关于简单目标识别与意图分析的机器学习实战研究（降噪处理）

为了实现目标精确识别，颜色分割只是其中的一种方法，对于大多数图片，我们还是更倾向于对图片进行高斯模糊去噪的方法，此外网上还有很多其他的方法，这里只介绍我认为好用的这一种。

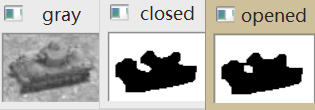
为什么选取高斯模糊去噪的方法呢？

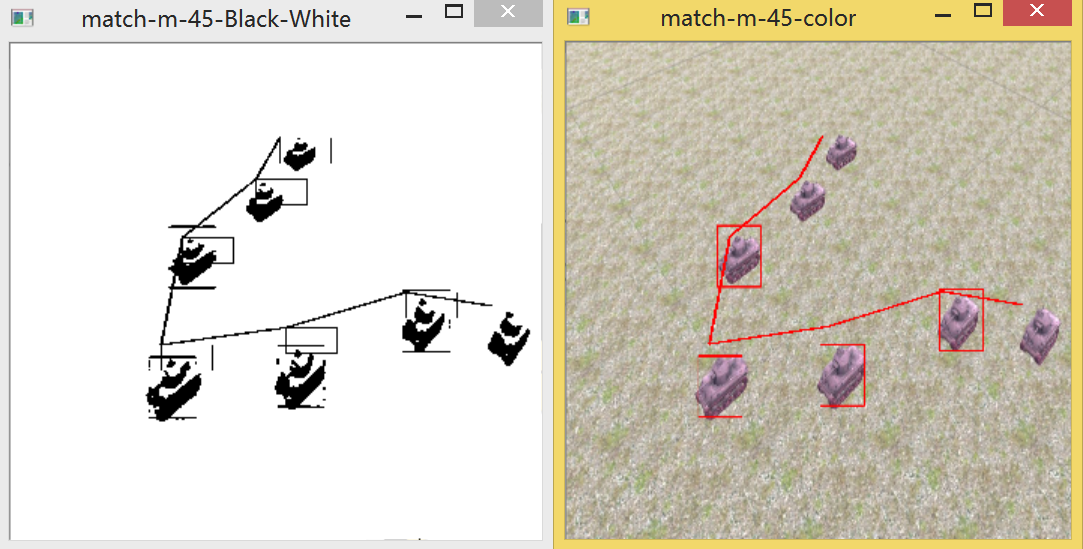
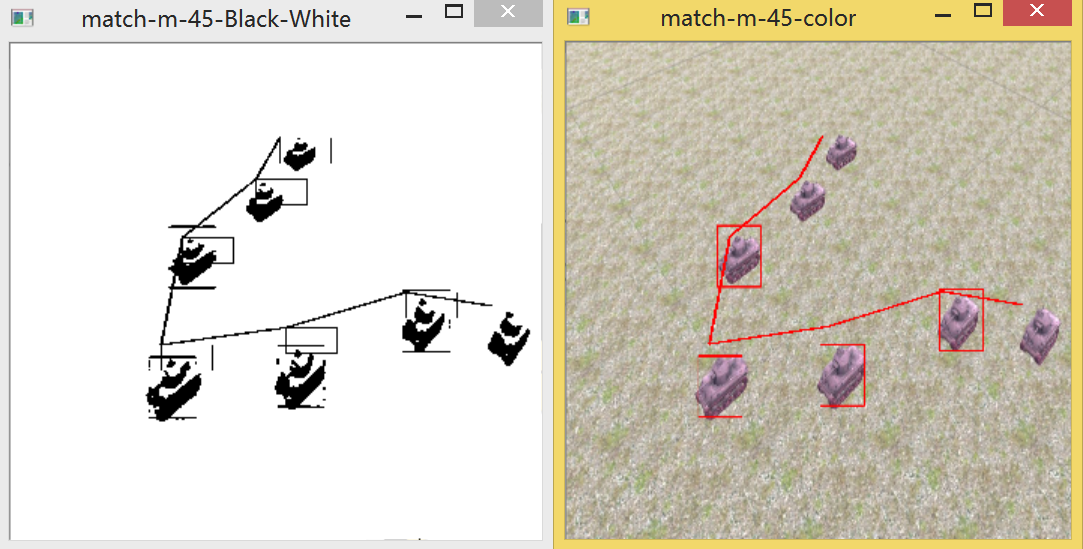
因为高斯模糊去噪的时候，函数内部的判别阈值可以自行设定，阈值的大小直接影响了去噪的强弱（这里不能说是去噪好坏，因为根据不同图片，想要达到的效果也不同）。有时候匹配的效果不尽人意，修改一下阈值重新运行一遍就好了。

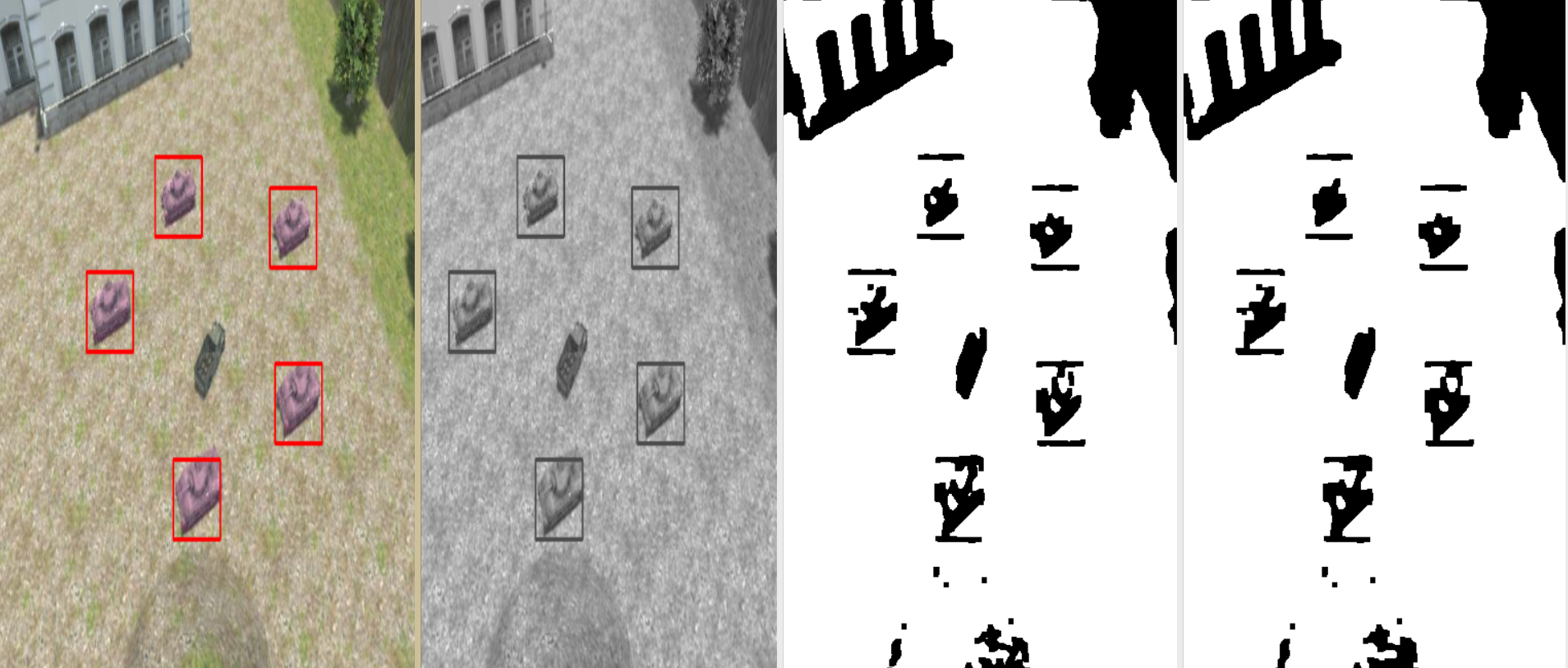
我们这里最终要做的还是要提高模板匹配的精度，使在有干扰的情况下完成模板匹配，所以在降噪的时候为了保持模板和目标内容的同步，需要将模板和目标图片同时进行降噪处理，之后的匹配也会在降噪后的图片之间进行。一言不合就上代码：

1. **def** Img\_Outline(original\_img):
2. gray\_img = cv.cvtColor(original\_img, cv.COLOR\_BGR2GRAY)
3. blurred = cv.GaussianBlur(gray\_img, (9, 9), 0)                    # 高斯模糊去噪（设定卷积核大小影响效果）
4. \_, RedThresh = cv.threshold(blurred, 140, 255, cv.THRESH\_BINARY)  # 设定阈值，对识别很重要，可以手动调节识别效果（阈值影响开闭运算效果）12
5. kernel = cv.getStructuringElement(cv.MORPH\_RECT, (5, 5))          # 定义矩形结构元素
6. closed = cv.morphologyEx(RedThresh, cv.MORPH\_CLOSE, kernel)       # 闭运算（链接块）
7. opened = cv.morphologyEx(closed, cv.MORPH\_OPEN, kernel)           # 开运算（去噪点）
8. **return** original\_img, gray\_img, RedThresh, closed, opened
10. ###原图
11. tpl ="D://PythonPicTemplate/binarization/tankTemplate3.jpg"
12. target = "D://PythonPicTemplate/binarization/tanks4.jpg"
13. tpl = Img\_read(tpl)
14. target = Img\_read(target)
15. red = (0, 0, 255)
16. green = (0,255,0)
17. col = red
18. thickness = 5
20. methods = []
21. **for** o **in** range(0,num):
22. methods.append(cv.TM\_CCORR\_NORMED)
23. '''''
24. 差值平方和匹配 CV\_TM\_SQDIFF
25. 标准化差值平方和匹配 CV\_TM\_SQDIFF\_NORMED
26. 相关匹配 CV\_TM\_CCORR
27. 标准相关匹配 CV\_TM\_CCORR\_NORMED
28. 相关匹配 CV\_TM\_CCOEFF
29. 标准相关匹配 CV\_TM\_CCOEFF\_NORMED
30. '''
31. th, tw = tpl.shape[:2]
32. rows, cols = target.shape[:2]
33. content1 = "正在分析模板大小："
34. printGreen(content1.decode("utf-8").encode("gbk") )
35. **print** "TemplateSize: ",th, tw
36. content2 = "正在分析目标大小："
37. printGreen(content2.decode("utf-8").encode("gbk"))
38. **print** "TargetSize: ", rows,cols
40. i =0
41. tl0 = []
42. br0 = []
43. **for** md **in** methods:
44. #print(md)
45. result = cv.matchTemplate(target, tpl, md)
46. min\_val, max\_val, min\_loc, max\_loc = cv.minMaxLoc(result)
48. **if** md == cv.TM\_SQDIFF\_NORMED:
49. tl0.append(min\_loc)
50. **else**:
51. tl0.append(max\_loc)
53. br0.append((tl0[i][0]+tw, tl0[i][1]+th))
54. cv.rectangle(target, tl0[i], br0[i], (0, 0, 255),2)
55. i += 1
56. #cv.rectangle(tpl,(0,0),(th,tw),(0,0,255),2)
58. original\_img, gray\_img, RedThresh, closed, opened = Img\_Outline(tpl)
59. original\_imgT, gray\_imgT, RedThreshT, closedT, openedT = Img\_Outline(dst)
61. #cv.imshow("original", original\_img)
62. #cv.imshow("gray", gray\_img)
63. #cv.imshow("closed", closed)
64. cv.namedWindow("opened", cv.WINDOW\_NORMAL)
65. cv.imshow("opened", opened)
67. #cv.imshow("original", original\_imgT)
68. #cv.imshow("grayT", gray\_imgT)
69. #cv.imshow("closedT", closedT)
70. cv.namedWindow("openedT", cv.WINDOW\_NORMAL)
71. cv.imshow("openedT", openedT)

结果如下所示：







下一次将使用特定算法对坦克军团特定连线处理，以及生成坦克阵型简图。