



湖州师范学院

2023 届毕业设计(论文)

课 题 名 称: 基于遥感图像的森林火灾检测技术研究

课 题 名 称 (英文): Research on forest fire

detection technology based on remote sensing image

学 生 姓 名: 许多华 学 号: 2019162226

专 业 名 称: 计算机科学与技术

指 导 教 师: 高强 职 称: 讲师

所 在 学 院: 信息工程学院

完 成 日 期: 2023 年 3 月 20 日

基于遥感图像的森林火灾检测技术研究

摘要：传统的森林火灾检测主要依靠人工巡视，这种方法不仅消耗大量的人力和物力，而且效率低下。随着深度学习技术的发展，将其应用到森林火灾检测成为可能。YOLO 检测算法具有检测速度快、精度高等优点被广泛应用。本课题研究基于 YOLOv7算法的森林火灾检测技术，采用了 Pytorch 神经网络框架构建森林火灾检测系统，并使用 labelImg 工具对搜集到的森林火灾数据集进行标注。数据集包含500张发生森林火灾的图像以及300张没有发生火灾的图像，按照9:1的比例划分训练数据集和测试数据集。统计不同 IOU 阈值下检测的平均精度（mean average precision, mAP）。实验表明，在 IOU 设为0.5时，其检测精度的平均值可以达到0.769，在 IOU 的取值位于[0.5, 0.95]时，检测精度的平均值可以可以达到0.733。实验表明，YOLOv7模型对森林火灾具有较高的检测精度。

YOLOv7模型先将图像划分为多个网格，每个网格预测落在该网格物体的位置和类别，采用卷积层和残差结构来提取特征，使用检测头预测物体的位置和类别，再使用非极大抑制算法(NMS)筛选出最优的预测结果，相比于之前的 YOLO 版本，YOLOv7使用了更合理的模型结构和训练方法，在检测速率不变的情况下，检测精度有了很大的提升。

关键词：YOLOv7，森林火灾，目标检测，深度学习

Research on forest fire detection technology based on Remote sensing image

Abstract: Traditional forest fire detection mainly relies on manual inspection, which not only consumes a lot of manpower and material resources, but also has low efficiency. With the development of deep learning technology, it is possible to apply it to forest fire detection. YOLO detection algorithm has the advantages of fast detection speed and high precision and is widely used. In this paper, the forest fire detection technology based on YOLOv7 algorithm is studied, the Pytorch neural network framework is adopted to build the forest fire detection system, and the lambelling tool is used to label the collected forest fire data sets. The data set consisted of 500 images of forest fires and 300 images of no fires. The training data set and the test data set were divided in a 9:1 ratio. The mean average precision (mAP) values of different IOU thresholds were collected. Experiments show that when IOU is set at 0.5, the detection accuracy can reach 0.769; when IOU is set at [0.5, 0.95], the average detection accuracy can reach 0.733. The experiment shows that the YOLOv7 model has high detection accuracy for forest fires.

The YOLOv7 model first divides the image into multiple grids, and each grid predicts the position and category of objects in the grid. Convolution layer and residual structure are used to extract features, detection heads are used to predict the position and category of objects, and then non-maximum suppression algorithm (NMS) is used to screen out the best prediction results. Compared with the previous YOLO version, YOLOv7 uses a more reasonable model structure and training method, and the detection accuracy is greatly improved while the detection rate remains the same.

Keywords: YOLOv7, Forest fires, target detection, Deep learning

目 录

第一章 绪 论	1
1.1 研究背景及意义	1
1.2 研究现状及发展趋势	1
1.2.1 传统的森林火灾检测方法	1
1.2.2 基于深度学习的森林火灾检测算法	2
1.3 课题研究的内容与结构	2
第二章 YOLOv7 算法	4
2.1 YOLO 算法介绍	4
2.1.1 人工神经网络	4
2.1.2 Pytorch 深度学习框架	5
2.1.3 YOLO 算法介绍	5
2.1.4 YOLOv3 的改进点	6
2.2 YOLOv7 的基本原理	7
2.2.1 引言	7
2.2.2 ResNet 残差结构	8
2.2.3 RepVGG 结构	8
2.2.4 BN 操作与卷积操作的合并	9
2.2.5 高效的聚合网络	10
2.3 YOLOv7 检测流程	11
第三章 仿真实验	12
3.1 样本数据集的选取与构建	12
3.1.1 数据集选取	12
3.1.2 数据集的预处理	12
3.1.3 实验环境介绍	13
3.2 YOLOv7 森林火灾识别检测模型	14
3.2.1 实验步骤	14
3.2.2 实验训练结果	15
3.2.3 森林火灾图像预测结果	18
3.2.4 结果分析	18
3.3 本章小结	20
第四章 总结与展望	21
参考文献	22
致谢	23

第一章 绪 论

1.1 研究背景及意义

保护森林对人类的生产实践活动具有重大的意义,保护森林是我们义不容辞的责任,然而森林火灾会对森林造成巨大的破坏,我国幅员辽阔,森林资源和植物种类极为丰富,所以要更重视对森林火灾的检测。森林火灾对生态系统的破坏是最具有灾害性的,会加剧土地荒漠化和水土流失。森林火灾的突发性强,破坏性巨大,是处置救助较为困难的自然灾害。森林火灾发生时往往极易蔓延,难以控制。那么在第一时间发现森林火灾并及时处理显得尤为重要,在林地内,往往人烟稀少,所以在森林内布置大量传感器的难度巨大,当前,判定森林火灾的依据主要还是通过对遥感图像进行检测,使用小型飞行器携带高清摄像机传回实时遥感图像。YOLO 算法模型已经能够提供基础的图像识别,如何提高检测的精度和准确率是当前面临的主要难题。

传统的人工巡视^[1]、瞭望台检测、卫星遥感对森林火灾的检测^[2]都存在较大的问题,随着深度学习的不断深入,卷积神经网络(RCNN)^[3]也有了很大的更新,对于图像特征的提取已经不需要手动提取,卷积神经网络模型可以自动完成特征提取。检测的准确率进一步提升。YOLOv7 算法^[4]克服了使用 YOLOv5s 算法检测森林火灾^[5]收敛慢的缺点,在保持识别准确率的基础上提高运算速度,这一提升对森林火灾检测模型有重大意义,意味着可以在一定时间内检测更大的面积。基于上文,本实验采用 YOLOv7 算法模型实现对森林火灾检测,分析 YOLO 算法在应用于森林火灾检测系统的优点和不足之处,为将来的森林火灾检测研究提供数据支撑。

1.2 研究现状及发展趋势

传统的森林火灾检测方法如人工巡检、瞭望台检测、卫星遥感检测都有一些难以解决的问题,在当前使用最多、最具有前景的还是基于深度学习的目标检测算法,深度学习会对图像有各个角度的数据挖掘学习、对原有数据扩展,丰富学习数据进一步学习。正因为 YOLOv7 有推理速度快,检测结果准确等特点,深度学习算法在对森林火灾检测中有了迅速的普及,并且取得的重大发展。

1.2.1 传统的森林火灾检测方法

早期人们对森林火灾的检测手段是人工巡视,由于这种做法大概率每天只能到一个地方巡视一次,并不能及时的发现森林火灾,还会增加很多的人力开销。瞭望塔较人工巡视有了一定的改进,它覆盖面积大,效果有所提升。但是观测效果易受到地理环境的限制,有死角和空白,对地面上的火情观测不到,而且观测人员容易受到雷电,蚊虫的威胁。

近几年来,机器学习和深度学习不断的发展,目标检测算法在森林火灾检测领域的应用也有了重大发展,祖鑫萍等提出了基于无人机图像和改进 YOLOv3-SPP 算法的森林火灾烟雾识别^[6]方法,经过实验得出检测到森林烟雾的精确率为 91.07%。qi-xin ZHANG 提出了基于 Faster R-CNN 算法用于野外森林火灾烟雾检测^[7],算法自动进行图像特征提取。避免了复杂的人工提取过程。Zhang Yongtuo^[8]将金字塔注意力机制引入森林火灾检测模型当中,通过对 PyAtten 的广泛评估证明了其在对各种类型的火灾和烟雾检测方面的效率和有效性,即使需要检测的部分占整张图像的比例很小。PyrAtten-ResNet-50 的平均准确率达到了 97.5%。Fast R-CNN 网络^[9]仍需要从候选区域中提取特征矩阵,并且要写入磁盘中保

存起来, 会加重内存的负担, 同时造成更大的时间开销。而且在测试阶段需要重新提取特征矩阵, 导致网络模型的检测速度较慢, 对森林火灾检测这种时效性很高的任务中, Fast R-CNN 算法的表现是不够出色的。

1.2.2 基于深度学习的森林火灾检测算法

YOLOv7(You Only Look Once)算法对输入的图片、视频可以同时进行识别和检测, 由于检测模型采用单阶段检测^[12], 对图像只需要一次前向传递即可同时预测图像中的物体类别和位置信息, 所以又称为: “你只需看一次” 算法。想比于过去的卷积神经网络, 其检测速度提高, 检测的精确度也进一步的提高, 在检测森林火灾领域内有很强的应用前景。秦常帅^[10]提出一种改进 YOLOv4 的算法, 使用了注意力机制来提升检测精度。李孝钊在 YOLOx 的基础上提出了 FM-YOLO 目标检测算法^[11], 通过对 backbone 的改进来提升精度。在基于深度学习的森林火灾检测算法中, 通常用 CNN 网络来去除图像中雾等干扰元素, 例如: AOD-Net^[13], 多尺度卷积神经网络和密集链接的金字塔除水网络 (DCPDN)^[14]。这些操作可以修正图像中烟雾的状态, 提高对森林火灾检测的精度。YOLO 对对象的检测是单位级别的, 主要由 backbone、neck 和 head 组成。backbone 用来自动提取图像的特征, neck 实现对特征的卷积和筛选, 头部(head)输出预测结果。如图 1-1 所示。

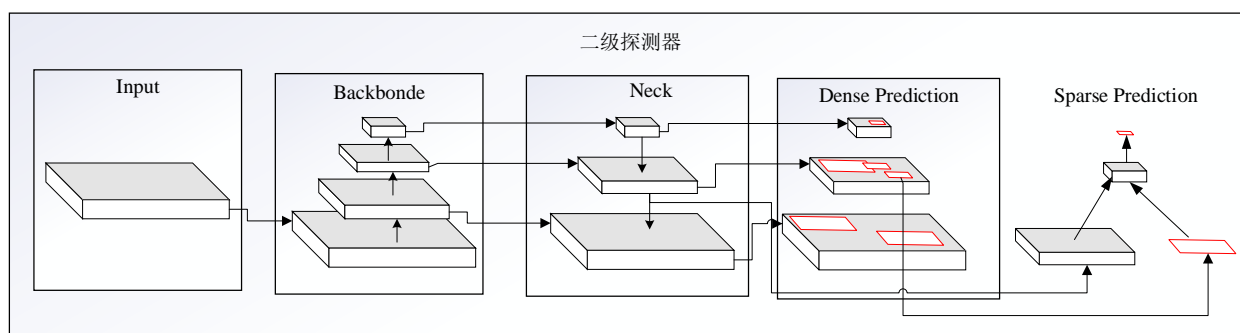


图 1-1 YOLO 模型

相比于 YOLOv5 算法, YOLOv7 算法的提出者做了一系列更新, 比如使用了更快更强的网络架构。对集成方法、检测方法都有明显的改善, 同时作者对模型架构做了调整, 比如: 复合缩放、E-ELAN (高效聚合层网络)、还有一个带有重参数化的免费礼包。在架构在没有增加硬件资源的同时提升了检测速率和检测精度, 其检测能力显著高于其他公知的检测模型。

YOLOv7 采用拓展的高效聚合层网络 (EELAN), 通过对图像的拓展、打乱重组, 再进行合并基本的数据得到输出。这种结构可以对长度不一样的网络进行控制, 从而达到有效的收敛。这种架构同样允许原始梯度不变的情况下还能保持有效的学习。在过去的深度学习框架中, 把内存和 GPU 的使用情况考虑进来, 虽然识别的准确率较高, 但是需要的 GPU 的计算资源更多, 而且在损失函数的反向传播中会导致数据丢失。YOLOv7 采用 EELAN 网络可以有效拓展网络的深度和提升识别的准确度。更利于提高对模型森林火灾的检测精度。

1.3 课题研究的内容与结构

本实验基于深度学习中的 YOLOv7 算法, 探讨了其在森林火灾目标检测方面的应用。采用 PyTorch 深度学习包构建了基于 YOLOv7 算法的检测模型和 mAP 精度图像, 并通过实验验证了该算法在森林

火灾目标检测任务中的性能表现。最终比较了平均检测精度和召回率等指标，分析了算法的优缺点和适用性。具体实验内容如图 1-2 所示。

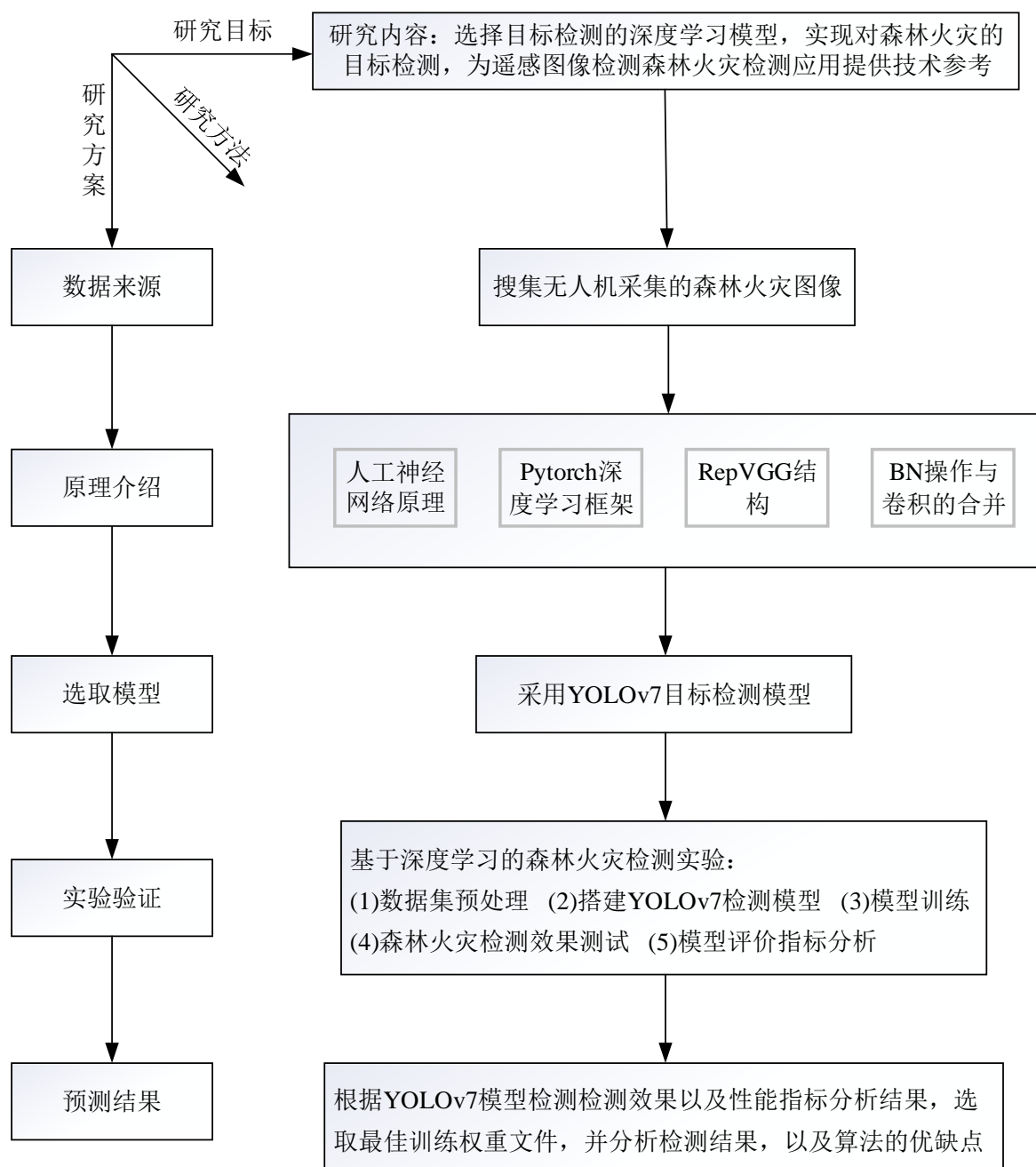


图 1-2 研究方案

第二章 YOLOv7 算法

2.1 YOLO 算法介绍

2.1.1 人工神经网络

人工神经网络是受生物神经网络的启发，从人脑的组织和结构中抽象出来的。在 21 世纪得到了迅速的发展，像生活中对垃圾短信的识别、病人的身体健康程度、日常看到某个物体，识别一下它们的种类等。神经网络或多或少的渗透到我们生活的方方面面。

神经网络的学习分为有监督式学习和无监督式的学习，深度学习训练时用到的是有监督式学习的神经网络^[15]。神经网络有 3 层结构：输入层、隐含层、输出层。每层由多个神经元构成，神经元结构如图 2-1 所示：

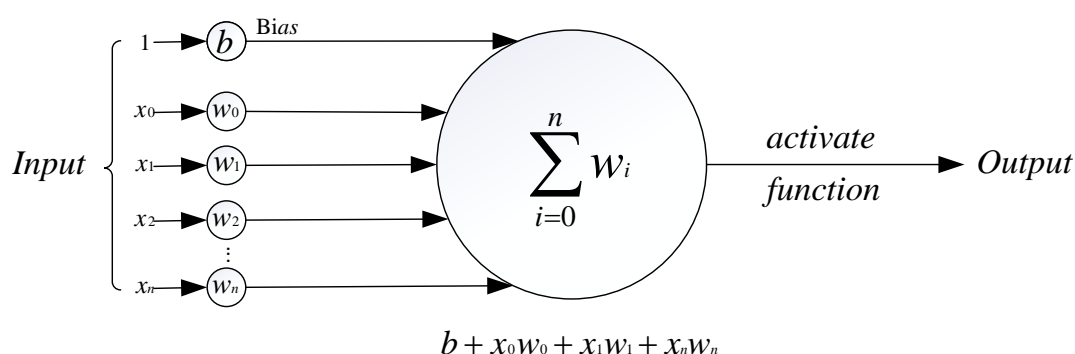


图 2-1 人工神经网络

每个神经元可以有 n 个输入，在输入到神经元轴突前都要乘以相应的权重(w_i)， $Bias$ 作为偏差，可以通过调节 $Bias$ 的值来上下移动函数。对上述参数求和将会得到该权重向量对应的得分情况 y ，假设我们的实验做的是二分类，即分为 a, b 两类，那就可以将 y 映射到 $(0, 1)$ 分布上面。映射函数采用 $\text{sigmoid}()$ 函数，如图 2-2，该函数可以把实数范围内的输入，转换为在 $[0, 1]$ 上的输出。如图。那预测结果为：

$$\text{Predict}|a = \text{Sigmoid}(y) \quad \text{Predict}|b = 1 - \text{Sigmoid}(y) \quad (1)$$

激活函数的种类有很多，比如 ReLU、tanh、Sigmoid、阈值激活函数等，它们适用的领域也不相同，在上述例子中的要求激活函数处处可微分，所以选择 Sigmoid 函数，但是该函数存在的缺点是提供了较大的负输入，可能会导致模型在训练过程中无法前进。

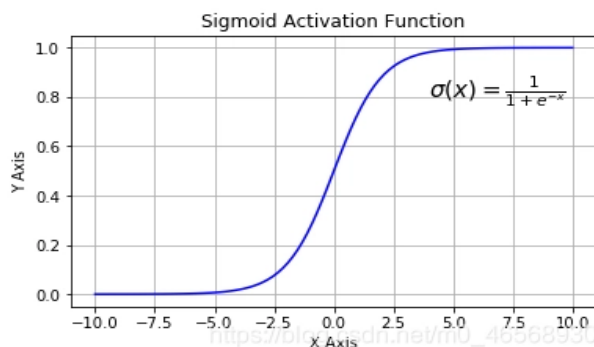


图 2-2 sigmoid 函数

2.1.2 Pytorch 深度学习框架

Pytorch 是基于 python 语言开发的机器学习库^[16]，由 Torch 团队开发维护。市场上较为流行的还有 tensorflow 框架，同样是深度学习框架。Pytorch 在调用 GPU 加速的同时还可以支持动态神经网络。随后 Tensorflow 推出 2.0 版本也加入了动态神经网络。二者的区别在于：Pytorch 在小规模的项目上表现优秀并且结构简洁，易于配置，并且 Pytorch 的 API 设计清晰简洁，允许 API 能不断更新并且与深度学习领域内的最新趋势保持同步，使用方便。而 tensorflow 更适合在工程级的、跨平台上面的大项目上操作，所以本研究基于 Pytorch 展开。

Pytorch 因为其 GPU 加速的张量计算和自动求导的功能被人们青睐有加。Python 在近几年取得了很好的发展，其生态逐渐完善，Pytorch 允许开发者使用 Python 的生态和库。让熟悉 Python 程序设计语言的用户易于使用，在基于 Pytorch 的开发中，Numpy 矩阵计算与 tensor 张量计算过程相似。可以让更多的开发者参与到深度学习的领域中来。

简洁的框架设计，虽然牺牲了一定的性能，但是复杂的设计会导致结构和边界的不清晰，从长远来看依然会导致性能下降。而且简洁的结构更容易后期维护，能及时的发现问题并作出快速响应。组件通过接口来定义，定义和使用组件的复杂性被简单的初始化函数调用，从而能将注意力集中在核心神经网络算法中，更利于初学者学习神经网络。

2.1.3 YOLO 算法介绍

YOLO 系列算法^[17]最开始由 Joseph Redmon 提出，用来做目标检测。目标检测是一种计算机视觉技术，它可以被用来做图像的分类、检测、定位。也可以做视频中某个对象的识别与跟踪。与 Fast-RCNN 网络不同，Fast-RCNN 模型将检测问题分为 2 类求解：分类问题和回归问题。虽然检测的精度较高，但是需要较好的硬件资源，加大了时间开销。YOLO 系列算法很好的解决了这个问题，将检测任务统一成一个回归问题，如图 2-3，在检测速度方面有很大的提升。

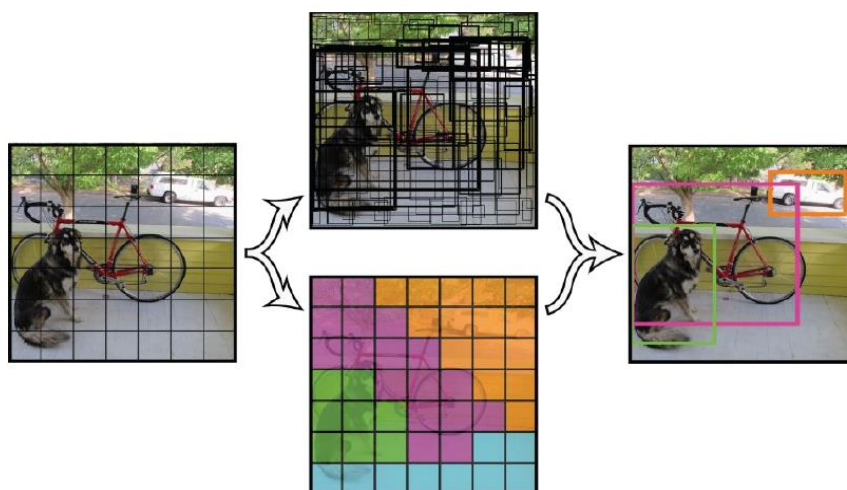


图 2-3 YOLO 算法示意图

YOLO 模型开始将图像分成 $S \times S$ 的格子，每个网格只关心预测中心落在这个盒子中的物体，每个网格中将会预测多个边界框 (bounding boxes)，同时包含这些边界框的得分，以及是各个物体的概率，置信度 (confidence score) 反应了边界框里是否有这个物体、这个物体位置的准确度定义：

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * IOU_{pred}^{truth} = \Pr\{\text{Class}_i\} * IOU_{pred}^{truth} \quad (2)$$

每个边界框有 5 个参数，边界框的参数有(x,y,w,h,confidence)，其中有(x,y)反应了坐标表示盒子的中心相对于网格单元格的边界。(w,h)反应了相对于整个图像，边界框的宽和高，confidence 是预测位置 and 实际位置的 IOU，IOU 对图像特征的反应能力是超过 UNION 的。这个分数反应了该分类出现在网格中的概率，也给出了预测框对于物体的适应情况。在 Pascal VOC 数据集上面对 YOLO 算法模型评估中，通常使用 7*7 的网格，每个网格预测 2 个边界框，Pascal VOC 数据集中有 20 个类，所以 YOLOv1 的预测模型要输出 7*7*30 个张量，每张图片使用 98 个边界框，分割结果如图 2-4 所示。

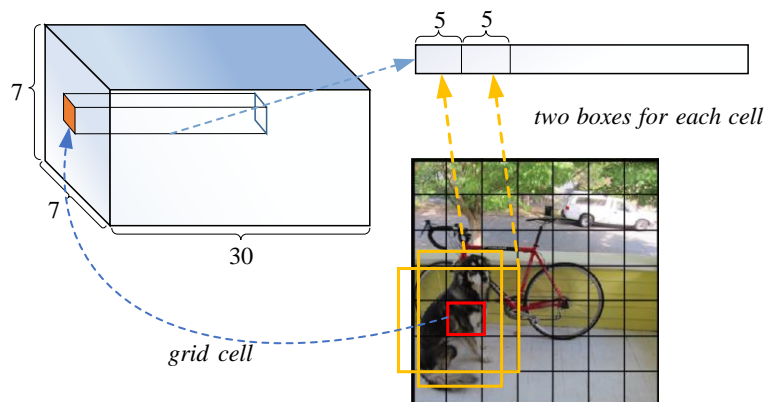


图 2-4 YOLO 分割示意图

YOLOv1 的卷积层和完全连接层分别是 24 和 2，网络输入的图片是 448*448。相比于 R-CNN 的参数设置更合理。YOLOv1 模型每秒对于图像的处理速度达到 45 帧每秒。是 Fast R-CNN^[18]模型处理速度的 10 倍，Fast YOLO^[19]模型在检测速度的提升更迅猛，已经达到了 155 帧每秒。可以满足实时的要求，但是定位精度较低，对小目标的检测能力有限。YOLOv1 中每个网格只能检测 1 个对象，每张图片最多有 49 个网格，所以每张图片中最多能检测出 49 个对象，当待检测的图片有较多的重叠对象，或者对象分布较为密集时，检测的精度会大大降低，其识别的种类也较少。

2.1.4 YOLOv3 的改进点

2018 年 YOLOv3 的提出对目标检测的各方面检测能力均有所提升，在保持推理性能的同时还能保持检测网络的精度。池化和我们的眼睛有类似之处，输入的特征数据都是压缩的，既可以减少运算的复杂度，又可以压缩图像特征。YOLOv3 在池化层使用尺寸不一样的内核，将它们连接起来融合这些特性。对图像的通道数降低，也就是降低图像的维度，在更高层次上反应图像的特征。YOLOv3 对设备的性能要求比较高。

YOLOv3 主要通过增加网络深度、优化非极大值抑制、引入多尺度检测和锚框、利用 FPN 结构以及优化训练策略等方式，进一步提高了目标检测的准确性和实时性。其中 Darknet-53 提取图像特征更加准确，采用多尺度检测技术和锚框思想可适应不同目标尺寸，优化 NMS 算法和使用 FPN 结构可提高小目标检测准确率，同时采用数据增强和跨领域预训练等训练策略可以提高模型的泛化能力和鲁棒性。这些改进使得 YOLOv3 在实际应用中具有更好的表现。

2.2 YOLOv7 的基本原理

2.2.1 引言

YOLOv7 是 YOLO 系列目标检测模型中最先进的版本，其网络结构如图 2-5 所示，在检测的效率和识别的准确度上做了很好的平衡。首次提出了扩展的高效远程注意网络(E-ELAN)、对缩放模型进行调整，使用串联结构调整策略、还有卷积重新参数化。YOLOv7 模型的整体结构和 YOLOv3 类似。YOLOv7 模型分别由输入、主干（backbone）、首部、预测四部分组成。将图像输入到网络模型中需要对图像进行 reset 处理，将图像的大小调整一致，以符合主干网络的输入大小要求。主干特征提取网络的组成部分有 Bconv 卷积层、E-ELAN 高效处理拓展层和 MPConv 卷积层构成的。其中 BConv 由卷积层、归一化(BN)层和激活函数，完成对图像特征的提取。首部模块采用路径聚合特征金字塔网络(PAFPN) 结构。创造性的引入新的路径，即从底部向高层传导的路径，让图片的底层信息更容易被网络捕获。从而实现了不同特征层的组合。预测模块采用了 RepVGG 模块(Rep)，采用 3*3 的卷积核，用来预测结果的置信度。相比 Reset 的多分支堆叠结构，Rep 使用多分支融合结构，在保持检测准确度的情况下，有效降低占用的硬件资源。由于森森火灾检测对准确度和实时性的要求都非常高，所以本实验采用 YOLOv7 作为检测模型。

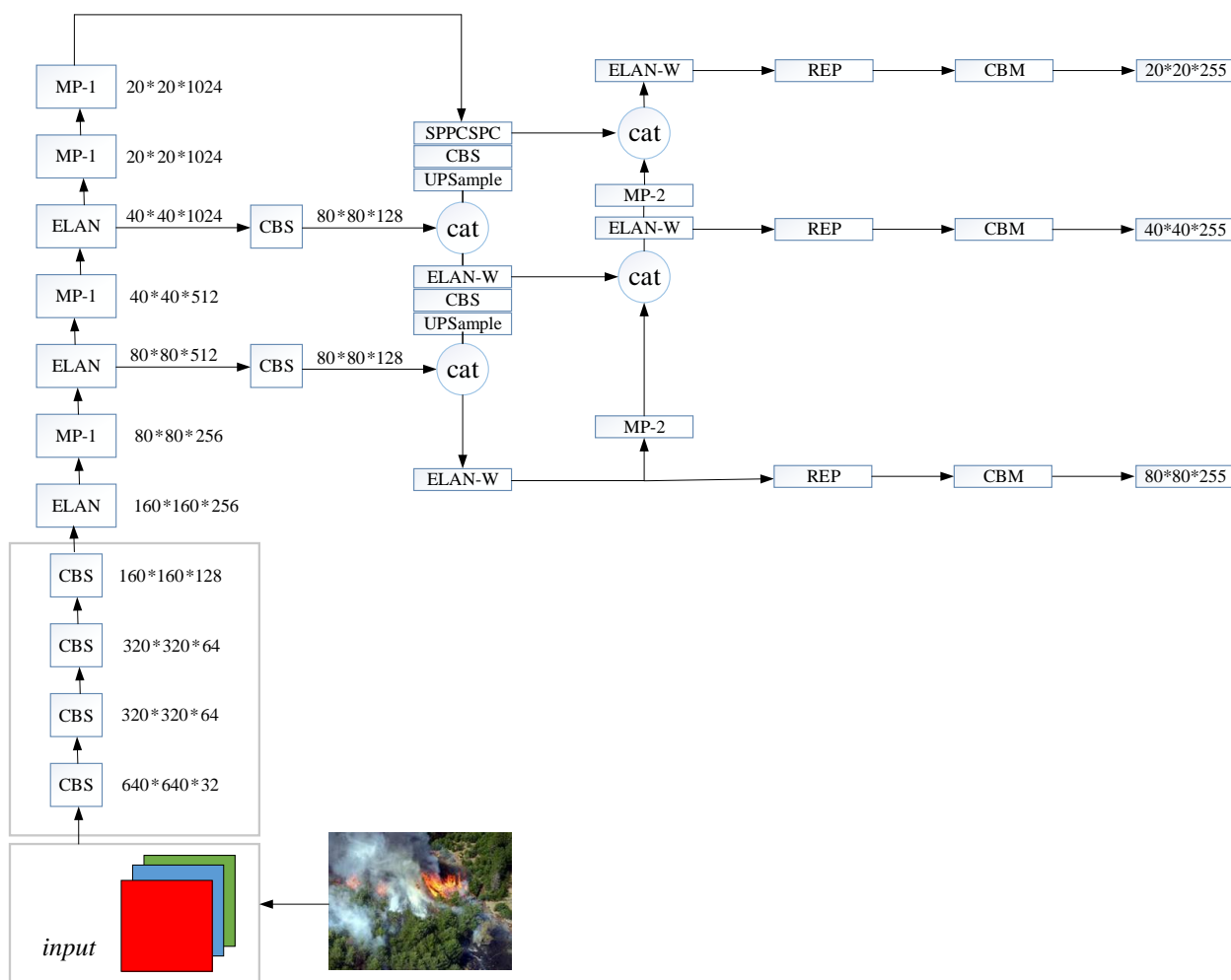


图 2-5 YOLOv7 示意图

2.2.2 ResNet 残差结构

ResNet 网络（如图 2-6）在 IagNet 上取得了不错的成绩。在以往的经验中，随着机器学习深度的增加，训练出来的模型应该更好，但事实并不是这样的。



图 2-6 ResNet 网络

当深度提升到一定程度后，网络模型的精确率趋向于饱和，不再提升。当深度再次增加，网络模型的检测精度将会下降，当网络的深度逐渐加深时，梯度会逐渐下降，甚至是消失，会导致准确率下降。ResNet 通过残差学习解决了网络训练的退化问题。ResNet 网络结构最大的特点是加入了残差结构，在卷积层数不变的情况下还可以提升模型的精确度。

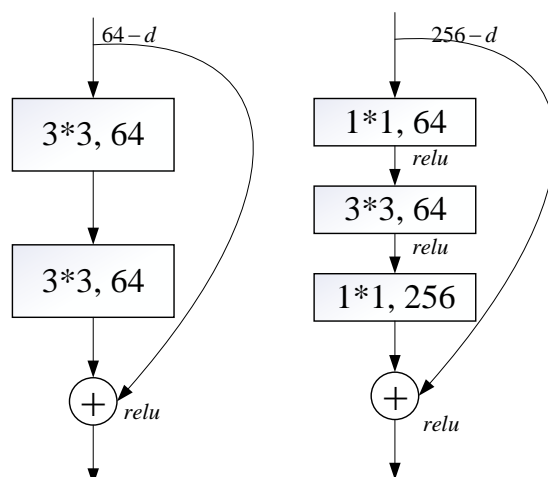


图 2-7 ResNet 中的残差结构

ResNet 在残差结构（如图 2-7）中给出了一些分支推荐，推荐使用 1×1 和 3×3 结构的分支，再对这些分支求和，这样做的好处是模型训练时至少不会发生倒退，但是需要训练中将模型中所有变量都保存下来，大量的分支结构导致占用的显存过多，造成额外的显存开销。这种做法会降低 YOLO 算法的运行速度。所以 YOLOv7 抛弃了 Resnet 结构。

2.2.3 RepVGG 结构

Resnet 网络的分支结构复杂造成模型的检测速度慢，YOLOv7 采用 RepVGG 网络模型加速，VGG 在过去的 YOLO 版本中并没有取得很好的效果，相对于 Resnet 结构，RepVGG 并没有考虑残差结构，但在 YOLOv7 版本中，RepVGG 网络在 imagenet 上取得了不错的效果，在准确率没有发生改变的情况下检测速度是 EFF（efficientnet）检测速度的 3 倍。RepVGG 独特的多分支融合技术，使其推理速度大大提

升。结构如图 2-8 所示：

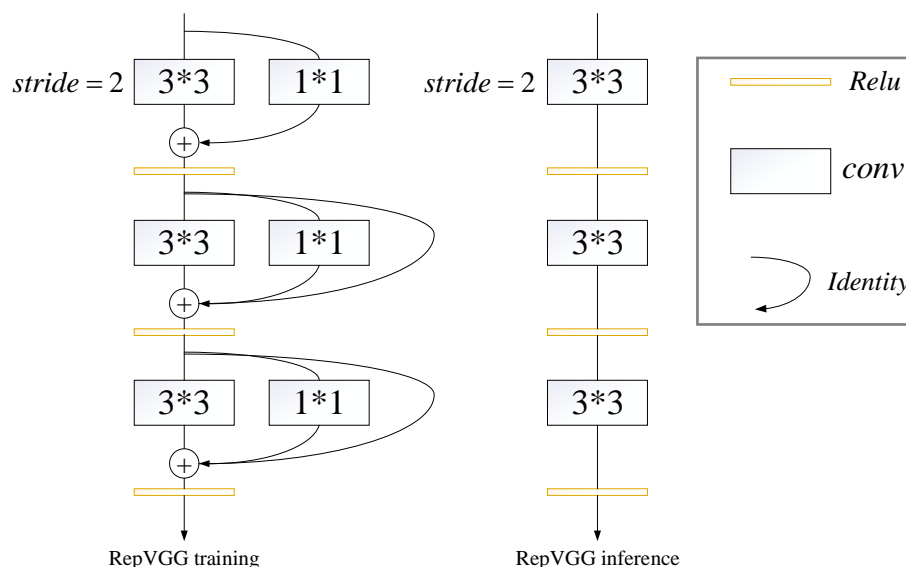


图 2-8 RepVGG 的训练结构和测试结构

英伟达版本的显卡在 3×3 的网络中优化效果是最好的，所以本研究基于 3×3 的卷积模型。YOLOv7 采用不同的训练网络和测试网络。在训练网络中采用卷积连接 BN，BN 再连接卷积这种 Reset 模式训练，在测试网络中把卷积操作和 BN 操作做成并行结构，把 3×3 的卷积和 1×1 的卷积融合，其中 1×1 的卷积不需要 padding，本身就可以保证特征图的尺寸一致，对于 3×3 的卷积需要在原特征图上面 Padding 一圈，然后把 1×1 的卷积核加到 3×3 的卷积核中，完成卷积核的融合操作，如图 2-9 所示。

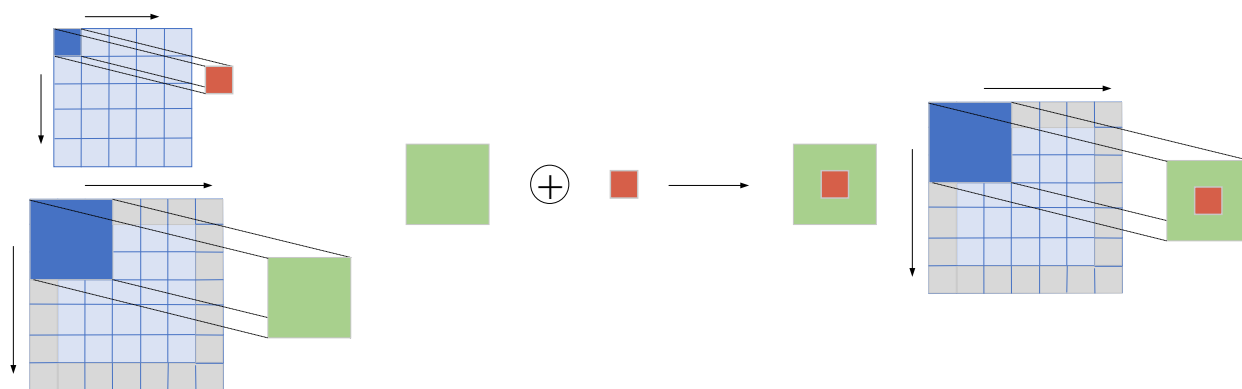


图 2-9 2×2 卷积与 3×3 卷积融合

Rep 卷积层在 YOLO v1 中虽然表现出色，但是将 Rep 卷积层应用到其他的网络模型时，效果并不理想，原因是破坏了原来算法中的残差结构和梯度连接。YOLO v7 作者反复调节 Rep 卷积层的位置，克服了原来深度学习算法的缺点，加入 Rep 卷积层，使得反向传播变的更容易。进而加快了算法的检测速度。

2.2.4 BN 操作与卷积操作的合并

在每次卷积操作完成后，训练的样本就可能变的较为分散，需要用归一化操作，即用 BN 操作使数据向中心靠拢。卷积操作和归一化操作都需要对权重做一次变换，每次 BN 所做的事情是一样的，使用

batch 数据、 x_1 、 x_2 、 $x_3 \dots x_n$ 对他们来进行归一化操作：

$$\hat{x}_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \hat{x}_i = \frac{\gamma x_i}{\sqrt{\sigma^2 + \epsilon}} + \beta - \frac{\gamma \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (3)$$

其中的 μ 和 σ 都是每个单独的 *channel* 中单独计算的，其中 β 是平移系数， γ 是缩放系数。 ϵ 是个很小的数，用来保证分数的分母不为 0。在运算的过程中， μ 和 σ 在每次卷积之后都可以求出来， β 和 γ 是在深度学习迭代中训练出来的。我们将 BN 操作向卷积操作的形式上变换：

$$\begin{pmatrix} F_{1,i,j} \\ F_{2,i,j} \\ \vdots \\ F_{C-1,i,j} \\ F_{C,i,j} \end{pmatrix} = \begin{pmatrix} \frac{\gamma_1}{\sqrt{\sigma_1^2 + \epsilon}} & 0 & \dots & 0 \\ 0 & \frac{\gamma_2}{\sqrt{\sigma_2^2 + \epsilon}} & & \\ \vdots & & \ddots & 0 \\ 0 & 0 & \frac{\gamma_{C-1}}{\sqrt{\sigma_{C-1}^2 + \epsilon}} & \end{pmatrix} \cdot \begin{pmatrix} F_{1,i,j} \\ F_{2,i,j} \\ \vdots \\ F_{C-1,i,j} \\ F_{C,i,j} \end{pmatrix} + \begin{pmatrix} \beta_1 - \gamma_1 \frac{\mu_1}{\sqrt{\sigma_1^2 + \epsilon}} \\ \beta_2 - \gamma_2 \frac{\mu_2}{\sqrt{\sigma_2^2 + \epsilon}} \\ \vdots \\ \beta_{C-1} - \gamma_{C-1} \frac{\mu_{C-1}}{\sqrt{\sigma_{C-1}^2 + \epsilon}} \\ \beta_C - \gamma_C \frac{\mu_C}{\sqrt{\sigma_C^2 + \epsilon}} \end{pmatrix} \quad (4)$$

此时归一化结束，也就得出了 1×1 的卷积，并且有 C 个通道。在数学形式上表现为：

$$F_i = w_i x + b \quad (5)$$

将卷积和 BN 计算拆开来，BN 操作的权重和偏执作为已知条件，将结果运算拆开来为：

$$f_{i,j} = (W_{BN} \cdot f_{i,j} + b_{conv}) + b_{BN} \quad (6)$$

先做卷积操作，再在卷积的基础上做 BN，然后将公式展开：

$$W_{BN} \cdot W_{conv} \cdot f_{i,j} + W_{BN} \cdot b_{conv} + b_{BN} \quad (7)$$

将 $W_{BN} \cdot W_{conv}$ 作为新的权重参数，将 $W_{BN} \cdot b_{conv} + b_{BN}$ 作为新的偏执，由于权重和偏执在上文中已知，此时，由一组新的卷积来代替原来的卷积和 BN 操作，完成卷积和 BN 的合并。

2.2.5 高效的聚合网络

如何设计出来高效的网络？关于这个问题的焦点往往集中在参数的多少、计算复杂程度以及单位时间内的计算量。以上角度是从内存使用的角度出发的。同时要考虑卷积层输出张量的元素数量。在考虑以上要素的基础上，作者提出了拓展版本“E-ELAN”，在梯度路径长度和计算模块数量增加到时都可以达到稳定状态。若参数模块堆叠很多层，那这种稳定状态也会被破坏，参数的利用效率会随着降低，E-ELAN 结构在不破坏原始梯度的情况下，提高学习能力。YOLOv7 模型训练中大量使用 E-ELAN。将两组 ELAN 的输出拼接而成得到 E-ELAN 的输出，因此 YOLOv7 模型在保持有较高的检测精度的基础上具有更高的检测效率。本文采用 YOLOv7 模型用于遥感图像森林火灾检测。检测效率和检测精度相比 YOLOv3 均有很大的提升。在缩放方式中先采用对宽度按比例放大，然后对深度进行同比例放大，对放大后的深度和宽度拼接后组合输出，在进行上面一系列数据变换的同时，还保持了原来数据进行了备份，对原来的数据进行交叉组合处理，同上一步的拼接结果组合，得到最终的输出结果。

2.3 YOLOv7 检测流程

将图像输入到 YOLOv7 模型中，经过卷积层和池化层的特征提取，再由平均池化层把特征图转换为一个一维向量。该向量输入到全连接层中得到边界框(bounding boxes)的位置、大小以及置信度(confidence)，筛选出网格中置信度高于阈值的预测框，再通过 NMS 算法二次筛选，将重叠的预测框去除，最后输出留下的预测框作为检测结果，具体的实验流程如图 2-7 所示。

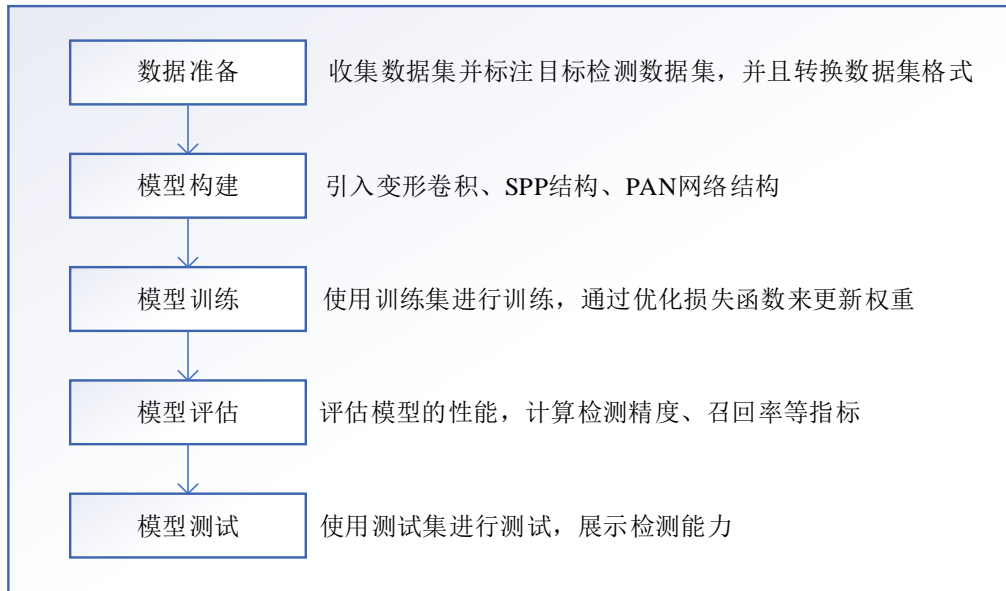


图 2-10 YOLOv7 检测流程

第三章 仿真实验

3.1 样本数据集的选取与构建

3.1.1 数据集选取

本研究所采用的数据集是来自于网上采集的森林火灾的图像，采集方式是无人机拍照。数据集中包含 800 个样本，图像的格式是 JPG。其中 500 张图片是含有森林火灾或烟雾的，90% 火情图片拍摄位置距离火灾中心点在 1000 米内。少量的图片由于火势较大，拍摄距离较远。另外 300 张是没有发生火灾和烟雾的复杂背景图片。此数据集可以用来检测森林中是否出现明火或者烟雾，在出现火灾或者烟雾的时候，为数据集标注 smoke、fire 标签。数据集的处理流程如图 3-1 所示：

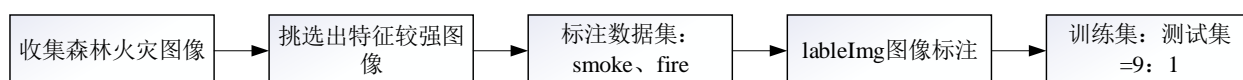


图 3-1 数据集处理流程

出现火灾的数据集中的部分图像如图 3-2 所示。



图 3-2 出现火灾的数据集图像示例

训练集中没有出现火灾的复杂背景图片如图 3-3 所示。



图 3-3 无森林火灾的数据集图像示例

3.1.2 数据集的预处理

原始数据集经过 lableImg 标注后是 xml 格式的标注文件，如图 3-4 所示，需要转化为 YOLO 模型

匹配的数据集格式，使用 Python 中 lxml 库来解析原始数据集，并提取原始数据中标注框的坐标信息，接着将坐标信息映射到[0, 1]内供 YOLO 模型使用。采用的坐标信息是相对于图片大小的，不是绝对坐标。将转换后的相对坐标写于与图片同名的 txt 文件中如图 3-4 所示，每一行包含归一化的坐标信息。完成对数据集的预处理，处理结果如图 3-5 所示。

```
<annotation>
  <folder>photo</folder>
  <filename>fire_11.jpg</filename>
  <path>E:\yolov7-pytorch\VOC2007\photo\fire_11.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>smoke</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>552</xmin>
      <ymin>470</ymin>
      <xmax>791</xmax>
      <ymax>845</ymax>
    </bndbox>
  </object>
</annotation>
```

图 3-4 labelImg 标注结果

```
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_1.jpg 115,3,273,78,1 141,88,250,135,1
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_100.jpg 122,129,178,153,0
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_11.jpg 552,470,791,845,1
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_12.jpg 498,601,700,965,1
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_13.jpg 226,588,387,904,1
E:\yolov7-pytorch\VOCdevkit\VOC2007\JPEGImages\fire_14.jpg 1130,338,1313,687,1
```

图 3-5 转为 YOLO 检测格式的结果

3.1.3 实验环境介绍

本实验采用 Windows 10 操作系统作为基础实验环境；Python 深度学习计算包选择 Pytorch 1.4.0 作为深度学习框架；研究中使用显卡的驱动的版本是 CUDA10.0，采用英伟达公司开发的针对深度学习加速的 GPU 加速库 CudNN7.0.5。深度学习的 CPU 环境为 Intel-i5-10210，采用的 GPU 环境为：RTX2080Ti，实验采用的硬件配置如下表(3-1)所示：

表 3-1 实验环境配置

参数名称	具体规格
CPU	Intel-i5-10210
GPU	RTX2080Ti
RAM	16GB
操作系统	Windows 10
深度学习计算包	Pytorch 1.4.0

3.2 YOLOv7 森林火灾识别检测模型

3.2.1 实验步骤

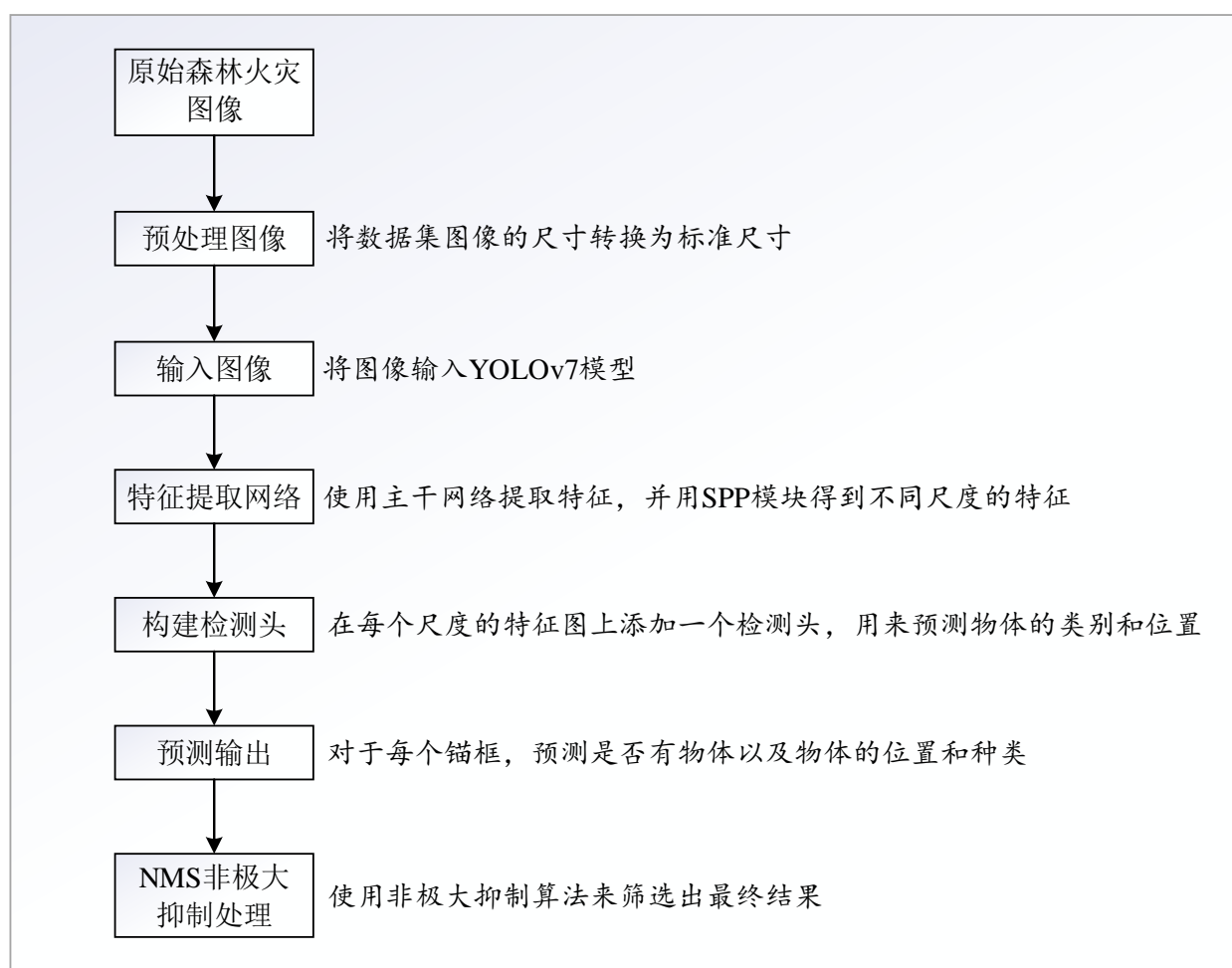


图 3-6 目标检测流程图

数据集准备完成后，采用准备好的数据集开始基于 YOLOv7 模型的森林火灾检测实验，实验的流程如图 3-6 所示。深度学习框架采用 Pytorch 计算包，从 YOLOv7 官方论文的网站中下载模型，根据表 3-1 搭建好实验环境后，开始进行基于深度学习的森林火灾检测实验，具体实验过程如下：

1) 划分训练集、测试集、验证集。将处理好的 YOLO 格式数据集标签文件，分别放到训练集、验证

集、测试集的路径文件夹下。

- 2) 模型调用预处理函数，对输入网络的图片预处理。使用归一化函数将像素值映射到[0, 1]上，在这个区间是深度学习擅长的区间。然后对图像旋转、裁剪等，使得输入模型的图像大小、通道数一致。
- 3) 将数据加载到数据加载器中。实例化一个数据加载器对象，分别将训练集、测试集、验证集的图像和标签传入数据加载器中，同时设置 batch-size 的大小以及是否采用多线程的方式加载。
- 4) 在预训练的基础上对模型训练，由于本实验准备的数据集较少，为了避免模型出现过拟合，在预训练的基础上开始训练，在训练过程中使用梯度下降法来降低损失、调整权重。学习率采用 0.01，由于训练的样本量较小，所以学习率就一直保持 0.01。具体参数如表 3-2 所示。

表 3-2 训练参数设置

参数名称	具体规格
learning rate	1E-2
batch-size	4
最大 epoch	100
预训练模型	yolov7.pt
img-size	640*640

- 5) 将数据加载器中的数据送到模型进行训练，传入的参数是图片和标签。经过计算，网络模型得到了当前的预测结果。
- 6) 将当前的预测结果传入损失函数计算损失值，此时用到 Pytorch 的自动求导机制，通过计算损失和权重之间的偏导，使用如下公式更新权重：

$$w = w - \alpha \frac{\partial loss}{\partial w} \quad (8)$$

- 7) 模型每迭代 10 次计算一次 mAP 值、损失值，使用 Python 库中的 pyplot 包，绘制损失随 epoch 变化的函数图像
- 8) 重复以上步骤，完成 100 次训练之后，得到平均检测精度(mAP)、召回率等指标。对森林火灾的检测效果和可行性做出评估分析。

3.2.2 实验训练结果

- 1) 损失函数图像如图 3-7 所示，随着训练步数的增加，损失逐渐减少，在训练到 100 步的时候，损失下降到 0.04，在训练过程中损失持续下降，模型的训练结果较好。Box 计算出的是 GIoU 损失函数均值，具体计算公式如下：

$$Box = \frac{1}{N} \sum_{i=1}^N (1 - GIoU_i) \quad (9)$$

公式中，N 指 batch_size 的大小， $GIoU_i$ 表示第 i 个样本的损失函数。

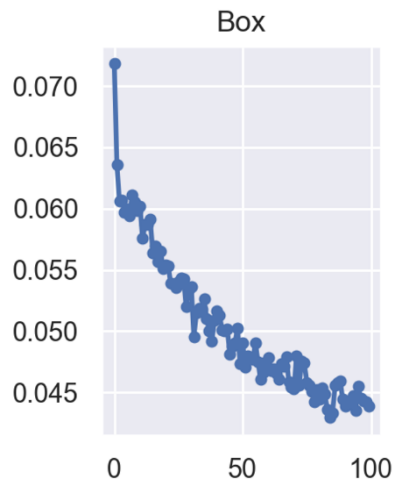


图 3-7 损失函数图像

- 2) 训练后得到的混淆矩阵如图 3-8 所示，可以看到 **fire** 的检测标签 TP 的检测值为 0.55，**smoke** 标签的 FN 值为 1，表明 YOLOv7 在对 **fire** 标签区分是可以的，但对 **smoke** 标签的区分上有很大的问题，由于烟雾数据集标注难度大、标注的范围难以确定、标注检测框数目也难以确定，所以在 **fire** 和 **smoke** 的分类问题上，YOLOv7 模型区分 **smoke** 与 **fire** 的能力有待提升

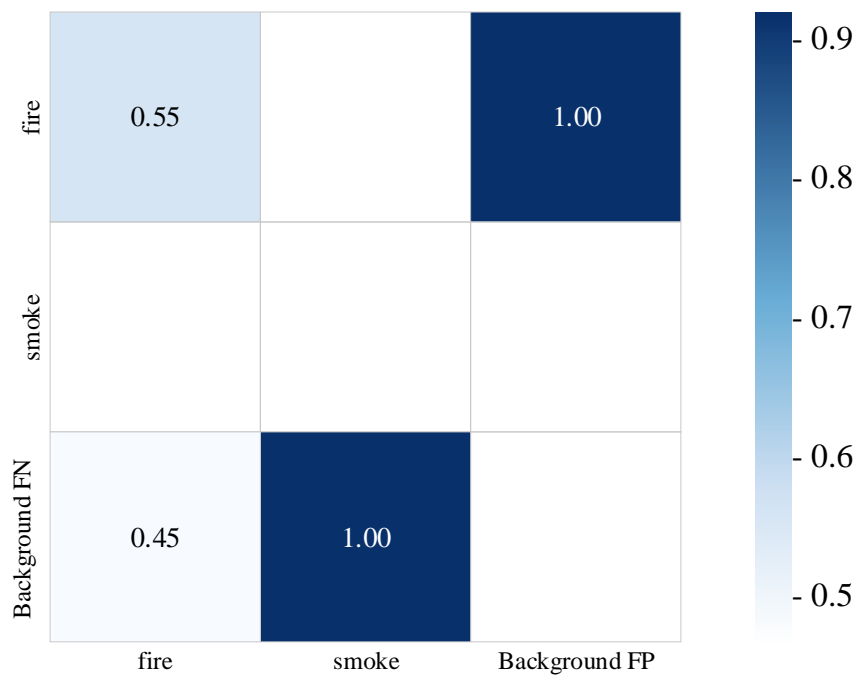


图 3-8 fire、smoke 混淆矩阵

- 3) F1 曲线如图 3-9 所示，F1 的值是衡量分类的指标，可以同时兼顾精确率和召回率，数值范围在[0, 1]之间，在不同的置信度阈值下，F1 得分高则模型的检测效果较好。实验中在置信度取值为[0.2-0.4]之间时，**fire** 的检测检测精度和召回率较高，**smoke** 标签的检测能力则不尽人意，在精度和召回率方面还有待提高。

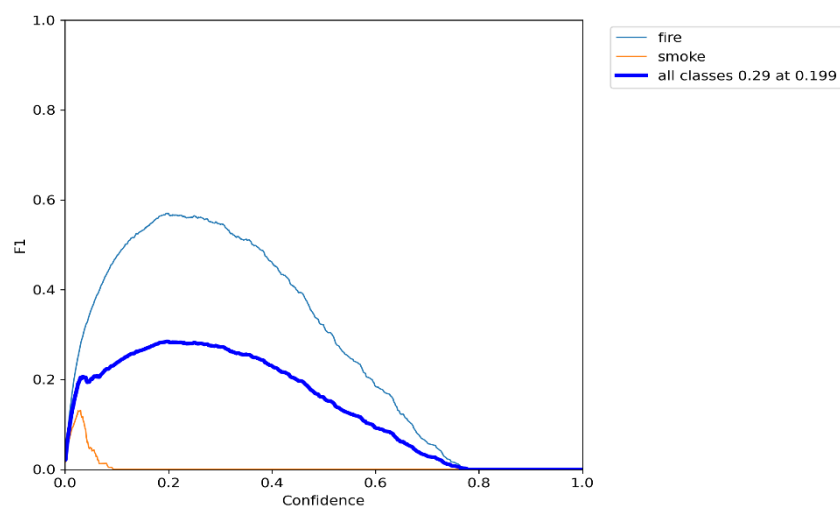


图 3-9 F1 曲线图

- 4) 单一精度随着置信度和召回率的变化如图 3-10(a)和 3-10(b)所示，当置信度大于 0.7 时，**fire** 标签的检测精度趋向于 1，当置信度为 0.681 时，所有分类的平均检测精度达到了最高，**smoke** 标签的检测精度较低。召回率与检测精度的关系曲线较为平稳，表明模型的检测能力较好。

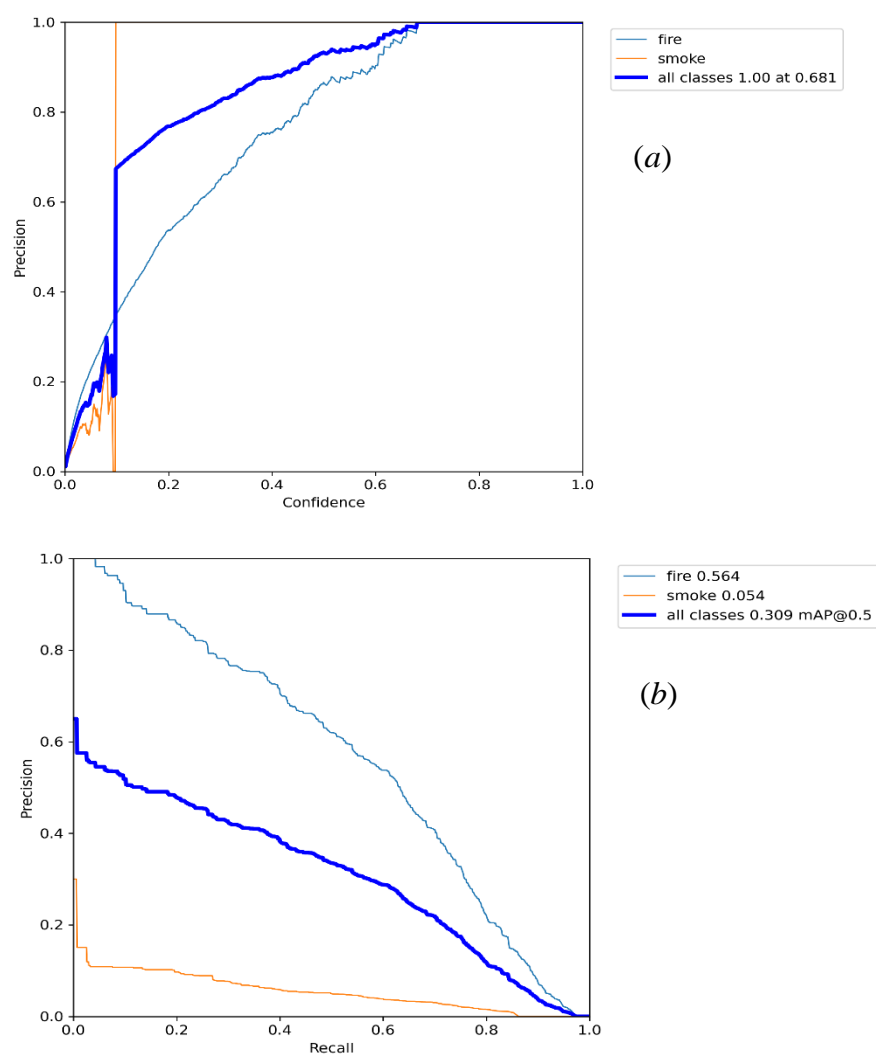


图 3-10 精度随着置信度和召回率的变化

3.2.3 森林火灾图像预测结果



图 3-10 best.pt 在测试集的检测效果

weights 文件即训练好的模型，一般使用 best.pt 去进行推理，训练得到的权重文件包括 best.pt 和 last.pt，以及默认 25epoch 保存一次模型，以及保存最后 5 个 epoch 的训练模型，这里选用 best.pt 进行测试，测试结果如图 3-10 所示。蓝色的框代表检测到的火灾，黄色框表示检测到的烟雾。该测试结果表明，在火势较小，拍摄角度高的图像中，容易分辨出来 fire 和 smoke 标签。但是在火势较大或者是夜间，图像的背景偏灰色，这时对 smoke 标签的检测产生了很大的影响，拍摄角度低的情况下，烟雾弥漫整个图像，肉眼都很难分辨烟雾还是背景，所以对 smoke 标签的检测效果一般。

该检测模型对火焰标签的检测能力较好，根据图像能很好的分辨出来是否发生森林火灾，至于对明火和烟雾的判断，还需要实验的进一步研究，比如选用更多的数据样本、对图像做更有针对性的预处理等。

3.2.4 结果分析

为了对实验结果进行评估，采用召回率(recall)和平均检测检测精度(mAP)评估训练模型的检测能力的大小，平均检测精测精度指在所有分类中检测的整体效果，召回率衡量检出正样本的数量与所有整样本数量的比例，精度(Precision)是所有检测结果框中正确检测的比例，计算方法如下所示：

$$Precision = TP / (TP + FP) \quad (9)$$

$$Recall = TP / (TP + FN) \quad (10)$$

其中 TP 指对正样本预测正确, FP 表示把负样本预测为正样本; TN 表示对负样本预测正确, FN 表示预测出来是负样本, 然而并不是负样本。模型的具体检测结果如表 3-3 所示

表 3-3 训练结果

参数名称	具体数值
检测精度	0.7686
mAP@.5	0.309
mAP@.5:.95	0.1645
训练集总损失	0.06356
训练集分类损失	0.005244
验证集边界框损失	0.06193
Recall	0.3038

根据 3-11 的图像中可以看出, 随着训练步数的增加, 检测精度持续增长, 召回率也有相应的提高。说明 YOLOv7 对森林火灾检测的效果较好, 但是由于 smoke 标签的复杂性, 对检测结果造成了一定的影响, 特别是在黑夜中的烟雾, 在标注数据集的时候都有很大的不确定性, 得到的烟雾检测效果并不理想, 但这并不影响我们对森林火灾的检测, 无论发现是烟雾还是火焰, 都可以判定为森林发生火灾。

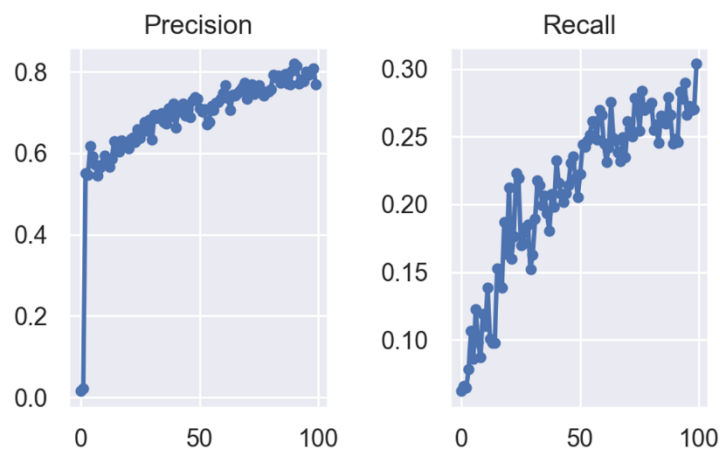


图 3-11 检测结果图像

平均检测精度图像如图 3-12 所示, mAP@0.5 表示在检测结果中, 置信度的阈值取 0.5, mAP@0.5:0.95 指的是在不同的置信度的阈值下检测精度的平均值。当 mAP 的值越大, 表明检测的准确度较高, 该指标结果证明, 在 YOLOv7 模型中, 该数据集在森林火灾的检测中表现良好。

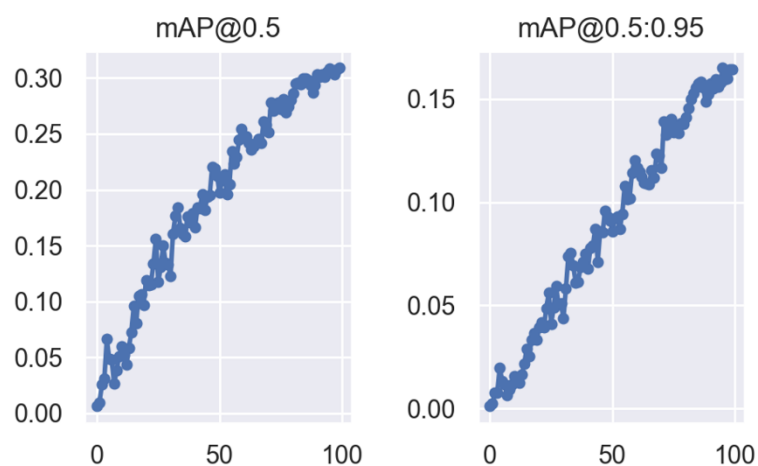


图 3-12 mAP 值和平均 mAP 值

3.3 本章小结

本章对研究内容进行了具体实验，最开始介绍数据集的制作和标注，以及分类的标签和数据集的预处理，然后在 minconda 中构建 Pytorch 虚拟环境，YOLOv7 模型需要部署在 Pytorch 虚拟环境中。将准备好的数据集输入到模型中，设置训练步长、图像输入大小、分类数量等参数，开始对数据集的训练，在训练过程中每 10 步计算一次 AP，并计算当前损失，直到训练完成。最后得到了训练结果，分析训练结果认为该模型可以较好的完成对森林火灾的检测任务。

第四章 总结与展望

森林是大自然的馈赠，是人类赖以生存的物质资源。森林可以显著影响气候的变化，牢牢的抓紧土壤，让它不受风雨的侵蚀。同时也可以净化空气，减少大气污染对人类的伤害。在现代化的生产实践中也对森林非常的依赖，学校用的桌椅、生活用的纸张、家具建材等。森林为人类的工业化进程中提供了不可缺少的物质原料，森林的重要性不言而喻。所以保护森林对人类社会生产发展有举足轻重的作用，其中森林火灾是造成森林破坏的重要因素。

文章从国内外研究状况入手，开始介绍了当前使用深度学习检测森林火灾的研究现状，以及当前用到的一些深度学习技术。然后从人工神经网络介绍最基本的深度学习原理，再进一步引入 YOLO 初代版本的检测原理，接着介绍了 YOLOv7 模型相比于以前的版本的改进点，例如：采用了高效聚合网络、合并卷积层和 BN 层这样的操作来优化训练策略，提高网络的检测能力。

第三章具体介绍了实验步骤过程：

- 1) 选取合适的森林火灾图像，对图像标注，并做预处理，转换为 YOLO 模型支持的标签格式
- 2) 配置模型所需要的虚拟环境，搭建网络模型，将数据集输入到模型中来
- 3) 对模型进行训练，每次将预测结果传入损失函数，计算损失的大小
- 4) 反向传播优化权重矩阵，直到模型训练完毕，保存训练结果
- 5) 分析实验结果，比如精确率、召回率、平均检测精度等

在本实验中，由于对深度学习缺乏系统的学习，以及时间紧迫，没有专门针对森林火灾检测进行参数优化，检测效率和准确度还可以进一步提升。实验所用到的数据集较小，由于网上没有同类数据集，所以需要自己搜集并标注，消耗了大量的时间。如果有机会迈入研究生的学习中，我将加强对计算机视觉的系统性学习，努力做出更完善的森林火灾检测系统。

参考文献

- [1] Bouakkaz F, Ali W, Derdour M. Forest Fire Detection Using Wireless Multimedia Sensor Networks and Image Compression[J]. Journal homepage: <http://iicta.org/journals/i2m>, 2021, 20(1): 57-63.
- [2] 冯炎,雷朝锋,潘岩等.基于卫星遥感技术的森林火灾监测研究[J].测绘技术装备,2022,24(03):15-19.DOI:10.20006/j.cnki.61-1363/P.2022.03.004.
- [3] 张苗,李璞,杨漪等.基于目标检测卷积神经网络的图像型火灾探测算法[J].消防科学与技术,2022,41(06):807-811.
- [4] Du, H.W., Zhu, W.Z., Peng, K. and Li, W.F. (2022) Improved High Speed Flame Detection Method Based on YOLOv7. Open Journal of Applied Sciences, 12, 2004-2018. <https://doi.org/10.4236/ojapps.2022.1212140>
- [5] 皮骏,刘宇恒,李久昊.基于 YOLOv5s 的轻量化森林火灾检测算法研究[J].图学学报,2023,44(01):26-32.
- [6] 祖鑫萍,李丹.基于无人机图像和改进 YOLOv3-SPP 算法的森林火灾烟雾识别方法[J].林业工程学报,2022,7(5):142-149.
- [7] Zhang Q X, Lin G H, Zhang Y M, et al. Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images[J]. Procedia Engineering, 2018, 211: 441-446.
- [8] Zhang Yongtuo et al. Pyramid Attention Based Early Forest Fire Detection Using UAV Imagery[J]. Journal of Physics: Conference Series, 2022, 2363(1)
- [9] 张倩,周平平,王公堂等.基于合成图像的 Faster R-CNN 森林火灾烟雾检测[J].山东师范大学学报(自然科学版),2019,34(02):180-185.
- [10] 秦常帅. 基于深度学习的森林火灾烟雾检测算法研究[D].东华大学,2022.DOI:10.27012/d.cnki.gdhuu.2022.000581.
- [11] 李孝钊. 基于深度学习的轻量化网络火灾烟雾检测方法研究[D].重庆理工大学,2022.DOI:10.27753/d.cnki.gcqgx.2022.000464.
- [12] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once : unified, real-time object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
- [13] Li B, Peng X, Wang Z, et al. Aod-net: All-in-one dehazing network[C]. Proceedings of the IEEE international conference on computer vision, Venice, Italy, 2017: 4770-4778.
- [14] Zhang H, Patel V M. Densely connected pyramid dehazing network[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 2018: 3194-3203.
- [15] 汪文野,刘静,贾南.基于神经网络的森林火灾预测分析[J].武警学院学报,2020,36(06):15-20.
- [16] Novac Ovidiu Constantin et al. Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network[J]. Sensors, 2022, 22(22) : 8872-8872.
- [17] AlSmadi Yazan et al. Early Wildfire Smoke Detection Using Different YOLO Models[J]. Machines, 2023, 11(2) : 246-246.
- [18] 曹林,王欣宇,李兴东.一种基于卷积神经网络的林火蔓延模型[J].林业机械与木工设备,2022,50(06):85-90+95.DOI:10.13279/j.cnki.fmwe.2022.0096.
- [19] Han ByungGil et al. Design of a Scalable and Fast YOLO for Edge-Computing Devices.[J]. Sensors (Basel, Switzerland), 2020, 20(23)

致谢

时光匆匆,在研究生初试的紧张中,开始毕业论文的写作;在被研究生复试支配的焦虑中完成写作;在过线但没学上的失望中完成答辩。

在这里由衷的感谢我的毕业设计指导老师——高强老师,无论是在大二的 C++程序设计的课程、备考研究生阶段,还是做毕业设计遇到麻烦时,高老师都给予了我无私的帮助和指导,在整个研究过程中给了我宝贵的建议和支持。在高老师的耐心指导和不断鼓励使我能够顺利完成这篇论文。