doi: 10. 3969 / j. issn. 1674 - 1242. 2023. 03. 006

Python网络爬虫在医学影像领域的发展现状与趋势研究

白金川[1]，王豪[2]，焦宝园[1]，娄元仓[1]，陈秋冰[2]，李中伟[2]
（1. 新乡医学院第三附属医院，河南新乡 453003；2. 新乡医学院医学工程学院，河南新乡 453003）

【摘要】爬虫是一类收集信息的自动化程序，当前越来越多的领域都在使用爬虫收集目标信息。由于 Python 具有可快速迭代的特性，在主要内容为图像处理与人工智能的医学影像中得到了广泛的应用。为了在保证程序运行效率的同时，减轻训练模型所需数据为计算机存储带来的压力，采用了能够大幅提高程序运行效率的异步式程序，并使用暂态文件保存数据。结果表明，异步式程序、暂态存储程序的运行效率分别是单线程的 4.722 倍、1.433 倍，在医学影像模型训练中使用爬虫可以降低对计算机存储性能的要求。

【关键词】Python；网络爬虫；医学影像；人工智能；发展趋势
【中图分类号】R4，TN，TP 【文献标志码】A 文章编号：1674-1242（2023）03-0260-07

Research on the Development Status and Trends of Python Web Crawlers in Medical Imaging

BAI Jinchuan[1], WANG Hao[2], JIAO Baoyuan[1], LOU Yuancang[1], CHEN Qiubing[2], LI Zhongwei[2]
(1. The Third Affiliated Hospital of Xinxiang Medical University, Xinxiang, Henan 453003, China;
2. Xinxiang Medical University, College of Medical Engineering Xinxiang, Henan 453003, China)

【**Abstract**】Crawlers are automated programs that collect information and are increasingly used in a wide range of fields. Python have been widely used in medical imaging, which mainly involves image processing and artificial intelligence, due to the rapid iteration capability of Python. In order to ensure the efficiency of program execution and reduce the pressure on computer storage caused by the required data for training models, an asynchronous design program that can significantly improve the efficiency of program execution is adopted, and transient files are used to store data. The results showed that the efficiency of asynchronous programs and transient storage programs were 4.722 times and 1.433 times higher than that of single threads, respectively. Using crawlers in medical imaging model training can reduce the requirements for computer storage performance.

【**Key words**】Python; Web Crawlers; Medical Imaging; Artificial Intelligence; Trends

0 引言

随着网络信息量的爆炸式增长[1]，收集有效信息的难度不断增大，越来越多的个人、组织选择使用爬虫工具收集目标信息。由于 Python 具有可快速迭代性，并且 Python 社区拥有众多优秀的第三方库与框架，越来越多的程序员使用 Python 编写爬虫程序。同时，Python 在图像处理领域也有良好的表现[2]，这使其在主要内容为医学图像处理的医学影像学科中得到了广泛的应用。随着现代医学的持续发展和医学图片所蕴含的信息量的不断增加，爬虫将更加深入地应用到医学影像学科。

收稿日期：2022-07-25
基金项目：新乡医学院 2021 年教育教学改革研究项目立项课题。
作者简介：白金川，硕士研究生，河南省南阳市人，研究方向：医学影像学。
通信作者：李中伟，讲师，博士研究生，研究方向：生物医学工程，E-mail：lizhognwei@xxmu.edu.cn。

1　Python 在医学影像领域的应用

Python 作为一种新兴的解释型编程语言被广泛应用于网络爬虫、人工智能（Artificial Intelligence，AI）、图像处理等领域，在主要任务是提高图片质量以达到特征值提取与分类任务的医学影像领域也有广泛的应用[3]。在图像处理领域，Python 有很多能够满足各种需求的第三方库。对于灰度转换、二值化等图像处理任务，可以通过 PIL 库与 OpenCV 库实现。对于图片的滤波，可以使用 SciPy、skimage 实现。在医学影像学领域，应用不同的 AI 模型对大量医学图片进行训练。在医学影像领域最常用的 AI 模型是机器学习（Machine Learning，ML），尤其是机器学习中的神经网络（Neural Networks，NN）[4]在医学影像领域得到了广泛的应用。在神经网络模型的发展上，受益于新技术如图形处理器（Graphics Processing Unit，GPU）、张量处理单元（Tensor Processing Unit，TPU）、云计算的发展，涌现了一大批包括 AlexNet[5]、ResNet[6]和 U-Net[7]在内的新兴优秀神经网络模型。与此同时，受益于庞大的使用者、活跃的开发者与广大的科研学者、医生的努力，在 Python 社区中出现了针对某些具体医学影像领域的第三方库与框架，如 pyradiomics 第三方库在放射医学影像领域的广泛使用[8]、可以连通多个如 ANTs、SPM、FSL 等神经影像学领域常用数据处理软件的 Nipype 框架[9]。无论是从其应用范围还是从其使用深度来看，Python 语言在医学影像领域都得到了广泛的应用。

在信息化社会，爬虫主要从互联网上获取公开的数据；而医学影像中所使用的数据应得到医学伦理委员会的批准，并且数据集需要有效、合法、合理。故现在鲜有研究者直接使用爬虫获取网络中的信息用于医学图像处理模型的训练。爬虫是一种自动收集信息的工具，在主要内容为机器学习的医学影像中所扮演的角色更多的是辅助算法训练。在为数不多的使用爬虫进行医学图像研究的文献中，爬虫扮演的更多的是一种获取、处理必要数据的辅助角色。例如，在常炳国[10]的研究中，使用爬虫技术构建分词词汇表，用于模型的训练。

2　网络爬虫的种类

在日常生活中使用的搜索引擎是通用网络爬虫[11]，如百度、Google、Bing 等。其工作原理是尽可能地将网络上的静态文件爬取下来。但它们并不能满足定制化搜索的需要，还不能爬取动态网页与大部分应用内的信息。针对通用型网络爬虫不能抓取应用内信息的情况，深层网络爬虫[12]出现了。深层网络爬虫能够高效检索出通用网络爬虫不能检索出的信息，如应用内的信息，其代价是爬取的内容十分有限[13]。一般，将以目标信息获取为主要关注点的爬虫定义为聚焦性爬虫，也称为主题爬虫[14]。在主题爬虫的编写中并不会特别注重爬取速度与爬取效率。此类程序的设计核心是尽可能提高目标信息获取率。在现实生活中，为了准确、快速地将网站更新信息保存到本地，出现了增量式网络爬虫。增量式网络爬虫可以在本地数据的基础上，监测并保存目标网站上更新的数据，不断维持本地信息的时效性[15]。

3　Python 爬虫

爬虫本质上是一个自动获取网络信息的自动化程序[16]，通过分析模拟请求返回的响应获取目标信息。任何能够处理网络通信的编程语言都能够编写爬虫程序。网络信息在近年来的爆炸式增长与网页生成技术在过去 20 年中的长足发展（如验证码[17]、异步加载[18]、内容管理系统[19]等），为传统爬虫的设计带来了不少困难。但是，Python 社区中的第三方库，可以有助于快速开发出爬虫程序。

3.1　网页爬虫

网页爬虫是将目标信息存储在网站网页上的一类爬虫[20]。对于仅由 HTML 文件构成的目标网页，可以使用 Python 内置的 urllib 库或 requests 库[21]来爬取。requests 库不仅能够完成简单的网络请求操作，还能够保持网络请求会话，使用第三方 Cookie 等操作，是 Python 社区中最常用的网络请求库之一[22]。对于采用 JavaScript、TypeScript 等语言构成的动态目标网页[23]，在爬取该类网站时，不但需要耗费很大的精力来分析网页源代码，还需要掌握其生成语言的语法。而 Web 自动化工具 Selenium 的出现显著改善了这一状况[22]。Selenium 能够直接控制浏览器，自动渲染网页源代码，直接抓取目标网页数据。值得一提的是，绝大多数爬虫程序都会选择使用 Selenium 通过各种网页验证码，如拖动验证码、简单图片验证码。为了能够快速、大量、高效地抓取目标数据，Python 社区中出现了一些爬虫框架，如 Scrapy、Pyspider、MechanicalSoup 等。

其中，Scrapy 是一款由 Python 语言编写的、开源的网络爬虫框架[22]，是目前在 Python 社区中使用最广泛的爬虫框架。Scrapy 原生支持异步、多线程操作，同时拥有众多强大的插件，能够连通其他数据库工具，如可以使用 Scrapy-Redis 插件做一个分布式网络爬虫[24]。

如图 1 所示，Scrapy Engine 是 Scrapy 框架的核心，主要负责 Scrapy 各个模块之间的通信。在 Scrapy 爬虫（spider）项目开始运行时，首先把在 spider 文件中的起始 url 传入调度器（scheduler）中。调度器与下载中间件协同工作构建对应的网络请求，并将网络请求传入下载器。下载器将从互联网中下载传入的网络请求，并通过下载中间件将得到的响应输送给爬虫程序（spider）。在爬虫程序中清洗输入的网络响应信息，并将结果输送到 Pipeline 中。最终，所有的目标数据都在 Pipeline 中得到保存。Scrapy 项目开始运行后，将执行上述爬取步骤，直到爬取 url 列表为空或出现错误。

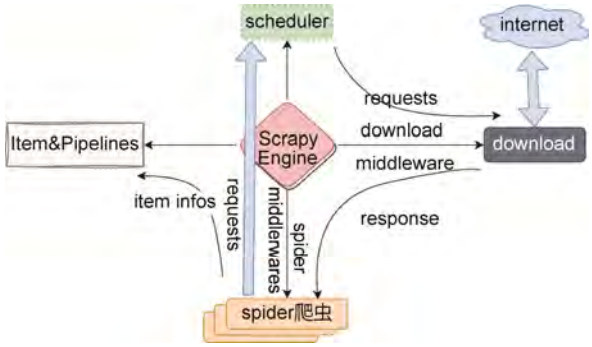

图 1　Scrapy 框架内部结构

Fig.1　Internal structure of a Scrapy frame

3.2　App 爬虫

App 实际上是各个网站的另类门户，其实质仍然是网络通信。App 爬虫通常借助专业的抓包监视软件来监视网络请求信息。抓包监视软件按实现监视的方式可以分为两类：一类是通过设置代理进行监视抓包的软件[25]，称为协议监视抓包软件，如 Charles、mitmproxy 等；另一类是通过监视网卡进行监视抓包的软件，称为网卡监视抓包软件，如 tcpdump、Wireshark 等。相比协议监视抓包软件，网卡监视抓包软件能获取内容更丰富、更广泛的数据包。通过抓包工具分析网络请求内容后，可以使用 requests 或 Scrapy 框架模拟网络请求获取目标数据。此外，移动自动化工具 Appium 能够直接控制 iOS 与 Android 手机完成各种操作[26]，如点击、拖动、滑动等。

3.3　反爬与反反爬

使用爬虫程序进行网络访问的代价很低，可以轻松地制作出大量的网络请求，会严重消耗服务器的网络宽带。网站可以采用一系列反爬措施，如检测请求 IP、UA、reference、Cookie 与验证码，以降低爬虫对网站的不良影响。一般将网站采用的这种措施称为反爬。若爬虫程序在遇到反爬措施时不采取对应的伪装手段，便不能正常访问目标数据。而在爬虫方面，秉承着可见即可爬的原则，将能够突破网站反爬措施的手段称为反反爬。

破解网站反爬的过程类似开盲盒的过程，需要不断实验才能获得目标数据。爬虫程序在设计时并不知道目标网站会采取哪种反爬措施，只有在程序设计过程中不断模拟、伪装请求，获得正确的目标信息，才能确定网站采用的是哪种反爬措施。

3.3.1　针对请求 Headers 字段的反爬

检测网络请求的 Headers 字段，判断是否返回正确信息是网站普遍采用的一种反爬措施。若网站检测的是 reference 字段与 user-agent（UA）字段，只需将浏览器中对应请求的字段加入程序请求的 Headers 字段中，就可以顺利突破网站反爬。

3.3.2　检测请求 Cookie 信息

检测请求 Cookie 信息被广泛应用在需要登录权限的网站中。若检测 Cookie 信息，程序需要建立包含先前访问该网站的信息参数的 session，通过建立的 session 请求，网站可以获得目标数据。在已知 Cookie 的情况下，可以直接将已知的 Cookie 赋值给新建的 session。

3.3.3　检测请求 IP

由于每台计算机的 IP 都是唯一的，若检测请求 IP，则有两种应对方式：一是通过分布式爬虫，将单个爬虫部署在多台计算机上运行；二是购买 IP 商提供的 IP 库，获取多个纯净 IP 供程序使用。

3.3.4　验证码反爬

常见的验证码形式有字符数字图片验证码、滑块验证码等。字符数字图片验证码需要下载验证码图片，进行数字图像处理后得到验证码的结果并通过验证。随着 AI 在图像识别领域的不断发展，破解此类验证码的难度不断下降。滑块验证码需要使用程序操作滑块到指定位置。普通的无 GUI 界面的爬虫难以通过此类

反爬，但借助 Selenium 程序可以方便地通过此验证。值得一提的是，为了提高程序的运行效率，一般都会开启 Selenium 的无头模式。

4　爬虫在医学影像领域的应用

医学影像的主要对象是医学图片，而爬虫本质上是一个网络信息收集工具，故爬虫通常不能直接应用到医学影像领域。但是，由于现代医学影像处理算法通常使用机器学习模型，需要大量数据作为支撑。爬虫作为一种信息收集工具，可以从网络上收集机器学习模型所需的目标数据。故爬虫可用于收集医学影像处理算法的训练数据，如 Codella[27]借助通用爬虫 Bing 将收集到的数据用于其设计的模型训练中。值得注意的是，机器学习不但在训练过程中需要大量的数据作为支撑，在某些模型的设计中也需要大量数据。在这种情况下，可以使用爬虫爬取到的数据来构建目标模型。

算法 1：异步请求，暂态存储。

输入：包含目标图片网址的文件。

```
file urls ← open(file)
session ← aiohttp.ClientSession(headers=headers)
foreach element url of the urls do
    tempfile ← tempfile.TemporaryFile()
    response ← session.get(url)
    tempfile.write(response)
```

输出：暂态文件。

在算法 1 中，使用 asyncio 实现算法的异步化，使用 aiohttp 实现网络请求的异步化，使用 tempfile 将响应保存在一个暂态文件中。暂态文件与普通文件在使用上并无太大差异，主要差别在于暂态文件是有生命周期的，当超过其生命周期时，系统会回收暂态文件所占用的系统资源。使用 set 数据结构构建 urls，实现 url 的去重，避免请求相同的图片。在异步函数 Download_img 中，为了提高网络请求的效率，使用 aiohttp 实现了异步网络请求。为了尽可能提高爬虫的成功率，使用了必要的 UA 伪装。对网络响应的状态码进行判断，若正常则通过 await 返回响应数据。aiohttp 并不是 Python 内置的包，需要借助 Python 的包管理工具，如 pip、conda 或 pipenv，安装后才可正常使用。Binarization 函数实现了图片二值化的功能，模拟医学影像算法，接收 3 个输入。首先通过 PIL 库读取图片文件并灰度化，其次通过遍历列表 table 进行二值化操作，最后保存二值化后的结果到文件中。在异步函数 main 中，遍历 urls 并使用上下文管理器新建一个暂态文件，将网络响应的内容写入该文件。该暂态文件仅在上下文管理器中有效，一旦超出上下文管理器的范围，系统将自动删除该暂态文件。暂态文件的引入可以降低程序的内存使用量，将现有数据保存在文件系统中，方便后续程序的调用，代价是会消耗系统的 I/O 资源。切分输入的 url 字符，得到请求文件在服务器端的文件名 img_name。将文件名 img_name 与暂态文件 tf 输入 Binarization 函数中，得到二值化算法的输出文件。最后，为了更好地与其他模块集成，避免在其他模块引用该文件时运行代码，确保在直接运行该脚本时执行网络请求、暂态文件保存与图片二值化处理，因此在 if_name=='_main_':中使用 asyncio 运行 main 函数，程序结束。

由于Python 3.5 新加入了 async 与 await 语法糖，故只有在 Python 大于或等于 3.5 的解释器中，才可以成功运行以上代码。在使用 await 时，await 的使用对象必须是 Awaitable；可以通过构建_await magic method 实现对象的 Awaitable。yield 是 async 与 await 关键词的基础。Python 中的 yield 关键词在 Python 中起到流程控制的作用：yield 会短暂挂起当前的任务，执行其他任务；当其他任务结束后，继续执行当前的任务。在程序中使用 async 与 await 后，可以使程序具有异步化的特性。网络请求的异步化能够显著提高程序的运行效率。

算法 2：单线程存储。

输入：目标图片的 url。

```
response ← session.get(url)
repeat
    response ← requests.get(url)
file.save(response)//若暂态存储，则使用 tempfile
until 爬取所有 url
```

输出：图片文件。

在算法 2 中，使用 requests 库实现单线程网络请求，使用 json 读取配置信息，使用 path-lib 操作文件系统，使用 tempfile 生成暂态文件，使用 pyinstrument 监测程序耗时。在 Con-trolled_Experiment 中设置了两组对照实验。对照 1：单线程请求，直接保存文件。

对照 2：单线程请求，暂态存储。对照组都需要通过 requests 实现单线程网络请求，但对网络响应的处理不同。在 catch_img_url 中实现网络请求的任务，分别设计了 _download_with_tempfile_decorator 与 _download_without_tempfile_decorator 方法接收单线程网络请求，实现对照组的主要功能。为了避免在大量请求目标网站时，服务器返回错误的响应，在 _decorator_time_delay 方法中实现延时功能：每爬取 50 条 url，程序休眠 50s。最后在 download_with_tempfile_decorator 与 download_without_tempfile_decorator 方法中组装对照组逻辑，实现对照功能。

如图 2 所示，在同一时间段执行对照组与实验组。单线程下载时，直接将响应信息保存在文件系统中；单线程下载、暂态存储时，将响应信息通过 tempfile 保存到暂态文件中；异步下载、暂态保存时，将异步爬虫得到的信息保存在暂态文件中。得到的结果以散点的形式呈现，并使用 SciPy 对数据进行线性拟合。将拟合后的直线与散点一同呈现。拟合直线的斜率为 θ，对应单线程下载，单线程下载、暂态保存，异步下载、暂态保存的拟合直线斜率分别为 $\theta_s=0.420$、$\theta_{sp}=0.293$、$\theta_a=0.089$。单位时间内请求 url 的数量为 $\rho=1/\theta$，对应的 url 请求密度分别为 $\rho_s=2.38$、$\rho_{sp}=3.41$、$\rho_a=11.24$。异步爬虫的效率是单线程爬虫的 4.722 倍，暂态保存爬虫的效率是正常存储爬虫的 1.433 倍。使用暂态文件不仅能够提高网络请求的效率，还能够临时存储数据且不占用过多内存，写出文件到文件系统，供后续程序使用。

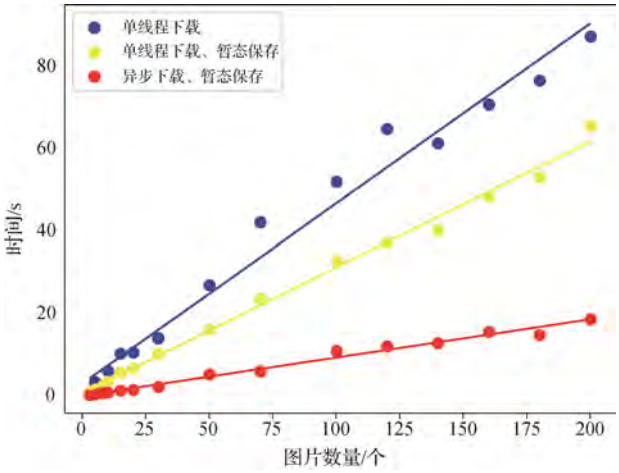

图 2　不同网络请求方式的时间比较

Fig.2　The time comparison of different network request methods

执行单线程网络请求时，程序在发送 HTTP 请求后会短暂地挂起当前线程，直至服务器返回响应后，当前线程才能够继续运行。在当前线程挂起后，CPU 会一直等待服务器的响应，从而大大降低 CPU 的运行效率。异步程序会建立一个任务列表，当使用异步请求时，程序在发送 HTTP 请求后并不会像单线程那样直接挂起，而是去执行列表中的下一个任务。当 HTTP 响应返回时，程序暂时挂起正在执行的其他任务，处理返回的响应。当处理完毕返回的响应后，继续运行被挂起的任务，直至执行完任务列表中的任务。异步程序会有效利用网络请求中的等待时间，使 CPU 的挂起时间显著缩短，程序的运行效率大幅提高。

5　爬虫在医学影像领域的研究趋势

当今世界及互联网数据的爆炸式增长，一方面挑战了计算机的运行速度，另一方面对计算机的存储空间提出了要求。同时，随着拍摄技术的不断发展与拍摄设备的不断更迭，图片所需内存容量不断增大。值得注意的是，机器学习模型的成功通常需要建立在大量的数据基础上。如果在医学影像处理中将训练模型所需要的图片数据存储在本地计算机中，一方面会对计算机的存储容量提出要求，另一方面会极大地浪费计算机存储资源。对于待训练模型的数据来源也需要注意：医学影像算法所使用的数据不同于商业模型所使用的数据，医学图像的采集需要得到医学伦理委员会的批准，并且医学影像模型所用的数据应合法、合理，这为医学信息的收集带来了不小的挑战。此外，有研究表明，由于在数据收集过程中存在差异性，同一算法应用在不同组织收集的数据集上，会出现不同的效果[28]。据此预计未来将出现一个医学图像系统，能够整合医院与其他医疗机构所有的医学图像资源，将其分门别类，最后提供统一的接口供外界请求与访问。

如图 3 所示，将去敏后的患者信息储存到数据库后，可以使用爬虫工具将有价值的去敏医学图像发送到医学图像信息收集系统中。该系统不但能够接收医院中有效的医学图像信息，也能够接收互联网上其他个人和组织提供的有价值的医学图像。但所有接收的图像信息都必须遵循统一的格式，以便最大化地提高算法的可移植性。同时，该系统将提供一个 API 接口，用于传输足够的医学研究所需要的训练图像，以支撑

AI 模型的训练。值得一提的是，爬虫的并行化能够显著减少由于等待网络连接而造成的非必要程序耗时，显著提高效率。同时在 AI 算法上，为了获取更快的处理速度，更有效地利用系统资源，未来的 AI 算法将向并行化算法领域不断发展。虽然爬虫与算法的并行化将提高编写算法的难度，但相信未来的开源算法框架能够解决该问题。

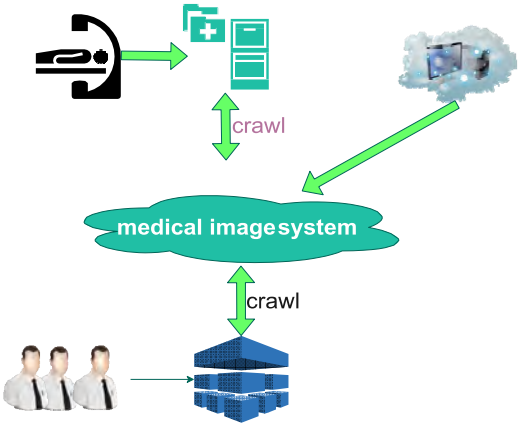

图 3　新型医学图像信息收集系统

Fig.3　New medical image information collection system

6　总结与展望

本文首先阐述了爬虫的定义与分类，并对使用 Python 语言编写爬虫的工具做了简要介绍。之后基于爬虫在当前医学影像领域的实际应用，提出了爬虫今后在医学影像领域的发展趋势，并着重设想了能统一收集医学图像信息并开放网络请求 API 的医学图像信息收集系统。总体来说，爬虫在医学影像领域主要起到以下两方面的作用：一方面用来收集医学图像的处理算法所需要的目标数据；另一方面用来辅助医学图像处理算法的构建。爬虫本质上是一个网络信息收集工具，而当今医学影像学的主要研究方向是在医学图像中应用不同的 AI 模型以取得更好的效果。医学影像领域的主要研究内容决定了爬虫在医学影像中只能起到辅助作用，并且该现状在今后一段时间内仍将持续存在。同时，在今后的研究中，爬虫与医学影像算法将朝着更高资源利用率的并行化方向发展。

参考文献

[1] KUMAR S, YADAV A K, BHARATI R, *et al*. Modeling and analyze the deep web: Surfacing hidden value[J]. **International Journal of Computer Science and Information Security**, 2011, 9 (6): 119.

[2] CHITYALA R, PUDIPEDDI S. Image processing and acquisition using python[M]. Chapman and Hall/CRC, 2020.

[3] GULO C A, SEMENTILLE A C, TAVARES J M R. Techniques of medical image processing and analysis accelerated by high-performance computing: a systematic literature review[J]. **Journal of Real-Time Image Processing**, 2019, 16(6): 1891-1908.

[4] KONONENKO I. Machine learning for medical diagnosis: history, state of the art and per-spective[J]. **Artificial Intelligence in medicine**, 2001, 23(1): 89-109.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep con-volutional neural networks[J]. **Advances in Neural Information Processing Systems**, 2012, 25: 1097-1105.

[6] HE K, ZHANG X, REN S, *et al*. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[7] RONNEBERGER O, FISCHER P, BROX T. U-net: convolutional networks for biomedical image segmentation[C]//International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, 2015: 234-241.

[8] 刘进，印宏坤，陈果，等. MRI 组学特征构建机器学习模型预测胸腰段再骨折[J]. **中国组织工程研究**，2022，26（33）：5323.
LIU Jin, YIN Hongkun, CHEN Guo, *et al*. A machine learning prediction model based on MRI radiomics for refracture of thoracolumbar segments[J]. **Chinese Journal of Tissue Engineering Research**, 2022, 26(33): 5323.

[9] CLOSE T G, WARD P G, SFORAZZINI F, *et al*. A comprehensive framework to capture the arcana of neuroimaging analysis[J]. **Neuroin Formatics**, 2020, 18(1): 109-129.

[10] 常炳国，刘清星. 基于深度学习的慢性肝病 CT 报告相似度分析[J]. **计算机应用与软件**，2018，35（8）：289-294，302.
CHANG Bingguo, LIU Qingxing. Similarity analysis of CT report of chronic liver diseases based on deep learning[J]. **Computer Applications and Software**, 2018, 35(8): 289-294, 302.

[11] 张红云，刘炜，熊前兴. 一种基于语义本体的网络爬虫模型[J]. **计算机应用与软件**，2009，26（11）：101-103.
ZHANG Hongyun, LIU Wei, XIONG Qianxing. A web crawler model based on semantic ontology[J]. **Computer Applications and Software**, 2009, 26(11): 101-103.

[12] 杨俊峰，黎建辉，杨风雷. 深层网站 Ajax 页面数据采集研究综述[J]. **计算机应用研究**，2013，30（6）：1606-1610.
YANG Junfeng, LIN Jianhui, YANG Fenglei. Survey on research of data collection from supporting Ajax technology deep Web sites[J], **Application Research of Computers**, 2013, 30(6): 1606-1610.

[13] 侯东阳，武昊，王军锋，等. 基于深层网络爬虫的 Web 地图服务发现方法[J]. **地理与地理信息科学**，2015，31（5）：10-13.
HOU Dongyang, WU Hao, WANG Junfeng, *et al*. A web map service discovery method based on deep web crawler[J]. **Geography and Geo-Information Science**, 2015, 31(5): 10-13.

[14] 王景中，邱铜相. 基于 TF-IDF 改进算法的聚焦主题网络爬虫[J]. **计算机应用**，2015，35（10）：2901-2904.
WANG Jingzhong, QIU Tongxiang. Focused topic web crawler based on improved TF-IDF alogorithm[J]. **Journal of Computer Applications**, 2015, 35(10): 2901-2904.

[15] 韩逸. 基于增量式爬虫的搜索引擎系统的设计与实现[Z]. 2015.
HAN Yi. Design and implementation of a search engine system based on incremental crawler [Z]. 2015.

[16] 周立柱，林玲. 聚焦爬虫技术研究综述[Z]. 2005.
ZHOU Lizhu, LIN Ling. Survey on the research of focused crawling technique[Z]. 2005.

[17] ELFARDY H, AL-BADRASHINY M, DIAB M. Aida: identifying code switching in infor-mal arabic text[C]//Proceedings of The First Workshop on Computational Approaches to Code Switching, 2014: 94-101.

[18] GARRETT J J, *et al*. Ajax: a new approach to web applications[Z]. 2005.

[19] COLLABORATION C, *et al*. Technical proposal for the upgrade of the CMS detector through 2020[Z]. 2011.

[20] 梁武，苏燕. 协方差特征爬虫网页语义概念树构建方法[J]. **科技通报**，2015，31（4）：85-87.
LIANG Wu, SU Yan. Construction method of webpage semantic concept tree based on covariance features reptile[J]. **Bulletin of Science and Technology**, 2015, 31(4): 85-87.

[21] CHANDRA R V, VARANASI B S. Python requests essentials[M]. Packt Publishing Ltd, 2015.

[22] MITCHELL R. Web scraping with python: collecting more data from the modern web[M]. O'Reilly Media, Inc., 2018.

[23] ARTZI S, DOLBY J, JENSEN S H, *et al*. A framework for automated testing of JavaScript web applications[C]//Proceedings of the 33rd International Conference on Software Engineering, 2011: 571-580.

[24] HAN X, ZHENG L. Design and implementation of firmware data acquisition system based on scrapy framework[C]//2020 IEEE international conference on power, intelligent computing and systems (ICPICS). IEEE, 2020: 168-174.

[25] 王跃，于世伟，路博，等. 基于爬虫的移动互联网应用监测分析系统研究[J]. **电视技术**，2015，39（13）：88-92.
WANG Yue, YU Shiwei, LU Bo. Research on domestic mobile internet application monitoring and analysis system based on crawler technology[J]. **Video Engineering**, 2015, 39(13): 88-92.

[26] 张胜桥，尹青，常瑞，等. 基于 APK 的 Android 应用程序 GUI 遍历自动化方法[J]. **计算机应用**，2016，36（11）：3178-3182.
ZHANG Shengqiao, YIN Qing, CHANG Rui, *et al*. APK-based automatic method for GUI traversal of Android applications[J]. **Journal of Computer Applications**, 2016, 36(11): 3178-3182.

[27] CODELLA N, CONNELL J, PANKANTI S, *et al*. Automated medical image modality recog-nition by fusion of visual and text information[C]// International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2014: 487-495.

[28] ZECH J R, BADGELEY M A, LIU M, *et al*. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study[J]. **PLoS Medicine**, 2018, 15(11): e1002683.