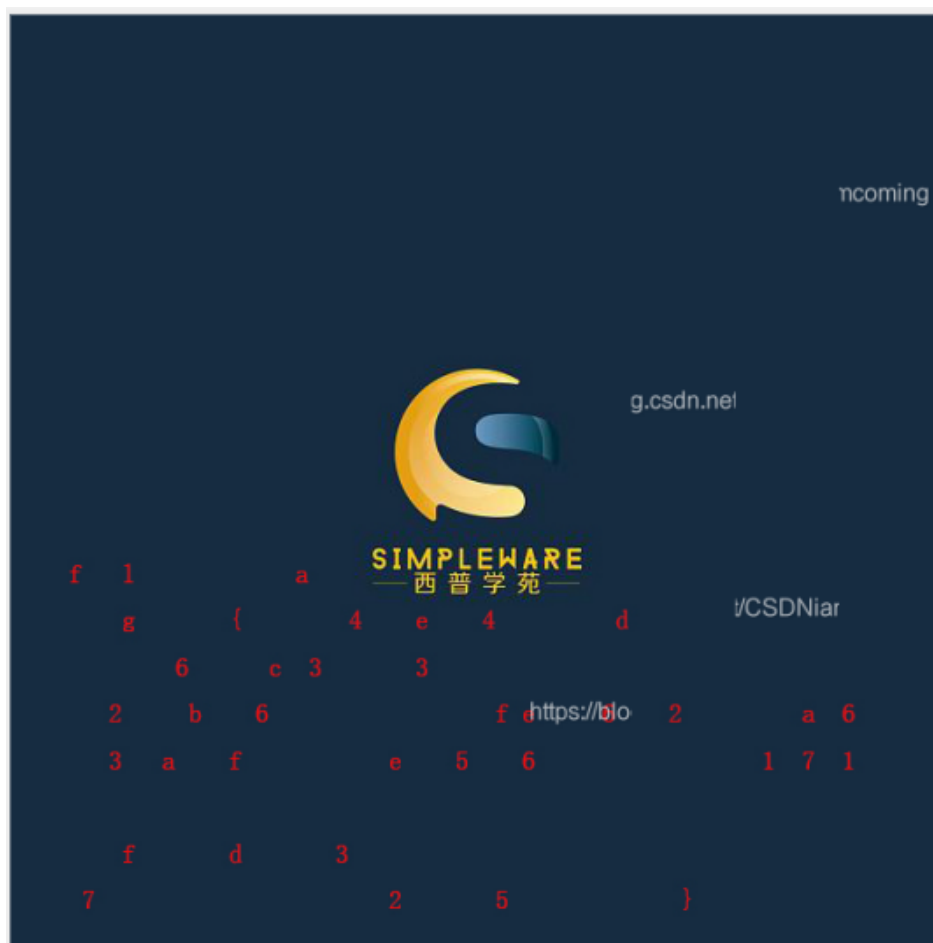
```

1 from PIL import Image
2 # 为了防止图片名重复,在解得的数独前加上列号

```



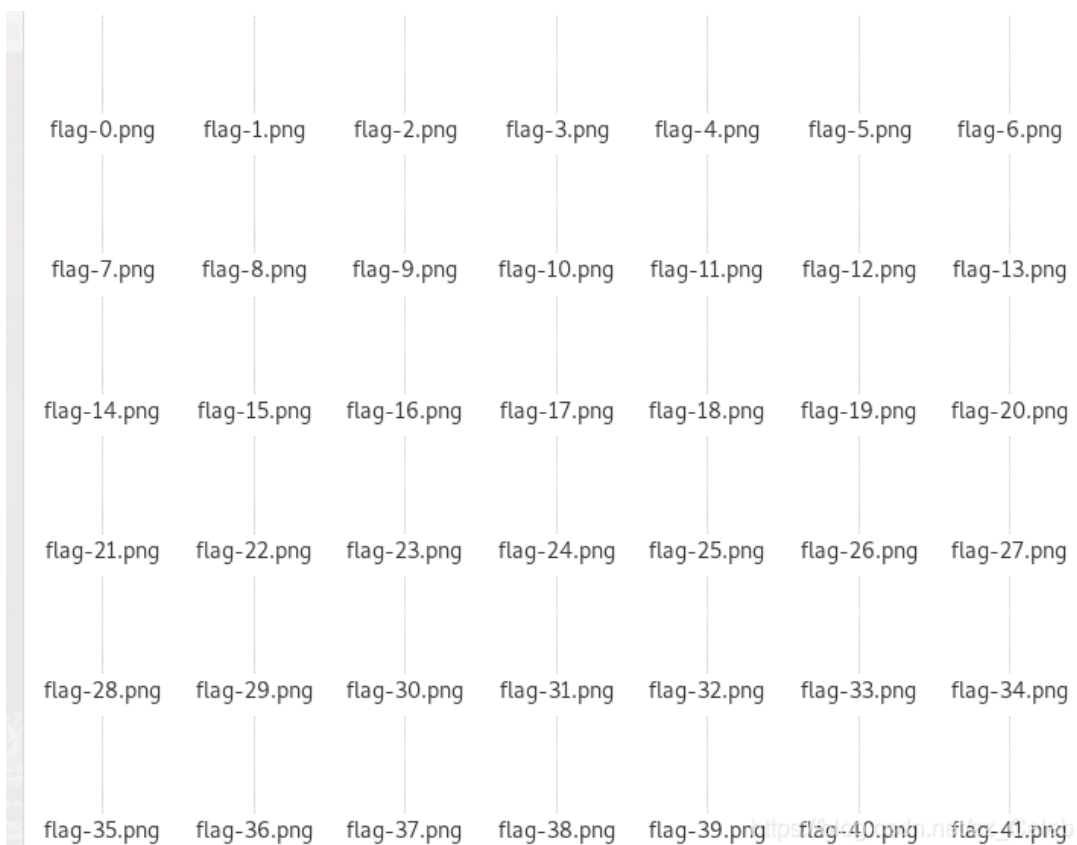
```
4 name_list = [[11, 27, 32, 44, 58, 63, 76, 85, 99],
5             [13, 25, 38, 49, 56, 61, 77, 82, 94],
6             [14, 29, 36, 47, 52, 65, 71, 88, 93],
7             [12, 24, 37, 48, 51, 69, 75, 83, 96],
8             [19, 23, 35, 46, 57, 64, 72, 81, 98],
9             [18, 26, 31, 45, 53, 62, 79, 84, 97],
10            [17, 22, 34, 43, 55, 66, 78, 89, 91],
11            [16, 21, 39, 42, 54, 68, 73, 87, 95],
12            [15, 28, 33, 41, 59, 67, 74, 86, 92]]
13 image_path = "./image.png" # 读取要操作的图片
14 img = Image.open(image_path)
15 width, height = img.size # 获取图片的宽度和高度
16 crop_width = width // 9 # 截切后图片的宽度
17 crop_height = height // 9 # 截切后图片的高度
18 # 创建一个新图片
19 image = Image.new('RGB', (width, height))
20 # 利用循环将原图片截成 9*9 个图片
21 for i in range(9):
22     for j in range(9):
23         box = (crop_width * j, crop_height * i, crop_width * (j + 1),
24               crop_height * (i + 1))
25 # 利用循环将所有图片拼接到新图片上
26         new_img = img.crop(box)
27         y = int(str(name_list[i][j])[0:1])
28         x = int(str(name_list[i][j])[1:2])
29         box2 = (crop_width * (y - 1), crop_height * (x - 1), crop_width *
30               y, crop_height * x)
31         image.paste(new_img, box2)
32 # 调用图片查看工具展示图片
33 image.show()
34 # image.save("./image/flag.png") # 保存图片
```



## 攻防世界–Misc–glance

convert 分离gif、montage 图片拼接

```
1 convert glance.gif flag.png
```



```
1 montage flag*.png -tile x1 -geometry +0+0 flag.png
```



