



(12) 发明专利申请

(10) 申请公布号 CN 113609028 A

(43) 申请公布日 2021. 11. 05

(21) 申请号 202111174173.1

(22) 申请日 2021.10.09

(71) 申请人 统信软件技术有限公司

地址 100176 北京市大兴区北京经济技术
开发区科谷一街10号院12号楼18层

(72) 发明人 黄明强

(74) 专利代理机构 北京瀚方律师事务所 11774

代理人 周红力

(51) Int. Cl.

G06F 11/36 (2006.01)

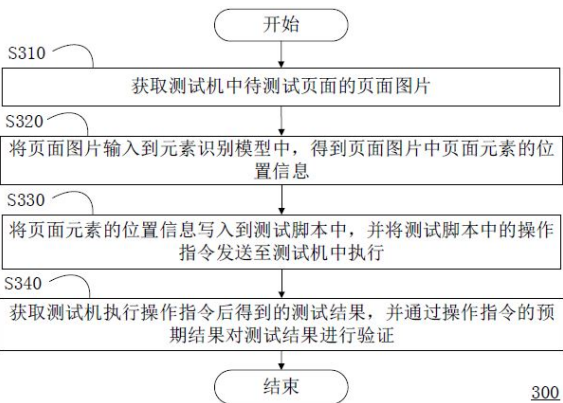
权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种页面测试方法、计算设备及可读存储介
质

(57) 摘要

本发明公开了一种页面测试方法、计算设备及可读存储介质,该方法在服务器中执行,服务器与测试机连接,包括:获取测试机中待测试页面的页面图片;将页面图片输入到元素识别模型中,得到页面图片中页面元素的位置信息;将页面元素的位置信息写入到测试脚本中,并将测试脚本中的操作指令发送至测试机中执行;获取测试机执行操作指令后得到的测试结果,并通过操作指令的预期结果对测试结果进行验证。根据本发明的技术方案,提升了页面元素识别的准确性。



1. 一种页面测试方法,在服务器中执行,所述服务器与测试机连接,所述方法包括步骤:

获取测试机中待测试页面的页面图片;

将所述页面图片输入到元素识别模型中,得到页面图片中页面元素的位置信息;

将页面元素的位置信息写入到测试脚本中,并将测试脚本中的一个或多个操作指令发送至测试机中执行,所述操作指令包括所述页面元素的位置信息;

获取测试机执行操作指令后得到的测试结果,并通过所述操作指令的预期结果对测试结果进行验证。

2. 如权利要求1所述的方法,其中,所述元素识别模型通过以下步骤生成:

获取多个包含页面元素的图片素材;

对获取到的图片素材中的页面元素进行标注,并对图片素材中的页面元素进行截取,将标注并截取后的多个图片作为训练集;

基于所述训练集,对待训练的元素识别模型进行训练,得到训练好的元素识别模型。

3. 如权利要求2所述的方法,其中,所述将所述页面图片输入到元素识别模型中,得到页面图片中页面元素的位置信息的步骤,包括:

将所述页面图片输入到所述训练好的元素识别模型中,得到框选出页面元素的页面图片;

基于所述框选出页面元素的页面图片,获取所述框选出的页面元素在所述页面图片中的位置信息。

4. 如权利要求1至3中任一项所述的方法,其中,将测试脚本中的操作指令发送至测试机中的步骤,包括:

将所述测试脚本中的操作指令发送至所述测试机中键盘鼠标操作接口,以便所述测试机执行所述操作指令来模拟鼠标和键盘对页面元素的操作。

5. 如权利要求1至3中任一项所述的方法,其中,所述通过所述操作指令的预期结果对测试结果进行验证的步骤包括:

将所述测试结果与操作指令的预期结果进行比对;

若所述测试结果与所述预期结果一致,则验证通过;

若所述测试结果与所述预期结果不一致,则执行中断处理并输出异常结果;

将所述测试脚本中的下一个操作指令发送至测试机中执行,直至所述测试脚本中的全部操作指令执行完毕。

6. 如权利要求1所述的方法,所述服务器还与视频采集装置连接,在所述获取测试机中待测试页面的页面图片的步骤之前,还包括:

获取视频采集装置所采集的测试机中待测试页面的页面视频,所述页面视频包括测试机中待测试页面的页面图片。

7. 如权利要求2或3所述的方法,其中,所述对获取到的图片素材中的页面元素进行标注的步骤,包括:

采用数据标注工具对获取到的图片素材中的页面元素进行标注。

8. 如权利要求2或3所述的方法,其中,所述待训练的元素识别模型包括:采用One-stage算法或Two-stage算法的预训练模型。

9.一种计算设备,包括:

一个或多个处理器;

存储器;以及

一个或多个装置,所述一个或多个装置包括用于执行根据权利要求1至8中任一项所述的方法的指令。

10.一种存储一个或多个程序的计算机可读存储介质,所述一个或多个程序包括指令,所述指令当由计算设备执行时,使得所述计算设备执行根据权利要求1至8中任一项所述的方法。

一种页面测试方法、计算设备及可读存储介质

技术领域

[0001] 本发明涉及软件测试领域,尤其涉及一种自动化页面测试方法、计算设备及可读存储介质。

背景技术

[0002] 在用户界面(User Interface,简称UI)自动化测试领域,通常需要定位元素控件在屏幕中的位置,然后通过控制鼠标或键盘在元素控件位置上进行触发操作,最后判断测试用例是否执行成功。在现有方案中,采用OpenCV提供的模板匹配功能,通过对比目标小图和屏幕大图来进行目标元素的定位。然而,采用该方案进行目标定位和自动化测试用例断言时,UI若发生细微的变化都会导致目标元素识别失败,从而导致测试用例不够健壮;而对于自动化测试人员,则需要频繁更换图片资源,增加了自动化测试用例的维护成本,而当自动化测试项目保存了大量的图片资源时,对图片资源的管理和维护也将成为难题。而且,由于OpenCV在ARM和MIPS架构上没有对应的官方包,存在无法使用pip工具安装OpenCV的问题,导致搭建自动化测试环境费力耗时。

[0003] 在现有方案中,还有通过应用属性来获取应用的元素控件坐标的方案,例如,在Web端中,通过获取元素控件的html标签属性来获取其在浏览器中的位置;在桌面应用端,通过基于Accessibility对Qt应用的元素控件添加ObjectName来定位元素控件的坐标;在App端,通过应用的标签属性获取元素控件的坐标。然而,上述方案存在元素控件定位不稳定、速度慢、维护难的问题。

[0004] 鉴于上述原因,需要一种页面测试方案,以提升元素控件识别的准确性和速度、提升自动化测试执行的效率,降低自动化测试用例代码测试资源的维护成本。

发明内容

[0005] 为此,本发明提供了一种页面测试方法、计算设备及可读存储介质,以力图解决或至少缓解上面存在的问题。

[0006] 根据本发明的第一个方面,提供了一种页面测试方法,在服务器中执行,服务器与测试机连接,方法包括步骤:获取测试机中待测试页面的页面图片;将页面图片输入到元素识别模型中,得到页面图片中页面元素的位置信息;将页面元素的位置信息写入到测试脚本中,并将测试脚本中的一个或多个操作指令发送至测试机中执行,操作指令包括所述页面元素的位置信息;获取测试机执行操作指令后得到的测试结果,并通过操作指令的预期结果对测试结果进行验证。

[0007] 可选地,在根据本发明的一种页面测试方法中,元素识别模型通过以下步骤生成:获取多个包含页面元素的图片素材;对获取到的图片素材中的页面元素进行标注,并对图片素材中的页面元素进行截取,将标注并截取后的多个图片作为训练集;基于训练集,对待训练的元素识别模型进行训练,得到训练好的元素识别模型。

[0008] 可选地,在根据本发明的一种页面测试方法中,将页面图片输入到元素识别模型

中,得到页面图片中页面元素的位置信息的步骤,包括:将页面图片输入到训练好的元素识别模型中,得到框选出页面元素的页面图片;基于框选出页面元素的页面图片,获取框选出的页面元素在页面图片中的位置信息。

[0009] 可选地,在根据本发明的一种页面测试方法中,将测试脚本中的操作指令发送至测试机中的步骤,包括:将测试脚本中的操作指令发送至测试机中键盘鼠标操作接口,以便测试机执行操作指令来模拟鼠标和键盘对页面元素的操作。

[0010] 可选地,在根据本发明的一种页面测试方法中,通过操作指令的预期结果对测试结果进行验证的步骤包括:将测试结果与操作指令的预期结果进行比对;若测试结果与预期结果一致,则验证通过;若测试结果与预期结果不一致,则执行中断处理并输出异常结果;将测试脚本中的下一个操作指令发送至测试机中执行,直至测试脚本中的全部操作指令执行完毕。

[0011] 可选地,在根据本发明的一种页面测试方法中,服务器还与视频采集装置连接,在获取测试机中待测试页面的页面图片的步骤之前,还包括:获取视频采集装置所采集的测试机中待测试页面的页面视频,页面视频包括测试机中待测试页面的页面图片。

[0012] 可选地,在根据本发明的一种页面测试方法中,对获取到的图片素材中的页面元素进行标注的步骤,包括:采用数据标注工具对获取到的图片素材中的页面元素进行标注。

[0013] 可选地,在根据本发明的一种页面测试方法中,数据标注工具包括labelme或labelImg。

[0014] 可选地,在根据本发明的一种页面测试方法中,待训练的元素识别模型包括:采用One-stage算法或Two-stage算法的预训练模型。

[0015] 根据本发明的另一个方面,提供了一种计算设备,包括:一个或多个处理器;存储器;以及一个或多个装置,一个或多个装置包括用于执行根据本发明的一种页面测试方法中的任一指令。

[0016] 根据本发明的还有一个方面,提供了一种存储一个或多个程序的计算机可读存储介质,一个或多个程序包括指令,指令当由计算设备执行时,使得计算设备执行根据本发明的一种页面测试方法中的任一方法。

[0017] 根据本发明的技术方案,通过元素识别模型对实时采集的视频进行识别,确定元素控件的位置信息,通过自动化测试脚本封装键盘鼠标的操作对测试集执行自动化测试,并通过服务器执行智能化分析,确定测试结果是否与预期一致,若一致,则继续执行自动化测试用例。本方案采用的智能识别模型解决了传统UI自动化测试中元素控件定位不准确、断言不稳定不准确的问题,提升了UI自动化测试中元素控件识别的准确性和断言的灵活性。

[0018] 进一步地,本方案中服务器、测试机以及视频采集装置设备分离,数据处理的部分由服务器来处理,处理速度非常快,解决了传统UI自动化测试中定位元素慢的问题,以及测试用例执行慢的问题,提升了自动化测试的处理和执行速度。

[0019] 上述说明仅是本发明技术方案的概述,为了能够更清楚了解本发明的技术手段,而可依照说明书的内容予以实施,并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂,以下特举本发明的具体实施方式。

附图说明

[0020] 为了实现上述以及相关目的,本文结合下面的描述和附图来描述某些说明性方面,这些方面指示了可以实践本文所公开的原理的各种方式,并且所有方面及其等效方面旨在落入所要求保护的主题的范围内。通过结合附图阅读下面的详细描述,本公开的上述以及其它目的、特征和优势将变得更加明显。遍及本公开,相同的附图标记通常指代相同的部件或元素。

[0021] 图1示出了根据本发明一个实施例的页面测试系统100的结构示意图;
图2示出了根据本发明一个实施例的计算设备200的示意图;
图3示出了根据本发明一个实施例的页面测试方法300的示意图;以及
图4示出了根据本发明另一个实施例的页面测试方法400的示意图。

具体实施方式

[0022] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例,然而应当理解,可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反,提供这些实施例是为了能够更透彻地理解本公开,并且能够将本公开的范围完整的传达给本领域的技术人员。

[0023] 在现有方案中,大多为通过应用属性来获取应用的元素控件坐标的方案,例如,在Web端中,通过获取元素控件的html标签属性来获取其在浏览器中的位置;在桌面应用端,通过基于Accessibility对Qt应用的元素控件添加ObjectName来定位元素控件的坐标;在App端,通过应用的标签属性获取元素控件的坐标。然而,上述方案存在元素控件定位不稳定的问题,由于通过Web端传输网络请求需要时间,因此采用属性定位的方式存在概率性失败的情况。上述方案还存在速度慢的问题,通过属性进行元素的定位实质上通过寻找整个属性标签的树形结构来进行元素的定位,若属性的数量庞大,树形结构层级较多,元素定位将花费很多时间,定位速度慢。上述方案还存在一定的局限性,若存在某个元素控件没有添加属性标签的情况,那么采用元素定位相较于相对定位或者索引定位会增加自动化脚本后期的维护难度和成本。

[0024] 针对现有技术中存在的问题,本发明提供了一种页面测试方法,以通过训练生成元素识别模型。利用该元素识别模型对采集的测试机待测试页面的页面图片进行识别得到准确度高的页面元素位置信息,利用该位置信息构建测试脚本,以提高元素控件识别的准确性和速度、提升自动化测试执行的效率,降低自动化测试用例代码测试资源的维护成本。

[0025] 需要说明的是,页面元素,也称界面元素、元素、元素控件、控件元素,是指可满足交互需求的软件、页面或系统界面所包含的满足用户交互要求的一系列元素。另外,页面,也指用户界面、UI、界面。

[0026] 图1示出了根据本发明一个实施例的页面测试系统100的结构示意图。如图1所示,页面测试系统100包括测试机110、视频采集装置120和服务器130。

[0027] 测试机110是任何需要进行页面测试的计算设备。例如,在UI测试中,测试机可以是一个带有待测试的页面的计算设备,在进行测试的过程中,将待测试的页面呈现在计算设备的显示器的显示窗口中,以便对待测页面进行后续的测试步骤。

[0028] 视频采集装置120设置在测试机呈现待测试页面的显示窗口前方,例如,视频采集

装置120的位置正对着测试机的待测试页面的显示窗口,使视频采集装置120可以采集到测试机上待测试页面的内容。视频采集装置120的位置可以设置在,当使用视频采集装置120所采集的画面上只有待测试页面,其余的干扰物尽可能少的时候视频采集装置所处的位置。视频采集装置120可以用来采集测试机待测试页面的页面视频或者页面图片。在一种实施例中,视频采集装置120实时采集测试机待测试页面的页面视频,以便根据待测试页面的实时情况对页面进行自动化测试。视频采集装置例如是相机、摄像机、手机、计算机等一切具备拍摄或摄像功能的装置。

[0029] 服务器130与视频采集装置120通信连接,以便服务器130可以获取到视频采集装置120所获取到的测试机110上待测试页面的页面视频或页面图片。服务器130也与测试机110通信连接,以便服务器130向测试机110发送操作指令,以及接收来自测试机110的测试结果。服务器在接收到来自视频采集装置120的页面视频或页面图片后,将根据元素识别模型识别出的页面元素的位置来生成测试脚本,再将测试脚本中包含的操作指令发送至测试机110中,以便测试机110执行操作指令。在测试机110执行操作指令后生成测试结果,测试机110会将测试结果发送至服务器130中。服务器130接收到该测试结果后,会对测试结果与该操作指令对应的预期结果进行比对校验。服务器130进行校验后,服务器130还将持续向测试机发送下一步的操作指令,直至当前测试脚本的全部的操作指令测试完成,然后接着进行下一个测试用例对应的测试脚本的测试缓解,直至全部测试用例测试完成,测试结束。

[0030] 在下文中将结合附图描述通过测试机110、视频采集装置120和服务器130实现的页面测试方法。

[0031] 根据本发明的实施方式,上述的测试机110、视频采集装置120和服务器130均可以如下所述的计算设备200来实现。图2示出了根据本发明一个实施例的计算设备200的示意图。

[0032] 如图2所示,在基本的配置202中,计算设备200典型地包括系统存储器206和一个或者多个处理器204。存储器总线208可以用于在处理器204和系统存储器206之间的通信。

[0033] 取决于期望的配置,处理器204可以是任何类型的处理,包括但不限于:微处理器(μ P)、微控制器(μ C)、数字信息处理器(DSP)或者它们的任何组合。处理器204可以包括诸如一级高速缓存210和二级高速缓存212之类的一个或者多个级别的高速缓存、处理器核心214和寄存器216。示例的处理器核心214可以包括运算逻辑单元(ALU)、浮点数单元(FPU)、数字信号处理核心(DSP核心)或者它们的任何组合。示例的存储器控制器218可以与处理器204一起使用,或者在一些实现中,存储器控制器218可以是处理器204的一个内部部分。

[0034] 取决于期望的配置,系统存储器206可以是任意类型的存储器,包括但不限于:易失性存储器(诸如RAM)、非易失性存储器(诸如ROM、闪存等)或者它们的任何组合。系统存储器206可以包括操作系统220、一个或者多个应用222以及程序数据224。在一些实施方式中,应用222可以布置为在操作系统上由一个或多个处理器204利用程序数据224执行指令。

[0035] 计算设备200还可以包括有助于从各种接口设备(例如,输出设备242、外设接口244和通信设备246)到基本配置202经由总线/接口控制器230的通信的接口总线240。示例的输出设备242包括图形处理单元248和音频处理单元250。它们可以被配置为有助于经由一个或者多个A/V端口252与诸如显示器或者扬声器之类的各种外部设备进行通信。示例外设接口244可以包括串行接口控制器254和并行接口控制器256,它们可以被配置为有助于

经由一个或者多个I/O端口258和诸如输入设备(例如,键盘、鼠标、笔、语音输入设备、触摸输入设备)或者其他外设(例如打印机、扫描仪等)之类的外部设备进行通信。示例的通信设备246可以包括网络控制器260,其可以被布置为便于经由一个或者多个通信端口264与一个或者多个其他计算设备262通过网络通信链路的通信。

[0036] 网络通信链路可以是通信介质的一个示例。通信介质通常可以体现为在诸如载波或者其他传输机制之类的调制数据信号中的计算机可读指令、数据结构、程序模块,并且可以包括任何信息递送介质。“调制数据信号”可以是这样的信号,它的数据集中的一个或者多个或者它的改变可以在信号中编码信息的方式进行。作为非限制性的示例,通信介质可以包括诸如有线网络或者专线网络之类的有线介质,以及诸如声音、射频(RF)、微波、红外(IR)或者其他无线介质在内的各种无线介质。这里使用的术语计算机可读介质可以包括存储介质和通信介质二者。

[0037] 计算设备200可以实现为服务器,例如数据库服务器、应用程序服务器和WEB服务器等,也可以实现为包括桌面计算机和笔记本计算机配置的个人计算机。当然,计算设备200也可以实现为小尺寸便携(或者移动)电子设备的一部分。

[0038] 在根据本发明的实施例中,计算设备200被实现为服务器130,并被配置为执行根据本发明的页面测试方法300或400。其中,计算设备200的应用222中包含执行根据本发明的页面测试方法300或400的多条程序指令,使得本发明的页面测试方法300或400可以在计算设备200中执行。

[0039] 图3示出了根据本发明一个实施例的页面测试方法300的示意图。方法300始于在服务器中执行,服务器可以实现为上文所述的服务器130。如图3所示,方法300始于步骤S310。获取测试机中待测试页面的页面图片。其中,测试机可以实现为上文所述的测试机110。

[0040] 随后,在步骤S320中,将获取到的页面图片输入到元素识别模型中,该元素识别模型将识别出页面图片中的页面元素,并获取到页面图片中页面元素的位置信息。

[0041] 其中,位置信息可以是能够表示出页面元素位置的坐标,也可以是其他的能够确定页面元素位置的信息,例如页面元素相对于某元素的相对位置。

[0042] 根据本发明一个实施例,元素识别模型通过以下方式构建。训练元素识别模型需要大量的标注好的带有页面元素的图片作为训练集,首先,收集带有页面元素的图片素材。图片素材可以通过互联网上提供的带有页面元素的网页图片处进行下载,还可以将本领域技术人员在测试过程中收集到的带有页面元素的图片作为图片素材,本发明对图片素材的获取方式不做限制。

[0043] 在收集好图片素材后,对图片素材中自带的页面元素进行标注。根据本发明实施例,采用数据标注工具对图片素材进行标注。数据标注工具可以选择labelme或者labelImg等数据标注软件,在数据标注工具中对图片素材当中的目标(目标即页面元素)进行选取(选取即截取),选取方式例如有多边形、矩形、圆形、直线、点等等,以矩形为例,采用矩形将图片素材当中包含的目标框选出来。选取目标后对目标进行标注,例如,若目标为按键,则将目标标注为按键,若目标为窗口,则将目标标注为窗口。完成目标的选取和标注后,仅保留图片素材当框选出的目标的区域进行保存,也就是说矩形框以外的内容不进行保留,仅保留带有页面元素的框选区域,这样便排除掉了图片素材当中的无关信息,保留了关键信

息。然后,所选取的区域将被保存成一个文件,该文件可以为JSON格式,文件所保存的选取后的图片即训练元素识别模型所需的特征数据,将全部的图片素材进行上述选取和标注后,得到所有的选取并标注后的图片作为训练集。采用这样的在线图像标注工具,不需要安装或复制大量的数据集,节约存储资源也提升了数据处理的效率。

[0044] 然后,搭建页面元素识别框架来训练上文获得的数据集,框架可以选择TensorFlow或Pytorch开源深度学习框架,待训练的元素识别模型(即预训练模型)可以选择One-stage算法或Two-stage算法对应的预训练模型。One-stage目标检测算法例如:OverFeat、YOLOv1、YOLOv2、YOLOv3、SSD、DSSD和Retina-Net等。采用One-stage算法的模型先进行特征提取,然后直接进行回归定位位置。

[0045] Two-stage目标检测算法例如:R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN和R-FCN等。采用Two-stage算法的模型先进行特征提取,然后对目标做预框选,最后定位目标的位置。由于Two-stage算法相较于One-stage算法会对目标做预框选,因此采用Two-stage算法的模型准确度更高、但处理速度较采用One-stage算法的模型要低。预训练模型可以选择采用One-stage或Two-stage算法中的任一算法的预训练模型,本发明对此不做限制。采用上文获得的数据集对预训练模型进行训练,通过损失函数不断对模型进行优化,最终得到训练好的元素识别模型。将页面图片输入到训练好的元素识别模型中,将得到页面图片中页面元素的位置信息。

[0046] 随后,在步骤S330中,将获取到的页面元素的位置信息写入到测试脚本中,并将测试脚本中的操作指令发送至测试机中的键盘鼠标操作接口,以便测试机执行该操作指令来模拟鼠标和键盘对页面元素的操作。其中,测试脚本包括一个或多个操作指令,操作指令包括页面元素的位置信息。

[0047] 根据本发明的一个实施例,将获取到的页面元素的位置信息作为参数传递给自动化测试脚本,该测试脚本(即自动化测试脚本)通过USB模拟鼠标键盘串口设备,将测试脚本中的操作指令发送给测试机,实现对测试机的待测试页面自动化测试,无需人工操作,实现模拟人工操作鼠标键盘对页面元素进行操作的效果。下面对操作指令进行举例说明:

click 左键单击
right_click 右键单击
double_click 左键双击
drag_to 拖拽到(绝对位置)
drag_relative 拖拽距离(相对位置)
input 输入

通过以上操作指令可以实现鼠标和键盘的一系列基本操作,这里所有的基本操作可以使用Python语言封装接口,支持传参传递。自动化测试执行过程中,将通过元素识别模型获取到的坐标,传递给封装的键鼠操作接口,最后执行测试脚本即可实现在测试机上执行测试用例中的测试步骤。例如,识别到一个元素控件的坐标为(100, 100),根据测试的需要,需要鼠标左键点击该元素控件,则将代码组装为:

Click(100,100)

运行测试脚本,即可实现服务器下发指令到测试机执行点击鼠标左键的动作。

[0048] 随后,在步骤S340中,获取测试机执行操作指令后得到的测试结果,并通过操作指

令的预期结果对测试结果进行验证。

[0049] 可选地,通过断言的方法来判断测试结果和操作指令的预期结果是否一致。例如,在一个测试用户登录的测试用例中,用户点击登录按键(按键是页面元素的一种)后,需要判断新弹出的网页是否与预期的网页一致,则为该测试用例增加一个判断新弹出的网页是否与预期的网页一致的断言。服务器通过智能化分析在判断两个网页是否一致时,若测试结果与预期结果为两个仅窗口大小不同但实质相同的网页,则两个网页将会被识别为两个相同的网页,测试结果将被认定为与预期一致;若测试结果与预期结果为不同样式但功能相同的两个按键,则测试结果将被认定为与预期一致,例如,音乐播放按键的样式可以为三角形、圆形或表示播放的任意样式,不同样式的功能为播放的按键都可以被认定为播放按键,智能化分析可以将不同样式但实质相同的按键认定为相同的按键,提升了断言的灵活性。

[0050] 根据本发明的实施例,服务器将测试结果与操作指令的预期结果进行比对,若测试结果与预期结果一致,则验证通过;若测试结果与预期结果不一致,则执行中断处理并输出异常结果。然后,将测试脚本中的下一个操作指令发送至测试机中执行,直至测试脚本中的全部操作指令执行完毕。

[0051] 接下来介绍本发明另一个实施例的页面测试方法。图4示出了根据本发明另一个实施例的页面测试方法400的示意图。如图4所示,方法400始于步骤S410,视频采集装置实时采集测试机中待测试页面的页面视频,然后将页面视频传输至服务器中。其中,页面视频的格式可以设置为WebM格式,页面视频可以实时录制并实时读取。

[0052] 随后,在步骤S420中,服务器在接收到来自视频采集装置的页面视频后,识别页面视频中的页面元素。根据一种实施例,由于页面视频包括测试机中待测试页面的每一帧的页面图片,若需要对页面视频中的页面元素进行识别,则将实时采集到的页面视频中待测画面对应的页面图片输入到元素识别模型中。元素识别模型可以通过上文步骤S320中所述的方式构建,在此不做赘述。

[0053] 随后,在步骤S430中,在服务器中判断是否识别到页面元素。若通过元素识别模型未识别到页面元素,代表当前实时获取的页面视频中不存在页面元素,则方法400执行结束。若通过元素识别模型识别到了页面元素,则执行步骤S440,元素识别模型返回元素的位置信息。其中,位置信息可以是能够表示出页面元素位置的坐标,也可以是其他的能够确定页面元素位置的信息,例如页面元素相对于某元素的相对位置。

[0054] 随后,进入步骤S450,判断测试用例是否结束。根据本发明的一个实施例,本领域技术人员在进行页面测试时,通常使用多个测试用例对页面进行测试。这里判断是否全部的测试用例已经测试完成,若全部测试用例测试完成,代表测试任务执行完毕,则方法400执行结束。

[0055] 若测试用例未全部测试完成,则执行步骤S460,服务器将获得的页面元素的位置信息写入到测试脚本中,并下发下一个操作指令至测试机中执行。一个测试用例对应一个或多个测试脚本,其中,一个测试脚本包括一个或多个操作指令,在测试用例未全部测试完成的情况下,服务器发送下一个该执行的操作指令至测试机,若一个测试用例中其中一个测试脚本操作指令已经全部下发过,则下发将该测试用例的下一个测试脚本的操作指令,若该测试用例中全部测试脚本已经执行过了(即该测试用例的全部测试脚本中的操作指令

皆已经下发至测试机中执行测试完成),则测试下一个测试用例,直到全部的测试用例测试完成。

[0056] 在这一步骤中,服务器将获得的页面元素的位置信息写入到测试脚本中,具体写入方式可以通过步骤S330中所述的页面元素位置信息写入测试脚本的方式实现,在此不做赘述。

[0057] 随后,在步骤S470中,测试机接收到服务器下发的操作指令后,执行操作指令,生成测试结果,并将测试结果发送至服务器中。

[0058] 随后,在步骤S480中,在服务器接收到测试机发送的测试结果后,将测试结果与测试机执行的操作指令所对应的预期结果进行比对,具体的比对方式可以通过步骤S340中所述的通过操作指令的预期结果对测试结果进行验证的方式实现,在此不做赘述。执行完步骤S480后,进入步骤S450,判断测试用例是否全部测试完成,若否,则继续进入步骤S460,下发下一个操作指令以进行下一步的测试。若是,表示全部的测试用例测试完毕,则方法400执行完毕。

[0059] 根据本发明的技术方案,通过元素识别模型对实时采集的视频进行识别,确定元素控件的位置信息,通过自动化测试脚本封装键盘鼠标的操作对测试集执行自动化测试,并通过服务器执行智能化分析,确定测试结果是否与预期一致,若一致,则继续执行自动化测试用例。本方案采用的智能识别模型解决了传统UI自动化测试中元素控件定位不准确、断言不稳定不准确的问题,提升了UI自动化测试中元素控件识别的准确性和断言的灵活性。

[0060] 进一步地,本方案中服务器、测试机以及视频采集装置设备分离,数据处理的部分由服务器来处理,处理速度非常快,解决了传统UI自动化测试中定位元素慢的问题,以及测试用例执行慢的问题,提升了自动化测试的处理和执行速度。

[0061] 并且,本方案还解决了UI自动化测试中,当需求迭代或环境因素所导致的UI变化,进而导致测试用例失败的健壮性问题,降低了自动化测试用例代码及测试资源的维护成本。

[0062] 这里描述的各种技术可结合硬件或软件,或者它们的组合一起实现。从而,本发明的方法和设备,或者本发明的方法和设备的某些方面或部分可采取嵌入有形媒介,例如可移动硬盘、U盘、软盘、CD-ROM或者其它任意机器可读的存储介质中的程序代码(即指令)的形式,其中当程序被载入诸如计算机之类的机器,并被所述机器执行时,所述机器变成实践本发明的设备。

[0063] 在程序代码在可编程计算机上执行的情况下,计算设备一般包括处理器、处理器可读的存储介质(包括易失性和非易失性存储器和/或存储元件),至少一个输入装置,和至少一个输出装置。其中,存储器被配置用于存储程序代码;处理器被配置用于根据该存储器中存储的所述程序代码中的指令,执行本发明的页面测试方法。

[0064] 以示例而非限制的方式,可读介质包括可读存储介质和通信介质。可读存储介质存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息。通信介质一般以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据,并且包括任何信息传递介质。以上的任一种的组合也包括在可读介质的范围之内。

[0065] 在此处所提供的说明书中,算法和显示不与任何特定计算机、虚拟系统或者其它

设备固有相关。各种通用系统也可以与本发明的示例一起使用。根据上面的描述,构造这类系统所要求的结构是显而易见的。此外,本发明也不针对任何特定编程语言。应当明白,可以利用各种编程语言实现在此描述的本发明的内容,并且上面对特定语言所做的描述是为了披露本发明的优选实施方式。

[0066] 在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施例可以在没有这些具体细节的情况下被实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

[0067] 类似地,应当理解,为了精简本公开并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。

[0068] 本领域那些技术人员应当理解在本文所公开的示例中的设备的模块或单元或组件可以布置在如该实施例中所描述的设备中,或者可替换地可以定位在与该示例中的设备不同的一个或多个设备中。前述示例中的模块可以组合为一个模块或者此外可以分成多个子模块。

[0069] 本领域那些技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件,以及此外可以把它分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外,可以采用任何组合对本说明书中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述,本说明书中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0070] 此外,本领域的技术人员能够理解,尽管在此所述的一些实施例包括其它实施例中包括的某些特征而不是其它特征,但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。

[0071] 此外,所述实施例中的一些在此被描述成可以由计算机系统的处理器或者由执行所述功能的其它装置实施的方法或方法元素的组合。因此,具有用于实施所述方法或方法元素的必要指令的处理器形成用于实施该方法或方法元素的装置。此外,装置实施例的在此所述的元素是如下装置的例子:该装置用于实施由为了实施该发明的目的的元素所执行的功能。

[0072] 如在此所使用的那样,除非另行规定,使用序数词“第一”、“第二”、“第三”等等来描述普通对象仅仅表示涉及类似对象的不同实例,并且并不意图暗示这样被描述的对象必须具有时间上、空间上、排序方面或者以任意其它方式的给定顺序。

[0073] 尽管根据有限数量的实施例描述了本发明,但是受益于上面的描述,本技术领域内的技术人员明白,在由此描述的本发明的范围内,可以设想其它实施例。此外,应当注意,本说明书中使用的语言主要是为了可读性和教导的目的而选择的,而不是为了解释或者限定本发明的主题而选择的。

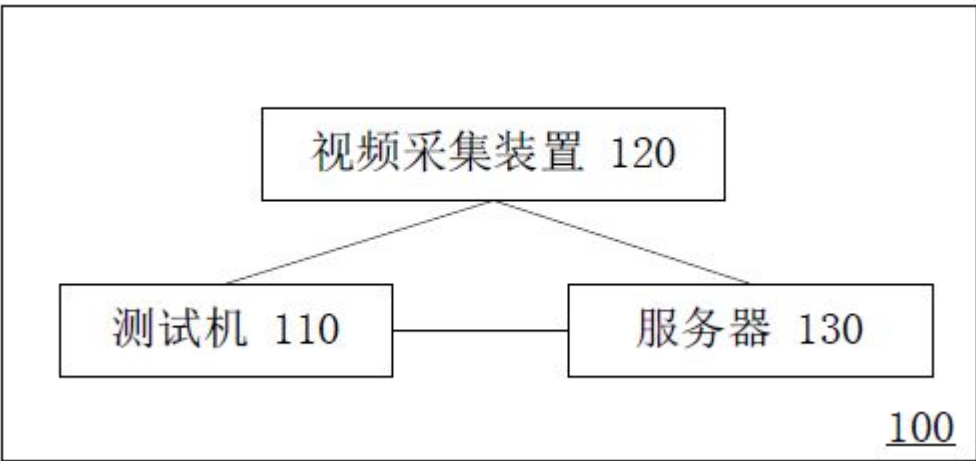


图1

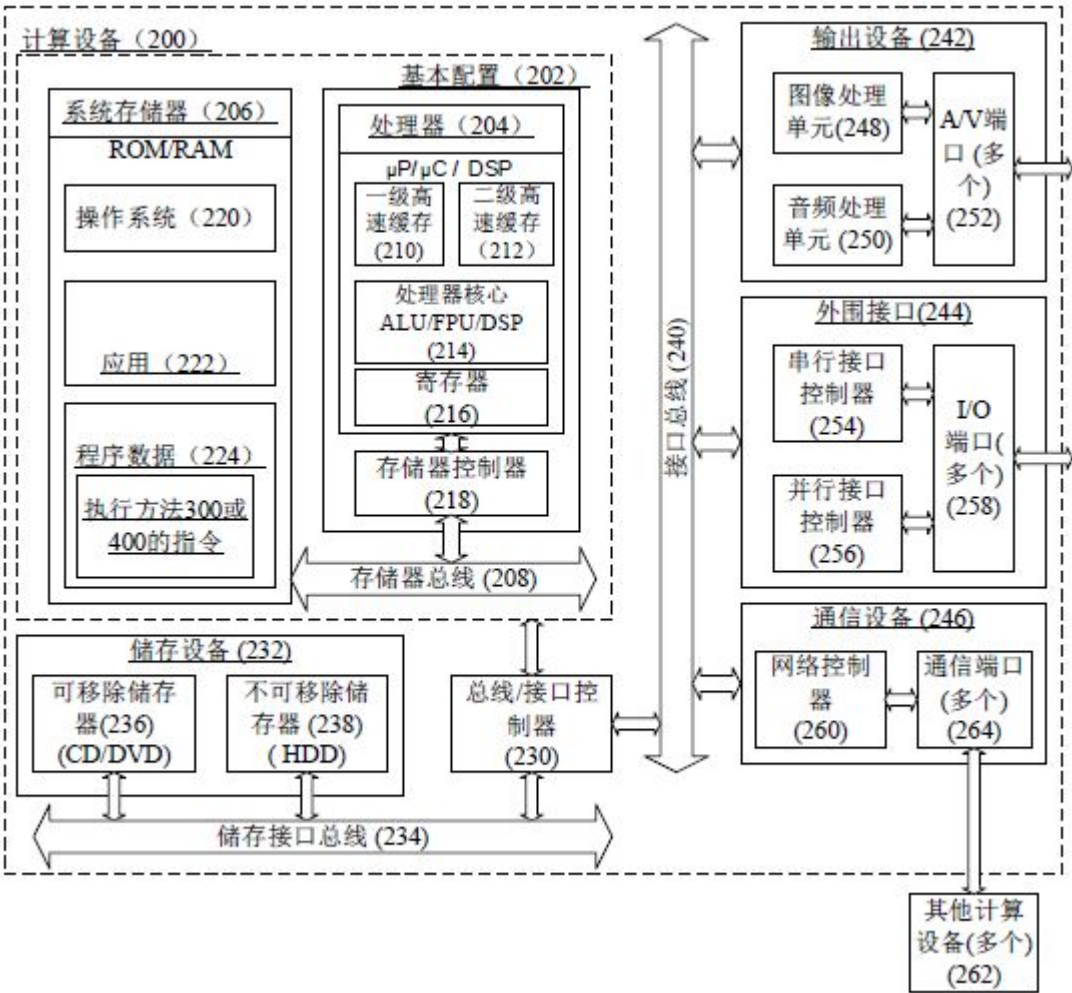


图2

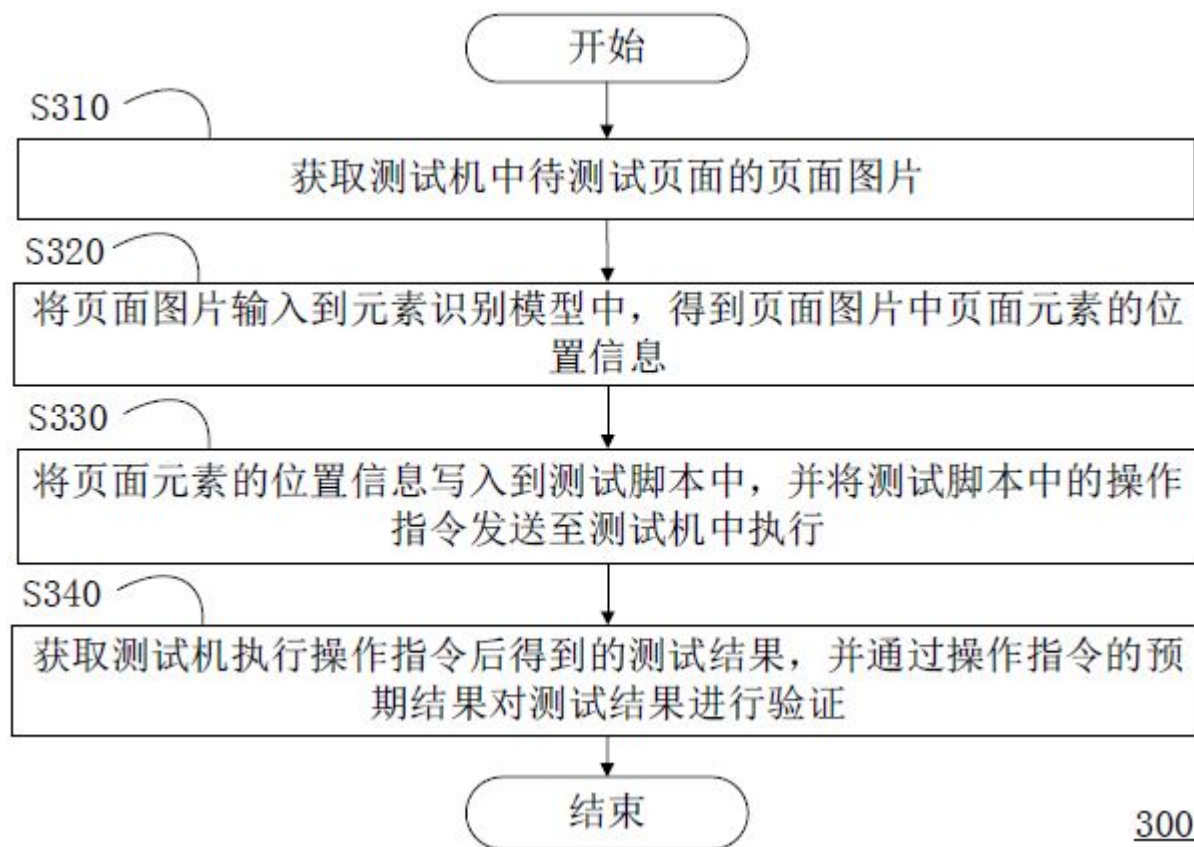


图3

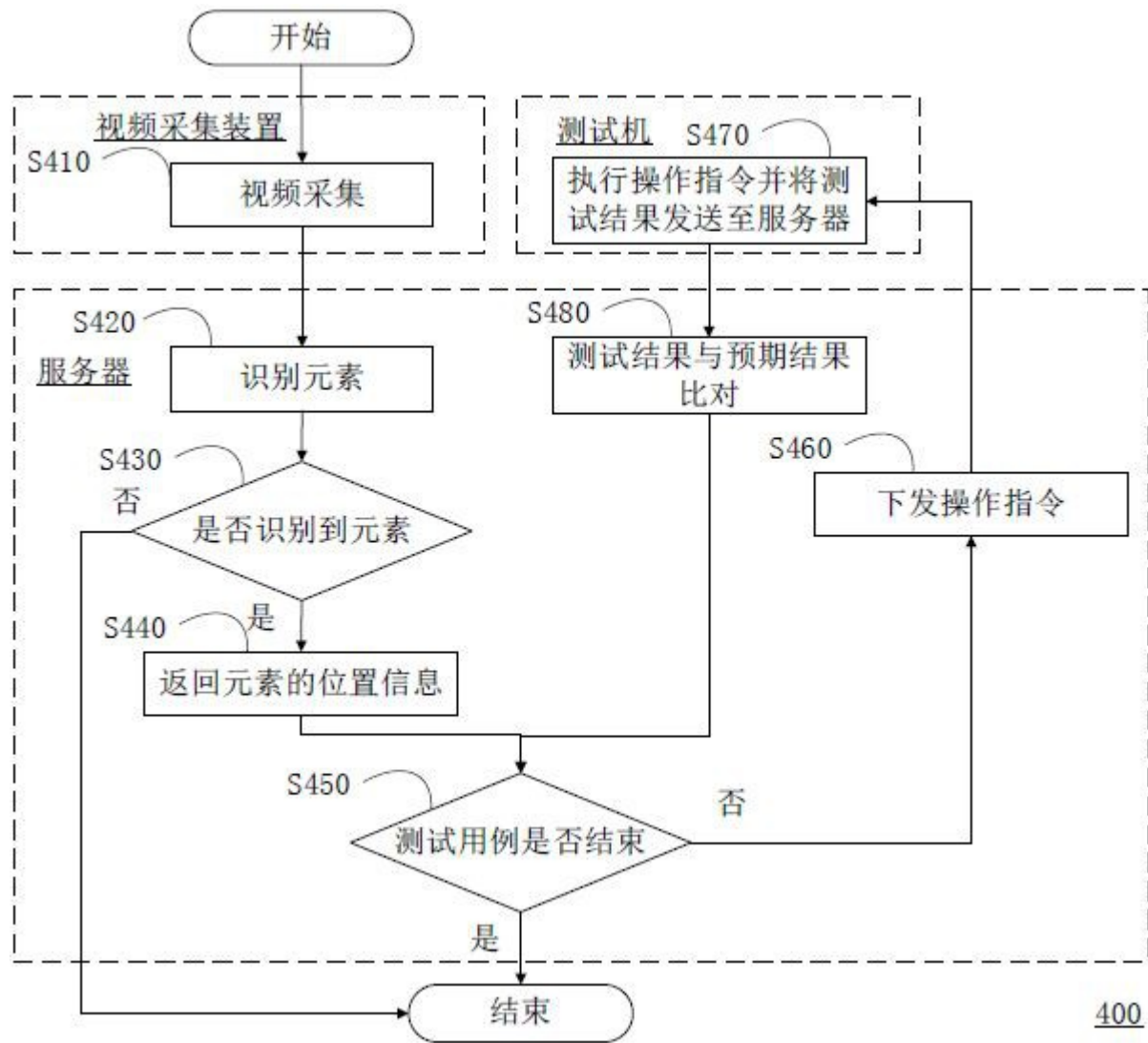


图4