

基于深度排序学习的脚本录制 动作参数获取方法

王鹏宇[†], 钱巨

(南京航空航天大学计算机科学与技术学院 江苏 南京 210016)

摘 要:在 GUI 可视化测试脚本中一般将目标控件图像作为点击动作的参数。在单个摄像机录制环境下构建 GUI 可视化测试脚本时,存在点击动作的目标控件图像获取准确率低的问题。为此,提出了一种基于深度排序学习的点击动作参数获取方法。在本方法中,利用计算机视觉技术来提取视频中指尖移动和屏幕变化等特征以获取更准确的用户与设备交互的触控区域,将该区域内包含的控件作为排序对象,使用深度排序学习算法对这些候选控件进行评分和排序,确保最符合预期的控件被选中作为点击目标控件。实验结果表明,使用该方法在单视频中获取点击动作参数具有较高的准确率。

关键词:GUI 测试;可视化脚本;目标控件定位;计算机视觉;深度学习;排序学习

中图分类号:TP311

文献标识码:A

Script Recording Method for Action Parameter Acquisition Based on Deep Learning and Learning to Rank

WANG Pengyu[†], QIAN Ju

(College of Computer Science and Technology, Nanjing University of Aeronautics
and Astronautics, Nanjing, Jiangsu 210016, China)

Abstract: In GUI visualization test scripts, the target control image is typically used as a parameter for the click action. When creating GUI visualization test scripts in a single-camera recording environment, there is a challenge in achieving high accuracy when capturing target control images for click actions. A method for acquiring click action parameters based on deep learning and learning to rank algorithms is proposed to address the aforementioned issues. This method utilizes computer vision technology to extract features such as fingertip movement and screen changes in videos, enabling more accurate determination of touch areas for user device interaction. The controls in this area are used for sorting objects, and deep learning and learning to rank algorithms are employed to score and arrange these candidate controls to ensure that the most relevant control is selected as the target for clicking. The experimental results demonstrate that utilizing this method to acquire click action parameters in single-shot videos yields high accuracy.

Key words: GUI testing; visual script; target control location; computer vision; deep learning; learning to rank

GUI 测试^[1]对于确保软件界面的正确性、功能的可靠性以及用户体验的良好性起着关键作用。自动化的 GUI 测试可以显著提高测试效率,同时

减少人力资源和时间投入,使得软件测试过程更加高效和可靠。

脚本自动录制是实现自动化 GUI 测试^[2]的关

键步骤。在这个过程中,自动化测试工具可以将人工测试视频转化为可视化测试脚本,这些脚本与视频中的操作一一对应。然后,可以根据这些脚本按照指令顺序执行,模拟人类进行测试的过程。如果脚本录制时出现错误,就会直接影响测试结果的准确性。在脚本录制过程中,获取点击动作的参数是其中一个较为困难的方面,尤其是在基于摄像头的非侵入式脚本录制中,如 RoScript^[4] 方法和 RoScript3D^[5] 方法。因为需要在视频中精确识别和定位点击目标控件,这对准确度提出了较高的要求。

RoScript 方法中,对测试人员设定特定的操作要求,将视频中最高的指尖坐标作为点击位置,并获取目标控件的图像。而 RoScript3D 方法则利用额外的深度相机来辅助获取手与设备的接触情况,取接触时的指尖坐标作为点击位置,并获取目标控件的图像。

上述点击动作参数获取的方法存在两个主要问题:一是使用额外相机辅助获取点击位置,增加了录制复杂程度;二是设定一定操作要求,仅使用最高指尖坐标来获取目标控件图像,准确率较低。若测试人员在录制过程中指尖稍微超出点击控件区域,就会导致参数获取失败。

针对以上问题,提出了一种基于深度排序学习的点击动作参数获取方法,在单个摄像机拍摄环境中,利用计算机视觉技术来提取视频中指尖移动和屏幕变化等特征以获取用户与设备交互的触控区域,使用深度排序学习算法对触控区域内若干个控件进行评分和排序,最终将获取的控件图像作为点击动作的参数。

通过利用指尖移动和屏幕变化特征,可以更准确地确定用户与设备交互的触控区域。深度排序学习^[6] 算法能够对触控区域内的若干个控件进行特征提取并排序,确保最符合预期的控件被选中作为目标控件。这种利用深度排序学习的智能化方法能够有效地提高点击动作参数的获取准确度,从而提升自动化 GUI 测试的准确性和可靠性。

本文的贡献如下:

(1) 分析总结了现有方法在视频中获取点击动作参数所存在的问题,并确定了改进方案。

(2) 提供了一种准确、可靠的方法来获取点击动作的参数,为 GUI 测试的自动化提供更准确的脚本录制方法。

1 问题定义与总体思路

非侵入式 GUI 测试脚本录制方法是将一个人工 GUI 测试视频转换为对应的可视化测试脚本,其主要分为以下几个步骤:

(1) 动作分组:将整个测试视频分割为多个独立的动作视频段;

(2) 交互动作识别:从分割后的视频中准确地识别每个动作的类型,例如点击、滑动或输入等;

(3) 动作参数确定:针对不同类型的动作,获取相应的参数信息。例如,对于点击动作,获取目标控件图像等参数。

然而,本研究专注于改进点击动作参数获取的准确性,希望提供一种准确的、可靠的方法来获取点击动作的参数,从而为 GUI 测试的自动化提供更准确的脚本录制方法。

由于测试人员执行点击动作时可能存在不精确的情况,使得准确捕捉点击位置变得困难,导致无法获取有效的目标控件图像。这是因为人的指腹和指尖之间存在一定的弧形距离,通常人们使用指腹与设备进行接触,而指尖位置通常会高于实际与设备接触的位置,高出的距离是由于手指的形状以及手指与触摸屏的夹角所决定的。通过对实验室 16 位参与者的数据进行分析,发现指尖和指腹与屏幕接触位置的距离大约在 2 mm 到 6 mm 之间。在图 1(a) 中,展示了一个指尖与指腹的水平方向距离为 4 mm 的案例。

在测试人员未刻意调整点击动作的情况下,如果使用检测到的最高指尖坐标作为点击位置,通常会超出目标控件范围,位于两个控件之间,导致无法获取有效的目标控件图像,如图 1(b) 所示。

根据图 1 可得知,当人与设备接触时的点击位置是一个区域,无法仅用单个坐标来描述点击位置。因此,本文的研究方法的总体思路是通过视觉技术获取更准确的人机交互触控区域,并对处于该区域内的控件进行分析,使用深度排序学习算法对控件进行评分和排序,取评分最高的控件图像作为表达实际动作目标的控件图像。这样可以更准确、更可靠地捕捉到用户的点击对象。该方法的总体流程如图 2 所示。如果触控区域中包含一个或多个控件,则使用深度排序学习算法来根据控件图像与响应界面的关联度进行排序。深度排序学习算

法会根据大量的数据学习每个控件图像与响应界面的匹配程度,为候选控件图像打分。最终选择评分最高的图像作为表达实际动作目标的控件图像。

然而,触控区域中不包含任何目标控件,说明控件检测算法可能失效。通过将触控区域作为目标控件图像,可以将其用作触发相应的点击事件的依据,使得程序可以在缺少具体控件信息的情况下完成相应的点击参数的获取。

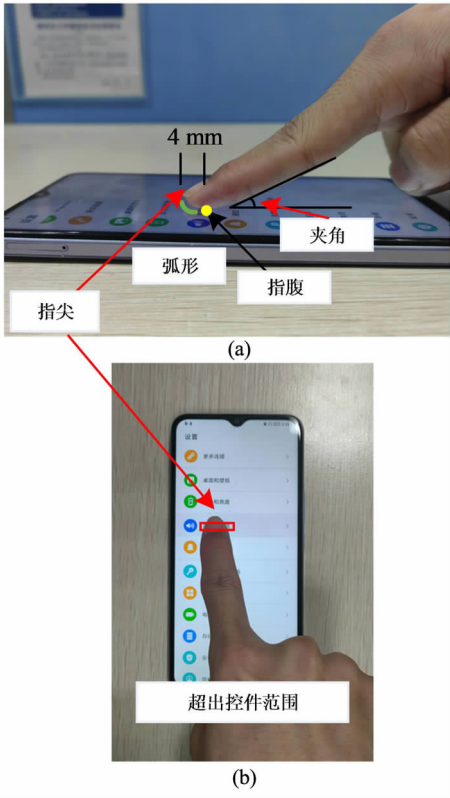


图 1 手指特征及案例失败示意图

2 控件深度排序学习算法

使用视觉技术来提取视频中指尖移动和屏幕变化等特征获取用户与设备交互的触控区域,根据控件位置信息取触控区域内包含的所有控件作为排序候选控件,采用深度排序学习算法对这些候选控件进行评分和排序,将最符合预期的控件图像作为表达实际动作目标的控件图像,实现点击动作参数的获取。

2.1 获取排序候选控件

对点击动作触发时进行特征分析。一般用户的点击动作持续时间在 0.5~1 s 之间,在此期间手指位置保持不变,指尖移动速度为零,并且屏幕

内容未发生变化。点击触发后,系统做出响应,屏幕进而发生变化。

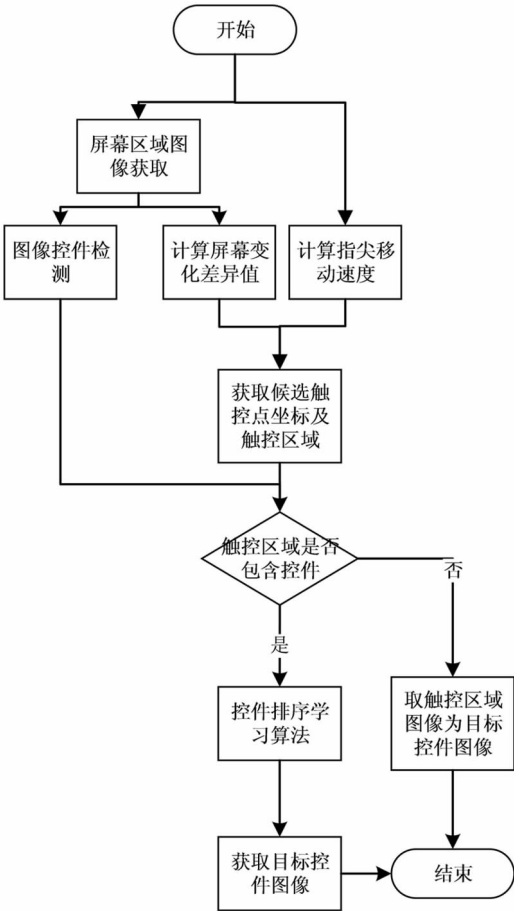


图 2 点击动作参数获取总体流程图

根据上述描述的特征,可以进行以下步骤来获取触控点坐标和触控区域:首先,从视频序列中提取出在屏幕内容发生变化之前指尖位置相对保持不变(即指尖移动速度 $v_{finger} \leq T_{still}$)的视频帧。设定阈值 T_{still} 可以抑制手指轻微抖动或指尖检测不准确等因素对指尖位置的微小变化的影响,减少噪声和抖动带来的干扰,将注意力集中在真正有意义的指尖位置变化上,从而提高交互的稳定性和可靠性。

然后,从这些图像中获取指尖坐标 (x, y) ,对该数据进行处理。在指尖坐标的 y 轴分量上减去指尖与指腹的偏差 T_d ,即 $y = y - T_d$,以得到与屏幕接触的指腹坐标。接下来,根据调整后得到的指腹数据,计算其中值,作为初步的触控点坐标。

本文以触控点为中心,构建一个一定大小的正方形区域,作为手指与屏幕的触控区域。根据控件

位置信息取触控区域内包含的所有控件作为排序候选控件。使用候选控件图像和响应界面的图像作为控件深度排序学习模型的输入,经模型计算后,将评分最高的控件图像作为点击动作的参数。

下面将介绍指尖移动速度的计算、屏幕变化特征的提取以及图像控件的检测。

(1) 计算指尖移动速度

本文采用基于 YCrCb 颜色空间的肤色^[7]检测方法来获取手部轮廓,取轮廓最高点作为指尖位置,根据指尖位置的变化进而计算指尖移动速度。为了处理手部区域中局部位置颜色的差异,并使每个位置都具有一定的容忍度,设置肤色矩阵区间为:

$$\text{color_range}_{\text{skin}} = \left\{ \begin{array}{l} (Y_{\text{low}}, Y_{\text{high}}) \\ (Cr_{\text{low}}, Cr_{\text{high}}) \\ (Cb_{\text{low}}, Cb_{\text{high}}) \end{array} \right\} \quad (1)$$

对于图像中的每个像素点,如果其 Y、Cr 和 Cb 分量的颜色值都在 $\text{color_range}_{\text{skin}}$ 区间范围内,那么该像素点被视为皮肤检测的候选像素点,将该像素点设置为白色,否则将该像素点设置为黑色。通过对该二值图像进行轮廓检测得到手部轮廓图。取该图中最高的顶点坐标作为指尖坐标。计算相邻两帧图像中指尖坐标的距离 Δd ,根据该测试视频的帧率 fps,计算指尖的移动速度 v_{finger} ,具体计算公式为:

$$v_{\text{finger}} = \Delta d \cdot \text{fps} \quad (2)$$

(2) 屏幕变化特征提取

首先,使用基于边缘检测与轮廓提取算法获取视频中屏幕区域的四个顶点坐标。接着,为了校正由于拍摄角度而引起的图像中屏幕区域的畸变,利用透视变化^[8]算法获取屏幕区域正视图,从而更准确地呈现屏幕图像内容,如图 3 所示。最后用屏幕变化差异值来描述屏幕变化特征。计算透视变换对应坐标,进而得到屏幕区域正视图,从而更准确地呈现图像内容。透视变换坐标计算公式为:

$$[x', y', w] = [x, y, 1] \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \quad (3)$$

$$u = \frac{x'}{w}, v = \frac{y'}{w} \quad (4)$$

其中, x 、 y 为原始图像坐标, u 、 v 为经透视变换后在新图像中的坐标。

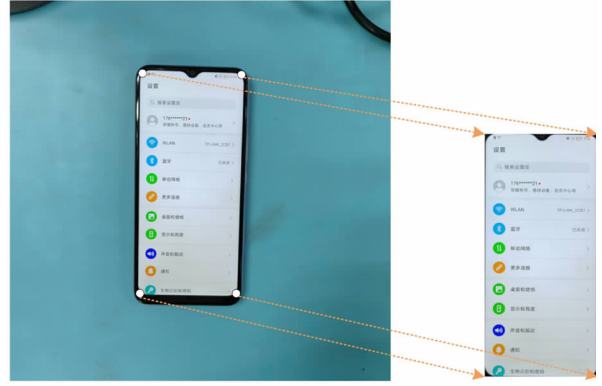


图 3 获取屏幕区域正视图过程示意图

本文采用了对比相邻两帧屏幕区域正视图的结构相似度 SSIM^[9] 的补作为两张图像的差异值 diff,用差异值来描述屏幕变化特征。相较于其他相似度计算方法,SSIM 考虑了图像的结构信息,而不仅仅是像素值的相似度。它通过比较图像的亮度、对比度和结构来评估图像的相似度,更符合人眼对图像质量的感知。通过采用 SSIM,可以更准确地衡量图像之间的相似程度,对 SSIM 取补从而更好地理解并描述屏幕区域的变化特征。diff 的具体计算公式如下所示。

$$\text{diff} = 1 - f(l(i, j), c(i, j), s(i, j)) \quad (5)$$

其中, l 、 c 、 s 分别表示亮度相似度、对比度相似度以及结构相似度。

(3) 图像控件检测

本文利用 Canny 边缘检测^[10]算法对图像中控件进行检测。在进行边缘检测之前,首先对屏幕区域图像进行预处理,去除图像中噪声以及增强图像中的边缘特征,可以提高后续边缘检测的准确性。然后进行边缘检测和轮廓提取。因已经得到屏幕区域的正视图,控件的外接矩形一般呈现为与屏幕边界相对平行的规则矩形,所以最终取各轮廓的外接矩形作为整体控件。

2.2 控件排序特征分析

从应用程序层面考虑,可以通过根据颜色和局部匹配特征等信息对控件进行评分,进而对控件进行排序。

控件图像很可能与响应界面图像采用相同的色彩主题。图 4 中展示的两个应用的应用图标与响应界面的颜色主题一致,分别为红和蓝。



图 4 应用图标与响应界面展示图

对于非颜色的图标控件,比如文字控件,响应界面中很可能包含该文字信息。图 5 展示了文字控件与响应界面之间的关联性。



图 5 文字控件与响应界面关联示意图

然而,深度学习模型具有提取更多特征的能力,因此可以更全面地找到控件和响应界面之间的关联性并进行排序。

2.3 控件排序算法实现

本文采用深度排序学习算法对一个或多个候选控件进行评分和排序,取最高评分控件的图像作为点击动作参数。

该模型以候选控件图像与响应界面图像组合编码的 $224 \times 224 \times 6$ 图像矩阵作为输入。其中,先以候选控件图像为中心填充,使其与响应界面图像具有相同的大小。填充部分使用黑色(0 值)进行填充。然后,将填充后的图像与响应界面图像进行拼接,并经过 `resize` 处理,完成编码过程。

接着,使用深度学习网络模型 ResNet-50^[11]对图像进行特征提取得到特征向量。对于这种选择唯一正确控件的任务,采用 Pointwise 排序学习方式将每个候选控件作为独立的样本进行处理,并为其分配一个分数来表示其是否为目标控件。分数为 1 表示为目标控件,分数为 0 表示非目标控件。通过训练一个二分类模型,可以为每个候选控件预测一个分数,然后选择分数最高的候选控件作为目标控件。图 6 展示了控件排序算法的处理流程。

本文采用二元交叉熵(Binary Cross-Entropy, BCE)作为损失函数。BCE 损失函数用于衡量模型预测结果与真实标签之间的差异。BCE 损失函数的计算公式如下:

$$L = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \lg(p(y_i)) + (1 - y_i) \cdot \lg(1 - p(y_i)) \tag{6}$$

其中, y_i 是第 i 个候选控件的二元标签,取 1 或者 0。 $p(y_i)$ 是输出属于标签的概率。 N 表示模型预测控件图像的组数。

3 实验分析

3.1 实验设置

本实验分析指尖与指腹偏差参数和触控区域面积大小参数对实验结果的影响,并确定实验条件中最佳的组合。首先,本文设定了三个不同的指尖与指腹偏差数值:2 mm、4 mm 和 6 mm,并通过实验来确定能够产生最佳实验结果的偏差参数。其次,考虑到移动触屏的手指触发尺寸^[12]通常约为 1 cm²,并通过观察用户与设备的交互过程,发现人机交互的接触面积会随着按压力度的增加而变大。故本文设置了两种不同尺寸的触控区域:边长为 1 cm 和 1.2 cm 的正方形区域。通过比较不同区域大小对实验结果的影响,来确定最适合的触控区域大小。

3.2 实验对象

使用本文提出的点击动作参数获取方法对安卓系统中设置、微信、京东等 6 个应用程序的人工测试视频进行测试,共涉及 187 个点击动作。

3.3 实验指标

以点击动作参数获取准确率为评价指标。如果点击程序计算得到的控件图像与点击真实目标控件的响应一致,则认为获取成功,否则获取失败。

点击动作参数获取准确率的计算公式如下：

准确率 = $\frac{\text{成功案例数量}}{\text{总案例数量}}$

(7)

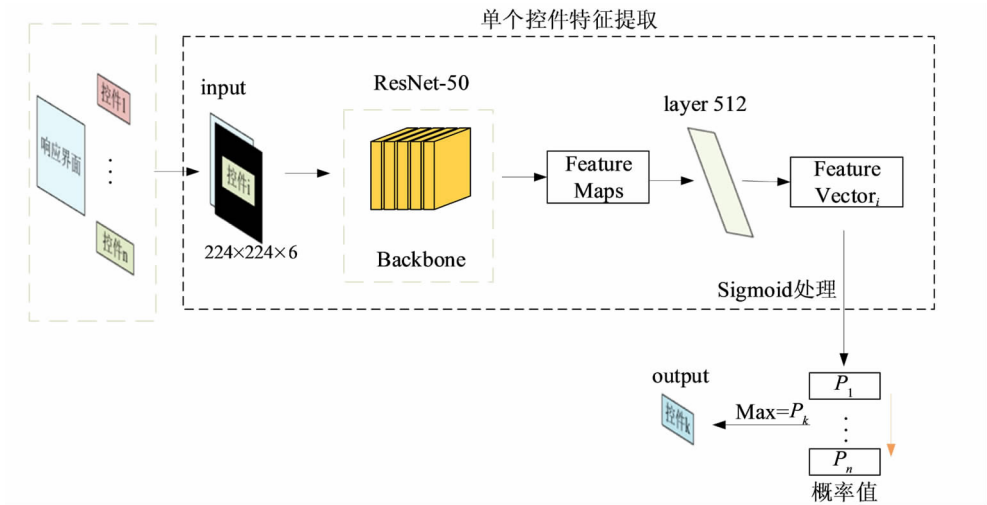


图 6 控件排序算法的处理示意图

3.4 实验结果与分析

根据实验设定的不同参数,使用本文提出的方法对上述视频进行点击动作参数获取,结果如表 1 所示。

表 1 不同参数的实验结果

偏差参数/ mm	触控区域 边长/cm	准确率
2	1	80.2%
4	1	90.4%
6	1	85.6%
2	1.2	77.0%
4	1.2	88.2%
6	1.2	81.3%

根据上表可知,在偏差参数为 4 mm 和触控区域边长为 1 cm 的组合实验条件下,点击动作参数获取的准确率最高。当偏差参数设置过小或过大时,程序计算得到的指腹触控位置可能偏离目标控件的中心,进而使得获取的触控区域不够准确。此外,如果使用较大的触控区域来获取控件,可能会导致触控区域内包含多个控件的案例增加或包含更多的其他非目标控件,增加了控件排序算法的处理难度,进一步降低了点击参数获取的准确率。因此,为了获得最佳结果,需要选择合适的偏差参数和触控区域大小参数。

对在偏差参数为 4 mm 和触控区域边长为 1 cm 的组合实验条件下的实验结果进行分析,发现在程序生成的控件图像中,与真实点击目标控件的响应完全一致的有 169 个,准确率为 90.4%。根据程序计算手指与屏幕的接触区域,在 176 个案例中,接触区域包含一个或多个控件,通过控件深度排序算法选择的控件可表达实际目标控件的准确率为 91.5%。其中,在 141 个案例中,接触区域仅包含一个控件,参数获取准确率为 93.6%;在 35 个案例中,接触区域包含了多个控件,参数获取准确率为 82.9%。该实验数据表明,在控件检测正确的情况下,本文所提出的方法在选择点击动作的目标控件方面非常可靠,能够达到较高的准确率。

此外还有 11 个案例,由于控件检测算法失效,将触控区域直接取为目标控件图像,这种情况下的准确率为 72.7%。图 7 展示了控件检测失败的情况,根据触控区域得到仅包含一个控件的图像可以成功触发点击动作,得到与点击真实目标控件一样的响应。但是,若根据触控区域得到的图像包含多个控件,则无法根据该图像判断真正的目标控件。

总体来说,本文使用基于深度排序学习的点击动作参数获取方法来获取表达实际动作目标的控件图像具有较高的准确率。在单个摄像机的录制环境中进行自动化 GUI 测试脚本录制,具有良好的应用前景。

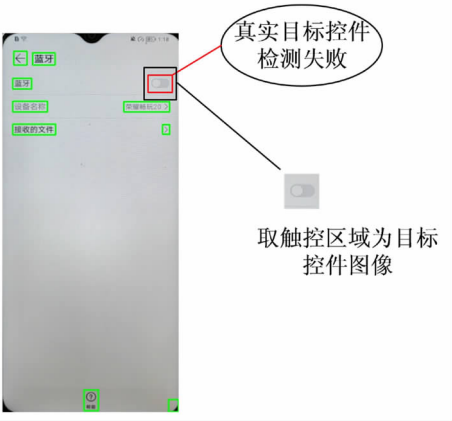


图 7 控件检测失效时处理示意图

4 结 论

本文提出了一种基于深度排序学习的脚本录制动作参数获取方法。该方法适用于从单摄视频中提取测试人员与设备交互的点击目标控件图像。在安卓手机系统中多个应用程序的实验研究表明，该方法获取点击动作参数具有较高的准确率，为 GUI 测试的自动化提供更准确的脚本录制方法。

参考文献

[1] LIN J W, MALEK S. Gui test transfer from web to android [C]//2022 IEEE Conference on Software Testing, Verification and Validation (ICST). IEEE, 2022: 1—11.

[2] RODRÍGUEZ-VALDÉS O, VOS T E J, AHO P, et al. 30 years of automated GUI testing: a bibliometric analysis[C]//Quality of Information and Communications Technology: 14th International Conference. QUATIC, 2021: 473—488.

[3] NASS M, ALÉGROTH E, FELDT R. Why many challenges with GUI test automation (will) remain[J]. Information and Software Technology, 2021, 138: 106625.

[4] QIAN J, SHANG Z, YAN S, et al. Roscript: a visual script driven truly non-intrusive robotic testing system for touch screen applications[C]//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering,2020: 297—308.

[5] 孟祥臻,钱巨. 基于深度相机的三维空间测试脚本录制方法与工具[J]. 计算技术与自动化,2022,41(02):141—146.

[6] LI H. Learning to rank for information retrieval and natural language processing[M]. Springer Nature, 2022.

[7] VANS H V, CHANDRASEKHAR K, ANIL C R, et al. Improved face detection using YCbCr and Adaboost[C]//Computational Intelligence in Data Mining: Proceedings of the International Conference on ICCIDM 2018. Springer Singapore, 2020: 689—699.

[8] 戴雯惠,樊凌. 基于改进透视变换的畸变图像校正方法研究[J]. 信息通信, 2020, 11: 63—65.

[9] NILSSON J, AKENINE-MÖLLER T. Understanding ssim [J]. arXiv preprint arXiv:2006.13846, 2020.

[10] SEKEHRVANI E A, BABULAK E, MASOODI M. Implementing canny edge detection algorithm for noisy image [J]. Bulletin of Electrical Engineering and Informatics, 2020, 9(4): 1404—1410.

[11] KOONCE B. ResNet 50 [M]//Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization California;Apress Berkely,CA, 2021: 63—72.

[12] ANTHONY T. Finger—friendly design: ideal mobile touch-screen target sizes[J]. Accessed,2012 ,27(4):2014.