

科研实训报告

目 录

第一章	引言.....	1
第二章	相关工作综述.....	1
第三章	数据介绍与处理.....	2
	3.1 数据特点.....	2
	3.2 数据初步处理.....	3
	3.3 数据处理优化.....	4
第四章	方法.....	4
	4.1 数据处理.....	4
	4.1.1 去雾.....	4
	4.1.2 数据增强.....	5
	4.2 卷积神经网络.....	6
	4.3 评估指标.....	6
第五章	结果与分析.....	6
	5.1 初步结果.....	7
	5.2 去雾结果.....	7
	5.3 数据增强结果.....	8
	5.4 分析.....	8
第六章	结论与展望.....	9
	6.1 展望.....	9
	6.1.1 数据.....	9
	6.1.2 模型.....	9
	6.1.3 指标.....	9
	6.2 结论.....	9
参考文献：		错误！未定义书签。

第一章 引言

卫星图像下目标物体的检测和识别是近年来物体检测的热门之一。卫星图像的检测结果可以很好用于很多应用如寻找建筑，追踪等。本文的任务就是对卫星图像下的猪场进行物体检测，寻求合适数据处理以及网络下训练出一个较好的模型，可以在新的地图中以较高预测率检测出猪场或者疑似猪场的位置。

第二章 相关工作综述

卫星图像相对于普通图像的物体检测有一定的特殊性。而且卫星图像的图像分辨率较大，需要进行处理。并且非相关专业人员标记的格式并不能直接使用，需要对标记进行处理。加上训练网络参数的调整，训练结果的评估方面也需要调整。所以本文的结构分为以下几个部分。

数据的介绍和处理。首先是对数据的特点的分析和介绍，然后是对数据的标记进行处理，使其符合 PASCAL VOC 数据的标记格式。然后是对数据和标签进行切割，切割成合适的图像大小，完成数据的初步处理。并根据初步的处理出现的速度慢的问题进行优化。

方法。数据初步处理之后，对数据进行针对性的处理，对图像存在的类似“雾”的现象进行去雾处理。并对处理后数据图像少的现象，使用数据增强来增加数据量。接着放入网络训练，并使用新的评估指标对训练的模型评估。

结果与分析。是对上诉数据处理以及网络参数调整的尝试的结果展示，并对结果进行分析。

展望与结论。对过程中出现的问题进行分析，提出下一步可能可以优化的步骤和方法并进行总结。

第三章 数据介绍与处理

3.1 数据特点

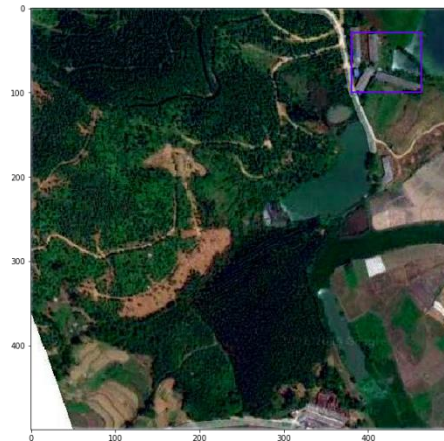
本次使用的数据集是由社科院方面采集的“猪场”数据，目标是通过对标注的猪场数据进行训练，并实现对新的地图的猪场的预测和目标检测。

数据集总共包括接近 200GB 的压缩包，包括从 3 月 3 日至 6 月 6 日期间标注的数据，总计十个日期标注的数据，除去标记重复和出错的日期，总计七个日期的数据。每个日期下的地区个数各有不同。每个地区的包含每个地图的大小的量级约为 25000*35000 像素，。标记文件为 shx 和 shp 文件，是卫星地图下常用的坐标表示文件。如下图，其中标注.tif 帮助理解，在处理时候无需使用。

Name
标注
原图.tif
标注.tif
黑白图.tif

压缩包的文件结构

猪场的大多呈现方形的白色猪棚状，部分小猪场的样式各不相同，呈现灰色或灰黑色或者其他特殊样式均有。给目标检测提供了不小的难度。



猪场的特征

3.2 数据处理

初步目标是通过标记文件识别并转换，得到每张地图中的 **Annotations** 信息，然后对处理后的原图和 **Annotations** 信息进行切割处理，处理成 $500*500$ 的图像大小，并将标记转换为常用的 **PASCAL VOC** 格式。但是实际处理的时候发现，数据标记出现问题，从 **shx** 和 **shp** 文件中获得的坐标信息是绝对坐标信息，也就是一个 $20000*30000$ 的地图的坐标信息却为几十万，通过观察发现这张地图是一张更大的地图的子图。但是无从得知这些子图在大地图中的相对位置，所以无法对标记进行处理。所以另寻方式获得标记。

发现每个地图中除了原图.tif 还有一个黑白图.tif 文件，这个文件中的标记区域的颜色是不同的，因此可以通过黑白图文件中对标记区域进行识别并生成标记文件进行训练。



切割的 $500*500$ 猪场和对应的黑白图

对此使用 **matlab**+区域生长算法对整张地图的标记区域进行识别，获得到的区域筛选掉只有几个像素的区域，剩下的区域取 **x** 和 **y** 坐标轴的最大值和最小值，生成 **PASCAL VOC** 格式的 **xml** 文件，再对 **xml** 和对应的图像进行切割成 $500*500$ 的大小，以待下一步处理使用。区域生长算法过程如下：

第一步：将图片转换为二值图

第二步：提取二值图中的轮廓信息

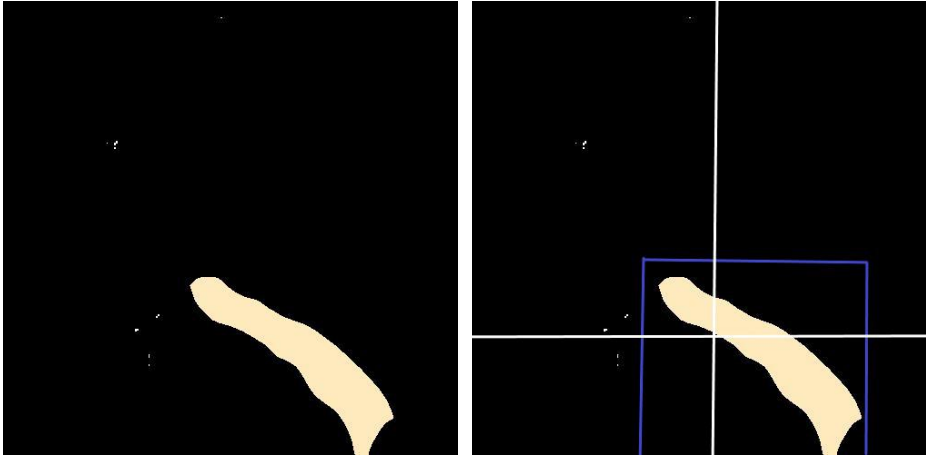
第三步：任取二值图中的一个点，进行区域生长。在该点的 $3*3$ 的矩形区域，检测这些点是否也是符合要求（在轮廓坐标信息中），如果是，则继续检查直到区域检测完毕。

第四步：保存上述区域的检测框信息，并从轮廓中删除该区域的点

第五步：重复三四步骤，知道轮廓中的点全部被检测

3.3 数据处理优化

上述方法已经可以取得猪场信息的信息了，但是有两个问题：发现切割的图像出现部分全白或者没有猪场的图片以及处理速度过慢。经过排查，原因是有二：①部分猪场的大小较大，且呈现长条状；②先识别再切割的对于细长状的物体，可能会使得不包含物体的区域被错误认为是包含物体的。如下图，可以看到切割后左下角是图像是没有猪场的（也有可能是空白）。



原图的一部分（左），应该切割的线（右）

因此采取一下优化策略：①更改顺序为先切割再识别猪场；②猪场的像素值在 200 至 250 之间，区别于黑白，所以放弃之前的转换为二值图的方法，直接取整个区域；③通过缩小检测区域来提高速度，从原来的检测 3*3 区域改为检测上下左右区域；④放弃原来的重复判断方法：ismember 方法。修改新建一个 flag 矩阵存储来判断。

通过上述的方式，大幅提高了识别物体的速度，从原来的一张地图要识别 20 分钟左右到同样的地图只需要 2-3 分钟。通过上述的处理后，原来的 200GB 的数据获得的 500*500 图像数量如下表。

日期	0606	0531	0524	0517	0510	0426	0419
地图数量	6	1	5	3	4	13	1
图像数量	342	180	245	86	213	212	30

总计 33 张地图和 1308 张 500*500 图像。将这些数据分为 trainval 和 test 两部分，其中 trainval 集包含 0606、0510、0426、0419 四个日期共计 24 个大地图 797 张图像。test 集包含 0531、0524、0517 三个日期共计 7 张大地图 511 张图像。

第四章 方法

4.1 数据处理

通过上述的处理已经完成对图像的提取和 Annotations 和 JPEGImages 的 PASCAL VOC 格式的数据生成。接着就是根据数据的特点进行特殊化处理的过程，对此采取了两个方法：去雾和数据增强。

4.1.1 去雾

通过观察原图片的特征，发现部分图像存在疑似“云雾”的白色模糊的现象，如下图的对比。因此对数据进行去雾处理，并对比去雾前后在测试集的表现。通过未去雾--未去雾，去雾--未去雾，未去雾--去雾，去雾--去雾四种训练和测试集合，发现使用未去雾--去雾和去雾--去雾都可以对评估指标有一定的提升（详见第五章分析）。其中去雾--去雾组合符合预

料的更胜一筹。



部分图像正常（左），部分图像有“雾”（右）

使用的去雾模型参考 Pytorch 中文网发布于 2018 年 10 月 9 日的 AOD-Net 图片去雾[1]，该模型可以实现对单张图片的去雾，并且模型非常小，训练速度快，预测速度快的特点，是对图片进行简单去雾的好选择。该模型还可以通过嵌入到其他神经网络中一起训练以寻找到最佳的去雾效果，但是由于时间关系没有进行这方面尝试。

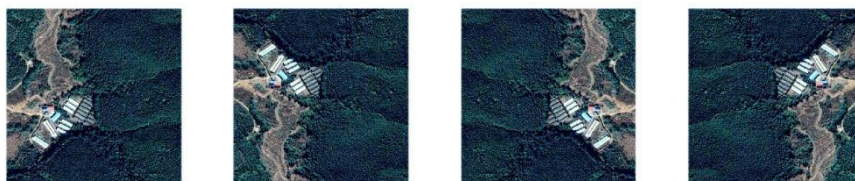


未去雾（左），去雾（右）

4.1.2 数据增强

因为原有的数据只有 1.3k 的图像，训练数据甚至只有接近 0.8k 张，而且猪场的也各有特点，希望通过数据增强来提升模型预测的准确度和召回率，以及提升模型的泛化程度。

此处使用常用的数据增强方式，翻转、旋转、倾斜的组合，并且根据地图的特点，加入了上下翻转。本次采用两种数据增强组合，第一种是对数据进行随机增强，1/2 的概率左右翻转，1/2 的概率上下翻转，随机旋转 -15° 至 15° ，随机倾斜 -10° 至 10° ，，一张图片输出 12 张增强图片。第二种是只采用翻转，一张图片通过左右上下翻转，加上原图一共输出 4 张图片。



第二种处理

4.2 卷积神经网络 SSD

本次使用的是经典的目标检测的卷积神经网络 SSD (Single Shot MulitiBox Detector)。SSD 采用的 one-stage 的方式, 使用多框测试的方式。此处使用的是 amdegroot 版本 pytorch 实现的 SSD[2]。在源代码的基础上, 阅读并修改, 使之可以训练自己的数据集。首先是修改类的数量为 1, 类名为猪场, 并将所有猪场的 difficult 值设置为 0。并修复源代码的存在的部分不适配的 bug。其他修改完善可见 github[3]。

4.3 评估指标

4.3.1 mAP

测试过程使用的是 PASCAL VOC 数据集中常用的指标 mAP 值, 具体的 mAP 值的计算可以参考链接[4], 使用 IoU 阈值为 0.5。但是不使用 VOC2007 的 detect metric 对各个类进行平衡, 因为只有一个类, 不需要这个操作。

4.3.2 其他指标

因为本次实验的要求是对猪场的检测有高的召回率 (recall) 和精确率 (precision), 所以使用 recall 和 precision 值进行指标评价。

首先是对预测框的类型判别, 包含 TP (true positive)、FP (false positive)、TN (true negative)、FN (false negative)。

1) True positives(TP): 被正确地划分为正例的个数, 即实际 为正例且被分类器划分为正例的实例数 (样本数);

2) False positives(FP): 被错误地划分为正例的个数, 即实际为负例但被分类器划分为正例的实例数;

3) False negatives(FN):被错误地划分为负例的个数, 即实际为正例但被分类器划分为负例的实例数;

4) True negatives(TN): 被正确地划分为负例的个数, 即实际为负例且被分类器划分为负例的实例数。

实 际 类 别	预测类别			
		Yes	No	总计
	Yes	TP	FN	P (实际为 Yes)
	No	FP	TN	N (实际为No)
	总计	P' (被分为Yes)	N' (被分为 No)	P+N

由此计算召回率和精确率:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

第五章 结果与分析

5.1 初步结果

根据上述的部分数据（其他数据尚未处理）处理并放入网络中训练，batch size 为 32，lr 为 $1e-4$ ，训练至 20000 个 iter 使接近拟合并在 test 集和没加入到训练的地图中取得相对好的结果。

iter	lr	Trainval AP	Test AP	未训练地图 AP
17500	$1e-4$	0.803	0.365	0.266
20000	$1e-4$	0.808	0.366	0.293
22500	$1e-4$	0.821	0.366	0.270

在同样的 batch size，修改 lr 为 0.0005，训练至 10000 个 iter 降低 lr 为 $1e-4$ 并训练至 11000 个 iter，获得结果

iter	lr	Trainval AP	Test AP	未训练地图 AP
10000	0.0005	/	0.361	/
11000	$1e-4$	0.868	0.370	0.293

因为训练初步结果的时还没有使用新的指标，所以使用 PASCAL VOC 数据集的指标 mAP 值进行分析。

5.2 去雾结果

对所有数据进行处理，并去雾，然后以 batch size 为 32 进行训练，并在合适的 iter 时降低 lr，获得相对最佳的数据结果。

iter	lr	Trainval AP	Test AP	未训练地图 AP
5000	0.0005	0.481	0.313	/
11000	$1e-4$	0.798	0.382	0.307
11500	$1e-5$	0.802	0.402	0.322

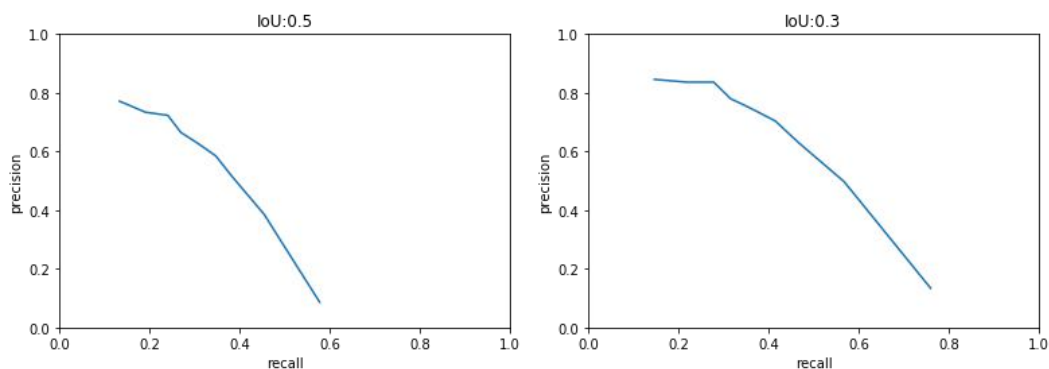
这是目前为止最好的 mAP 结果，并对该数据进行新的指标测试，recall 和 precision 值的检测，并画出 recall 和 precision 的对应折线图，如下。

IoU: 0.5

score	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Recall	0.58	0.45	0.38	0.35	0.31	0.27	0.24	0.19	0.13
Prec	0.09	0.38	0.51	0.59	0.63	0.67	0.72	0.73	0.77

IoU: 0.3

score	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Recall	0.76	0.57	0.47	0.41	0.36	0.32	0.28	0.22	0.15
Prec	0.13	0.50	0.63	0.71	0.75	0.78	0.83	0.84	0.85



然后使用未去雾和去雾的表现较好的进行交叉检验。结果如下：

方式	测试集 AP
未去雾--未去雾	0.366
未去雾--去雾	0.327
去雾--未去雾	0.325
去雾--去雾	0.402

5.3 数据增强结果

数据增强的思路很正确，但是在使用数据增强的时候出现问题，在存储新生成的图像的时候，没有去掉周围的白色填充，导致新生成的图像和标记没有对应，所以训练结果一直欠拟合。训练的 mAP 在 0.15 左右就不在提升，与预期完全相反。但是由于时间关系没有方法继续尝试。

5.4 分析

根据上述的结果可以看到，在原来的训练结果下，可以取得相对可以的结果。并且使用去雾之后，也对效果有所提升。并且检测出去雾和未去雾的结果是不可交叉测试的，结果之后更差。所以最佳的是使用去雾训练并用来预测去雾的图像。

对于数据增强，因为代码错误导致没有能够看到应该的结果，可以在接下来继续尝试并获得更好的结果。

因为是实际案例，并且观察给出的黑白图和实际猪场的位置，发现黑白图的猪场的大小和实际的大小有所差别，有的大有的小。并且也发现部分和猪场很相似的民房，并被预测成猪场。也就是说标记数据的实际情况可能会使得训练效果变差以及测试效果出现偏差。所以使用 PASCAL VOC 的 IoU 限制 0.5 过高，IoU 为 0.3 较为合适。

观察 recall 和 precision 的折线图可以看出，总体效果提升的空间还很大，并且 recall 最高也只有 0.5 以上一点。经过观察发现切割的缘故，导致部分猪场被切割之后，产生了部分被切割的猪场，这类图片中的猪场较小，加上一些猪场本身就很很小，很难被预测到，也是造成 recall 率很低的缘故。

第六章 结论与展望

6.1 展望

根据上述的分析，优化和展望可以分为数据、模型和指标三方面

6.1.1 数据

- ①首要的是可以对去雾后的数据进行正确的数据增强再进行训练；
- ②也可以对原来的训练地图进行处理，也就是说原来是切割后通过黑白图识别猪场，可以通过识别猪场后，回到原图中，对该区域进行不同位置的框选，可以得到更多的真实数据；
- ③可以解决如何处理大物体被切割到不同图像的问题；
- ④可以扩大筛选的严格度，也就是生成标记阶段，不计入小物体，提高猪场门槛。这样子应该会提高准确率，但是相应的可能会降低预测小猪场的能力；
- ⑤可以采用特殊的方式对小物体进行数据增强[5]
- ⑥如果觉得一开始图像像素太低，可以重新处理原始数据。因为空间文件，在切割完原图后，存储的时候没有存储为 jpg 格式并进行压缩，所以分辨率有所下降，可以从原图（每一个地图 3GB 左右的 tif 文件）重新分割并保存。

6.1.2 模型

- ①使用的 SSD 模型的局限性，SSD 模型使用候选框大小与猪场的大小并不是很适应，可以通过修改 SSD 的候选框参数以及数量，并通过优化参数来获得更好的预测效果。

②也可以使用其他神经网络模型如 Mask RCNN 等来进行训练。

6.1.3 指标

如分析章节所述，PASCAL VOC 的 IoU 指标相对严格，可以减低 IoU 来查看结果和预测目标。

6.2 结论

虽然实验过程遇到很多困难，但是慢慢克服困难，并且通过实验和分析已经对数据和模型的特征有个一定了解，也取得相对较好的训练效果，但是这个结果还有很大的提升空间，相信通过改进是可以让预测结果达到预想的目标的。

参考链接

1. <https://www.pytorchtutorial.com/pytorch-image-dehazing/>
2. <https://github.com/amdegroot/ssd.pytorch>
3. <https://github.com/cxh666>
4. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173
5. <https://github.com/cxh666/dataset-augmentation>
6. <https://blog.csdn.net/a819825294/article/details/51239686>
7. <https://blog.csdn.net/jacke121/article/details/79886917>
8. <https://blog.csdn.net/shenziheng1/article/details/50878911>
9. https://blog.csdn.net/qq_35608277/article/details/78598016
10. <https://blog.csdn.net/u010368556/article/details/70175380>
11. <https://unordered.org/timelines/59d5af2422c01000>
12. <https://blog.csdn.net/renyuanxingxing/article/details/78835275>