**数据结构与算法**

**期末作业**

题 目：yolov5s图像处理算法的研究

姓 名： 魏靖

学 号： 2021213513

日 期： 2022.05.13

**1 概述**

首先明确定位，yolov5是一种单阶段目标检测算法（和图像识别区分开来），是图像处理算法的一种。本研究基于对yolov5s的研究，探索图像处理背后的具体内容，了解图像处理相关算法的基础和实现，包括：图像压缩、数据处理和提高图像精度三个主要问题。

**2 算法原理**

**2.1 基本结构与实现思路**

yolov5s的结构：

输入端：Mosaic数据增强、自适应图片缩放、自适应锚框计算

主干网络：Focus、CSP

Neck网络：FPN+PAN

输出端：GIOU\_Loss

我们的研究将主要从输入端入手，研究基于yolov5s模型的目标检测算法中数据的收集、存储和初步处理方面的原理。

**2.1.1 认识数据集**

数据结构包括存储结构和逻辑结构，在图像处理的算法实现中，怎么实现图像数据化、建立图像数据库，是图像数据处理的前提和基础。

根据对十余篇建立图像数据库的方法论文及专利的学习，总结出以下问题：

1. 图像数据集的创建应当包含收集和标注两个大步骤；
2. 标注后的数据集的数据应当包含但不限于以下内容：

图像 {  
 名称   
 存放路径  
 图片尺寸  
 其它属性（灰度等等）  
 }

存放路径可以用三种方式来表达：

①给定一个目录（相当于数组）

②给定一个txt文件（文件里面有很多图片的from和名称，不妨把它想象成“口袋”）

③给定一个列表（比如说单链表创建通讯录的思想，和这个就有共通之处）

因而我们看到名称、路径、尺寸等等都可以用相对简单的字符串、指针、数组等来表示，涉及到其他属性时，又不得不提“特征值”。

第一次接触到“特征值”这个概念是在线性代数上，特征值应用在矩阵的变换、压缩上面功能不小，我们的图像“内容属性”的那一部分，也就是将图像抽象为一个向量化的特征值矩阵，这样存储的数据也就可以进一步来讨论处理和应用的问题了。

**2.1.2 图像的“写入”与简单缩放**

特征提取是计算机视觉和图像处理中的概念。它是指使用计算机提取图像信息，进而决定每个图像的点是否属于一个图像特征。特征提取的结果是将图像上的点分为不同的子集，这些子集通常属于孤立的点、连续的曲线或者连续的区域。

而不同的算法有不同的特征点提取想法和优劣比较，目前提取特征的方法主要分为两大类：基于整体特征和基于局部特征的提取方法。局部可能更适合目标检测过程中的锚和对比得到预期框，但在神经网络压缩的过程中失真度可能就会增加。

（1）写入：

●输入端使用的数据增强方式是随机缩放、随机剪裁、随机排布的Mosaic方式。（简单理解就是部分打马赛克，利用强随机的优势，基于概率归一和统计学，得到更为丰富的数据和更精确的结果）

它的优点主要在于：

①提高了数据集的复杂度。对多张图片，进行随机处理，使数据集更丰富，特别是进行随机缩放的操作，可以增加许多小目标，训练得到的模型鲁棒性也会更好；

②减少GPU内存的使用。Mosaic使得一次从数据集读取进行训练的图片张数不需要设置得很大（多张Mosaic拼接之后，在设定的像素部分所能检测的内容就更多），因此，训练时使用一个GPU也能达到比较好的效果。

（2）缩放：

自适应的黑边越少，那么多余的部分就越少，检测的算法效率也就能加快（空间复杂度降低）。

●自适应缩放：（就是个简单的算法实现过程）

①计算长宽的缩放比例，求出长、宽缩放系数

②选择二者间较小的缩放系数

③计算需要填充的区域

int sizec[],sizek[];

int Nc,Nk,N,H,B=0;

const int a=800,b=600;

Nc=800/sizec[i];

Nk=600/sizek[i]; //①

if(Nc>Nk)N=Nk;

else N=Nc; //②

H=(1-N)\*sizek[i]/2

B=(1-N)\*sizec[i]/2 //③

需要注意的是：自适应缩放仅在模型推理时执行，训练时仍然将原始图片裁剪。

**2.1.3 yolov5s的数据处理**

●自适应锚框计算

针对不同目标，初始设定默认长宽的锚框；进行训练时，在此基础上输出一个预测框；计算真实框和预测框的差距；最后反向更新参数。yolov5s将功能嵌入到算法结构中，每次训练时都自适应地计算不同训练集中的最佳锚框值，避免了单独程序的运行，使代码引用相对来说更快捷方便。

具体的指令为：

parser.add\_argument(‘–noautoanchor’, action=‘store\_true’,help=‘disable autoanchor check’);

**2.2 课程联系**

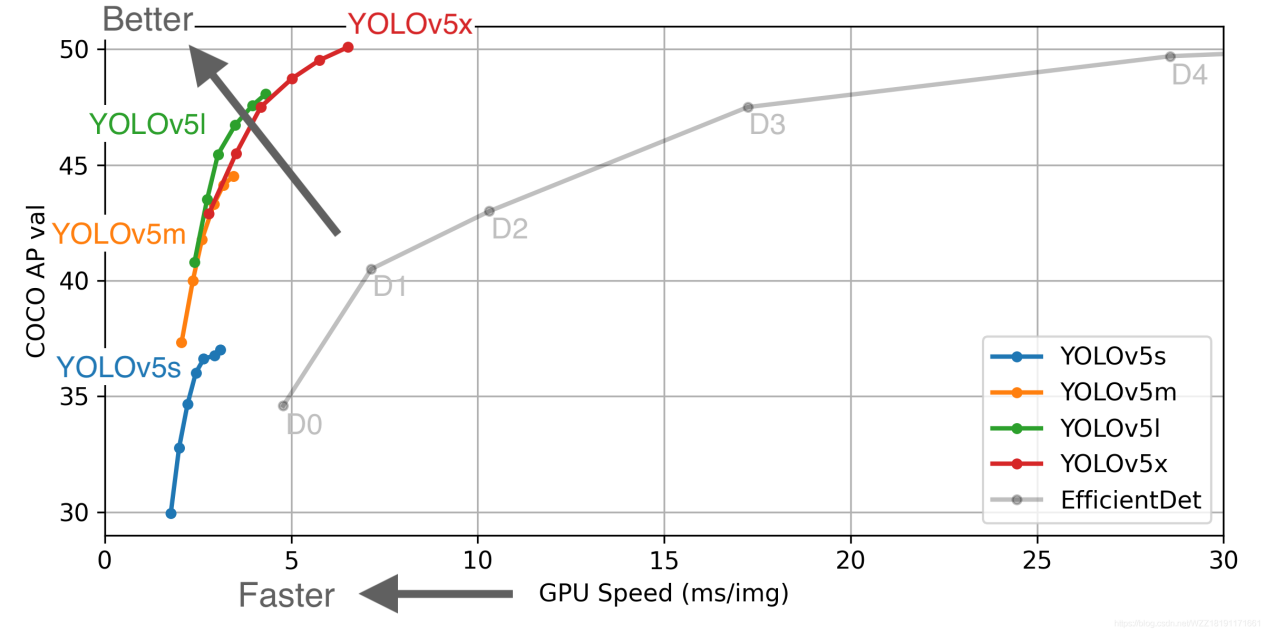
数据结构与算法是人工智能的学科基础，人工智能的研究是基于不断优化的数据结构之上、探索具体算法实践的过程。入门时做的第一个项目的技术基础就在于计算机视觉技术基于人工智能TensorFlow及pytorch框架之上的yolov5，明白了思路，但是对于具体算法的实践依旧处于空白阶段，因此借《数据结构与算法》期末作业的机会，展开部分内容的具体算法研究。

在上述结构的分析中，我们提到了这样一些内容，都和课程关联密切：

1. 数据结构（数据集的建立，包括类和对象的问题）
2. 鲁棒性（程序和算法设计考虑的关键点之一）
3. 自适应图像缩放算法（已有简单编写代码）
4. 其他算法的实现思路

因而，研究yolov5s图像处理算法对加深数据结构与算法的理解、促进课程知识的消化、未来实践应用来说意义匪浅。

**3 结果分析**



本图引自：

https://blog.csdn.net/WZZ18191171661/article/details/113789486

上图横轴表示的是算法在GPU上的推理时间(毫秒/图)，显然越小越Faster；纵轴表示的是该算法在测试数据集上的AP指标（凭《人工智能导论》的印象来说，AP越大越好，具体原理因为书不在手边而网上的东西比较杂所以看着有点吃力，就姑且这样吧）。

通过观察我们可以得出以下结论：

在yolov5网络和EfficientDet的比较中，yolov5的整体精度高于EfficientDet，推理速度均优于后者；yolov5的几种网络中，yolov5s因为网络最小，加之自适应的嵌入，推理速度最快，虽然精度略低，但是仍能达到EfficientDet0的水准。（因为没有比较EfficientDet，研究重点在输入端，而yolov5最大的特点还莫过于神经网络和损失函数GIOU\_Loss，所以原因不多加以说明）

**4 结论**

**4.1 优化想法**

**模型扩展：提高图像精度**

在实际问题中图像可能受到噪声、背景的干扰，也有可能发生平移，还涉及到尺度、视角、光照等等的变化，同时可能受到摄像机性能、画面清晰度的限制，基于此，我认为可以联系一种AI提高分辨率的算法：Real-ESRGAN

参考地址：https://github.com/xinntao/Real-ESRGAN

**4.2 应用前景**

应用yolov5s，实现图像处理的目标检测部分，网络小，速度快，在垃圾分类、人脸捕获、交通信号灯设计中的车流量分析等方面都有很广阔的应用前景。

**5 课程反馈**

**5.1 难易程度**

7

本课程的基本思想其实不难理解，难在算法的实现和优化扩展上面。但本课程的相关思想又十分重要，是后续研究的基础，如果不深入理解、举一反三也还是会陷入一知半解的地步。因而要真正学透彻，难度不小。

**5.2 精力投入**

10

如难易程度所说，从课后思考到实验报告，如果把书上说的内存分配、地图路径等等全部应用到代码实操上来，需要自己下去花很多的时间和精力去查阅资料和文献（也感谢这门课让我学会了使用学校的图书馆资源系统），并且我认为这样的过程对深化理解理论其实很有用。

**5.3 内容**

印象最深的是哈希（因为一讲就明白了其中的思想，让人一下子觉得好妙，老师还通俗易懂地讲了哈希的使用条件等等）。感兴趣的有树——那一章真的是看着好像很简单，但是一回顾就会发现很多东西都和它相关。比如排序树和Huffman编码（现在回忆起来具体实现还有一些吃力）。

**5.4 意见建议**

因为大部分同学属于是上学期才学习C++，所以真的代码实现这一块会需要下很大功夫才能慢慢赶上学习进度——所以我的建议是针对第一学期的。像我自己在寒假鼓起勇气参加了“码拉松”，结果发现真的即使学习了C++的基础，值得提高的方面还是很多，因而我提出：

①从大一上期开始多注重代码实践应用（哪怕就讲那一个语言）。

②班级内成立学习小组（会代码的领着不会的相互交流学习），这样一些交流讨论可以交给小组内完成，从理论结构到代码实操的无缝衔接也会更好吧。

③很喜欢学校出的“学习指导”，因为这本书完全从C++语言的角度进行数据结构与算法的分析和指导，完全符合学习内容，相对于市面上大把大把的C语言参考书来说对编程新手更为友好。