# 检测程序总结

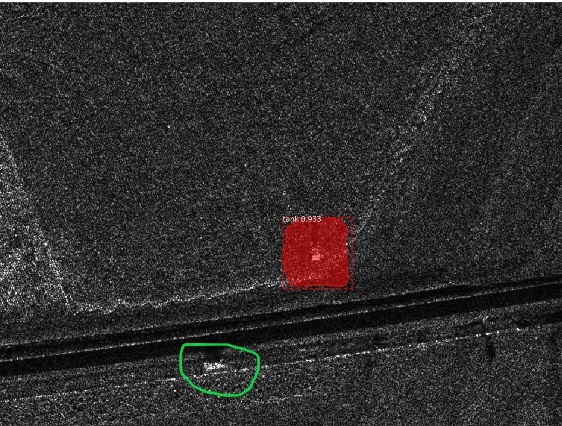
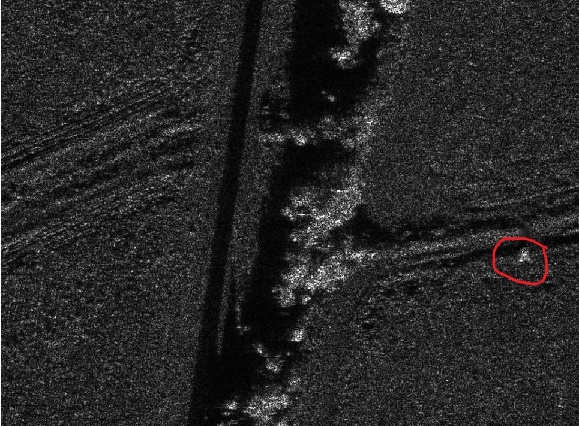
数据集：1000张坦克仿真图

划分：训练：测试=8:2

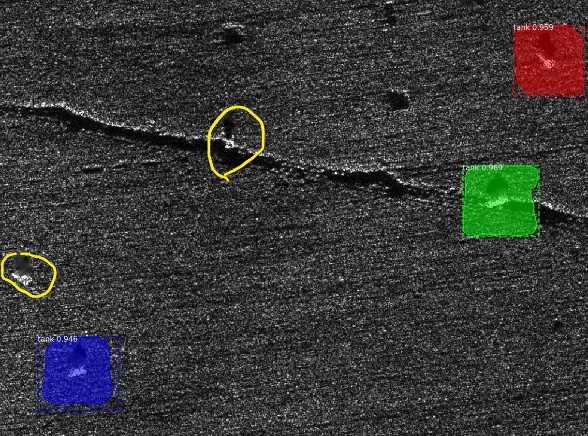
说明：检测结果图中单以线圈出的区域为未检测到的坦克所在，以不同颜色块填充的区域为程序检测出的坦克所在。

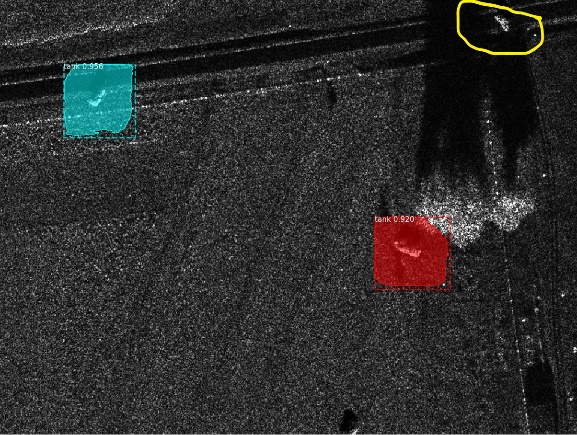
误检和漏检可能原因分析：

1. 框中包含其他纹路。由于数据集中的每个标记数据是包含了坦克所在区域在内的一块区域，故而相当于在数据集中添加了多个隐藏类别，导致检测时有一定难度。

示例图像：

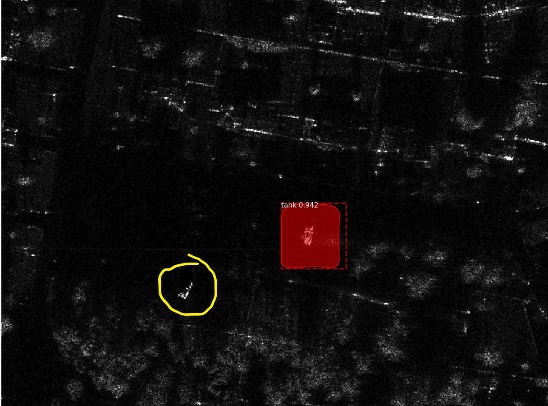
上图可以看到在靠近道路等其他纹路信息较为突出的目标时使得检测未成功。

1. 图像靠近边界。在初期运行代码时发现原图像过大，故而首先对其进行了裁剪，裁剪方式为取第一个目标所在的位置进行图像裁剪，裁剪后图像大小为1024\*768，然后对剩余的带标签目标进行判断，包含于该区域的目标则写进数据集中。在该过程中，由于一部分区域刚好位于裁剪后图像的边界处，同时（如第一条所述）每个带标签数据都包含了一块区域，故而会产生一定的误导，一部分位于边界的目标被写入了数据集中，另一部分则没有，对检测产生了一定的影响。

示例图像：

1. 像素亮度不统一。计划尝试使用图像预处理的操作试进行更正，但由于其他原因搁置了，在下文中说明。

示例图像：



# 3.31更新

由于原程序仅提供了判断某区域是坦克的概率，没有提供全数据集的误检漏检率，故而考虑自己写一个以进行统计。在代码构建的过程中发现了两个较为重要的发现。

1. 上文中提到在构建数据集的过程中进行了裁剪与带标签数据的判断，在此过程中会符合的数据写入到数据集中，但由于粗心，**反复写入的是第一个目标的位置信息。**该操作对数据集产生了**巨大影响**，修复后重新生成数据集，重新进行训练的过程中遇到了不应该发生的错误，推测应该是没有维护好代码导致之前的某次修改产生了错误并且没有还原到可运行的版本，目前正进行修复。
2. 在最初的设计中只想先运行一个二分类（坦克&背景）的问题，再进行多分类（1,2,3类坦克&背景），后来就忘记需要重新制作数据集进行分类的事情导致1类2类3类坦克归于一个标签中，同样产生了一定的影响。即在结果图中有一些小型坦克（推测是其中一类）没检测出来。

示例图：