

Misc_常见拼图手法

原创

M3ng@L 于 2022-01-11 10:34:52 发布 283 收藏

分类专栏: [杂项知识总结](#) 文章标签: [python](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_51999772/article/details/122425898

版权



[杂项知识总结](#) 专栏收录该内容

10 篇文章 1 订阅

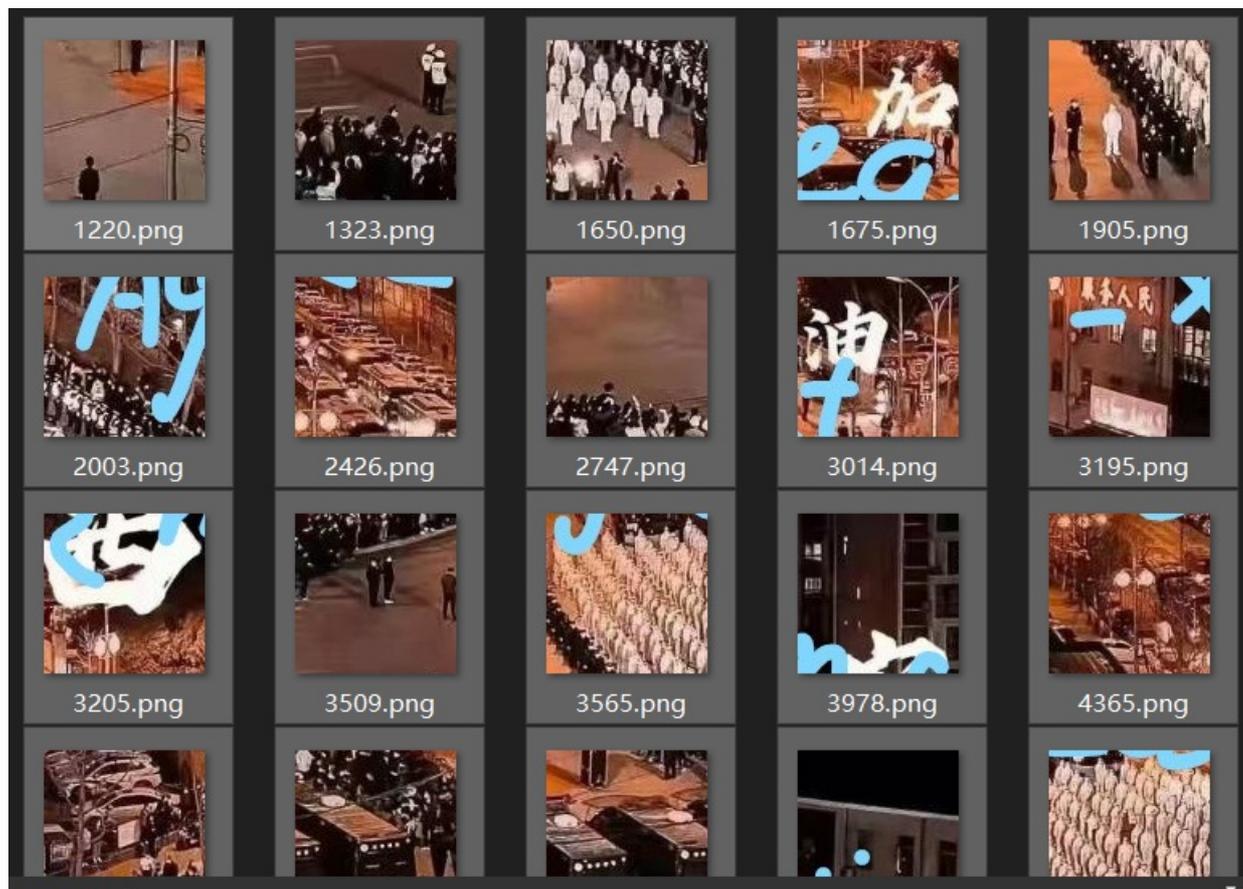
订阅专栏

Misc_拼图手法

montage

montage是用于将多个图片合并为一个图像的

效果是这样的:



使用命令后合并为这样



图像是乱序的，但是不要紧，之后会用到gaps命令进行整合

命令格式

语法

```
montage input_file [options] output_file
```

简单的拼接1.png, 2.png, 3.png的图像生成成为out.png（注意三个图像之间会有很大空隙，这是因为我们还没设置图片之间的间距参数）

```
montage 1.png 2.png 3.png out.png
```

或者是使用正则匹配，匹配目标图像（用于图像很多的时候）

```
montage *.png out.png
```

接下来就是一些常用的相关参数

```
-geometry #  
-tile # 设置每行每列的原始图片数量
```

一般使用格式

```
montage input_file -tile 8X6 -geometry +0+0 output_file
```

这里"-geometry +0+0"的用处是让图片之间没有间隙

“-tile 8X6”的作用是让原始图片按12*4的排列方式排列（也就是8行6列）

```
menglin@kali:~/桌面/chips$ montage *.png -tile 8X6 -geometry +0+0 ../out.png
```

参考文章

[Linux montage 命令 command not found montage 未找到命令 montage 命令详解 montage 命令未找到 montage 命令安装 - CommandNotFound](#) 坑否

gaps

作用是还原拼图，也就是我们刚刚用montage命令创建的乱序的图像

命令格式

```
python3 gaps --image=input_image --generations=x --size=y --population=z
```

经过我自己的试验

“-generations”设置的参数多少都可以，最好就等于原始图片的数量，也就是piece的数量

“-size”这个参数是最关键的，程序会根据这个选项设置的参数来判断piece的数量

举个例子，原始的图片量是48，当我设置size为100时，程序返回的Pieces为48，最后拼接的图像也是对的

```
root@kali:/home/menglin/桌面/misc_steg/Pick_Picture/gaps/bin# python3 gaps --image=/home/menglin/桌面/out.png --generations=48 --size=100 --population=50
=== Population: 50
=== Generations: 48
=== Piece size: 100 px
=== Pieces: 48
```

当设置size为150时，程序返回的Pieces为20，这个时候返回的图片仍然是乱序的

```
root@kali:/home/menglin/桌面/misc_steg/Pick_Picture/gaps/bin# python3 gaps --image=/home/menglin/桌面/out.png --generations=48 --size=150 --population=50
=== Population: 50
=== Generations: 48
=== Piece size: 150 px
=== Pieces: 20
```

“-population”选项设置的参数似乎对返回的结果没有太大影响~~（ctf_misc的话差不多看的清楚就可了）~~，但是当参数设置比较大的时候，运行速度会慢一点点，所以建议这个参数值设置小一点

最后就返回正确结果



参考文章

[Misc工具（10.9更） - Ga1@xy's Wor1d \(ga1axy.top\)](#)