LOGO

工具复现

论文总结汇报

Wireframe-Based UI Design Search Through Image Autoencoder

汇报人：李若璇

时间：11-30

LOGO

目 录

CONTENTS

01 论文背景概述

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer dolor quam, pretium eu placerat eu semper et nunc. Nullam ut turpis dictum luctus mi quis luctus lorem.

ONE

UI设计的重要性

- 一个设计良好的GUI可以使应用程序简单、实用和高效地使用，这将显著影响应用程序的成功及其用户的忠诚度。例如，良好的GUI设计可以
  - 区分应用程序与竞争对手
  - 吸引用户下载
  - 减少用户的投诉
  - 留住关键用户

ONE

UI设计的困难

- 为了满足用户的需求，一个好的GUI不仅需要设计的具体知识，还需要设计师对设计空间的理解。然而，由于UI设计师的短缺，对UI设计空间不太了解的软件开发者，往往不得不在软件开发中扮演设计师的角色。所以需要一种有效的机制来支持此类开发人员在其UI设计工作中探索和了解UI设计空间。
- 所以为开发者提供一个UI设计搜索引擎来搜索现有的UI设计，可以帮助开发者快速建立对GUI设计空间的真实理解，并从现有应用中获取灵感，用于自己应用的UI设计。

ONE

现有UI设计搜索引擎的不足

- 现有的UI设计搜索方法大致分为:
  - 基于描述软件功能的关键字搜索
  - 基于UI设计模式的关键字搜索
  - 基于GUI组件的关键字搜索
- 但是一些关键字很难描述所需UI设计的视觉语义，例如使用的视觉组件及其布局。
- 还有迄今为止最大的移动UI数据集——Rico，创建目的是支持五类数据驱动的应用程序：设计搜索，UI布局生成，UI代码生成，用户交互建模和用户感知预测。但是Rico评估仅显示了几个示例，没有对检索准确性、数据问题、模型限制、失败案例和有用性评估进行任何详细研究。

ONE

作者在文章中解决的问题

- 作者提出了一种新的基于深度学习的方法，以无监督的方式使用卷积神经网络来构建UI设计搜索引擎，该引擎在UI设计的巨大变化面前依旧具有良好的表现。
- 作者通过探索不同的线框图方法，构建了下载次数最多的Android应用程序UI设计的大型线框图数据库，并开发了一个基于Web的搜索界面来实现他们的方法。

02 工具相关方法概念

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer dolor quam, pretium eu placerat eu semper et nunc. Nullam ut turpis dictum luctus mi quis luctus lorem.

TWO

方法

01

使用基于GUI的自动探索方法，建立一个包含多种UI设计的大型数据库

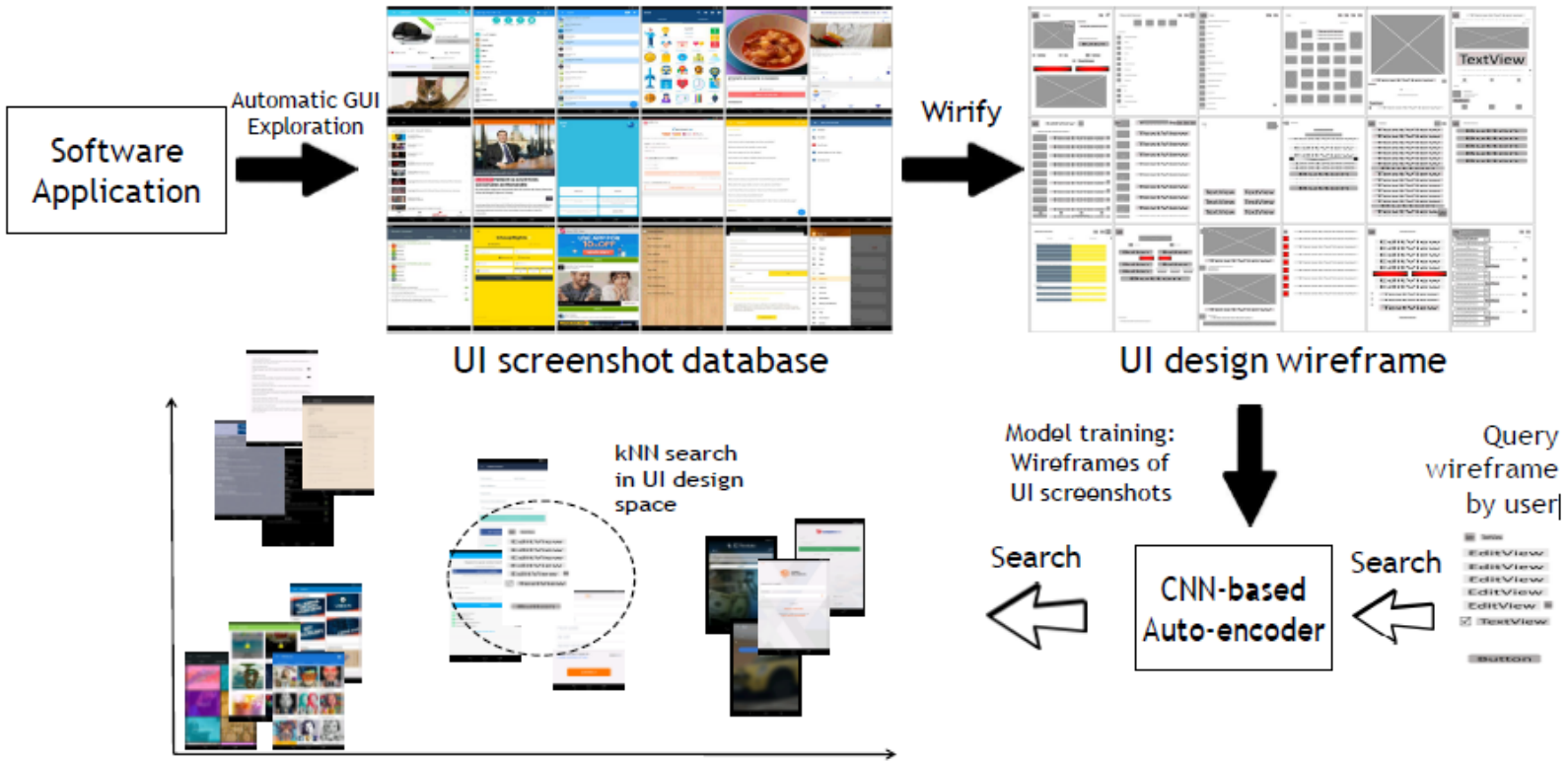
02

训练基于 CNN 的线框自动编码器，使用大型 UI 设计线框数据库对 UI 设计的视觉语义进行编码

03

使用经过训练的线框编码器将 UI 设计嵌入到潜在向量空间中，并支持基于线框的 kNN UI 设计搜索

方法概述


Software Application
Automatic GUI Exploration
Wirify
TextView
UI screenshot database
UI design wireframe
kNN search in UI design space
Model training: Wireframes of UI screenshots
Query wireframe by user
Search
CNN-based Auto-encoder
Search

TWO 01

包含多种UI设计的大型数据库

为了向开发人员展示真实的UI设计空间。作者采用自动数据收集方法:

1. 首先从现有应用程序构建一个大型UI设计数据库
2. 然后使用收集到的数据进一步构建我们的线框数据集。

分为3步:

- 自动GUI探索——Automatic GUI Exploration
- 接线——Wirification
- 线框的最佳表示方式探索

自动GUI探索——Automatic GUI Exploration

通过模拟用户与应用程序的交互来自动探索应用程序的GUI，并输出GUI截屏图像和运行时可视化组件信息，这些信息识别每个组件的类型和在截屏中的坐标。同时为了提高收集到的UI设计的质量，可以进一步启发式**过滤掉无意义的UI**。有如下三个步骤:

1. 模拟器通过模拟用户的操作与应用程序进行交互。
2. 当进入一个新页面时，我们的工具会截取当前UI的屏幕截图并转储XML运行时代码。
3. 计算不同UI动作的权重以过滤无用页面。

自动GUI探索——关键技术（1）

技术目的：确保探索的UI的覆盖范围，作者使用了一定的规则来确定每个潜在可操作组件被按下的权重，规则如下：

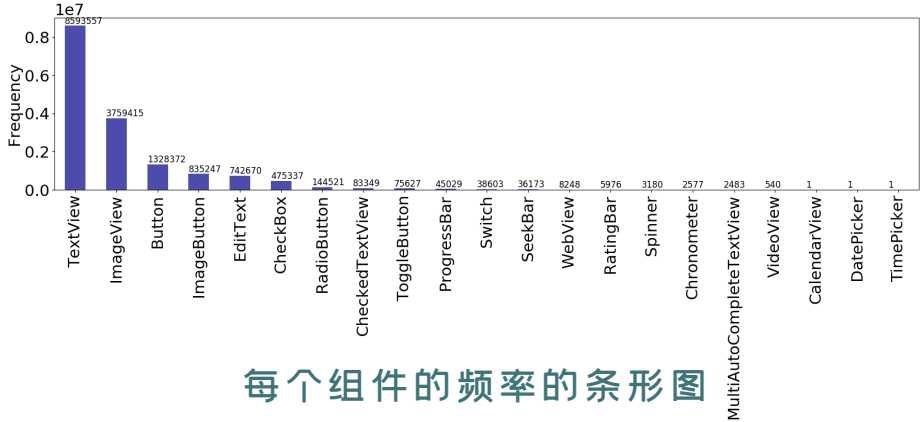
1. 将频率较高的动作赋予较低的权重，这样可以给其他罕见的动作执行机会；
2. 产生更多后续UI的动作具有更高的权重，以便探索各种UI；
3. 控制一些特殊的动作（如硬件后退和滚动），以防它们在错误的时间关闭当前页面或影响其他的动作。

自动GUI探索——关键技术（2）

每个可执行组件的实际权重公式:

$$weights(a) = (\alpha * T_a + \beta * C_a) / \gamma * F_a$$

其中a、T_a、C_a分别为动作、不同类型动作的权重和当前UI中未探索的可执行组件的数量，α、β、γ为超参数。



每个组件的频率的条形图

接线——Wirification

由于作者的方法是基于线框的UI设计搜索。因此，作者需要进一步为数据库中收集的每个UI截图获取一个UI线框图。有如下两个步骤:

1. 通过分析流行的设计师工具和UI的底层实现细节，在设计级别定义了一组对于不同类型的用户交互必不可少的线框组件
2. 通过探索实验找到了每个组件的“正确”表示。

接线——最终选择的16种组件

考虑原则:

1. 设计线框时功能相似且视觉效果相似的组件不会有太大差异;
2. 很少使用的组件对于 UI 设计可能不是很有用。

接线——具体截图与组件配对

使用我们在自动探索应用程序期间转储的 XML 运行时代码文件将 UI 屏幕截图连接到 UI 线框。

TWO 02

基于CNN的线框自编码器

目的：确定 UI 设计的相关性

为此，作者选择使用基于 CNN 的图像自动编码器架构，该架构只需要一组未标记的输入图像进行模型训练。自动编码器将 UI 线框图像作为输入。

1. 编码器通过卷积和下采样层将输入线框压缩成潜在向量表示
2. 解码器通过上采样和转置卷积层从这个潜在向量表示重建输出图像。

重建的输出图像应尽可能与输入图像相似，这表明潜在向量从输入线框设计中捕获信息特征以进行重建。然后可以使用 UI 设计的这种潜在向量表示来衡量 UI 设计的相关性。

卷积——Convolution

卷积运算对图像执行线性变换，使得不同的图像特征变得显着。根据 CNN 可视化的研究，浅卷积层检测简单的特征，例如边缘、颜色和形状，然后在深卷积层中组合以检测特定领域的特征：

- 将图像表示为像素值矩阵：$0 \leqslant p_{hwd} \leqslant 255$
- 在每个位置，卷积操作将内核元素与图像的内核大小子区域相乘，并将这些值相加为输出值
- 一个卷积层可以应用多个内核 (n)。卷积层之后的输出矩阵 $(h \times w \times n)$ 称为特征图，可以将其馈入后续网络层进行进一步处理。

下采样和上采样——Downsampling & Upsampling

- 在编码器中，下采样层将前面卷积层的输出特征图作为输入，并产生空间缩减的特征图。

- 在解码器中，我们使用与下采样相反的上采样层。它们通过用多个值替换输入特征图中的每个值来增加特征图的空间大小。上采样层逐渐增加特征图的空间大小，直到解码器最终重建与输入线框大小相同的输出线框。

模型训练——Model Training

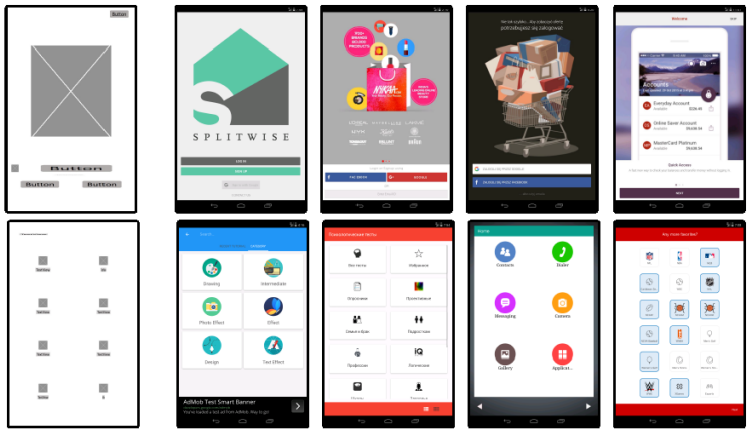
线框自动编码器以最小化均方误差 (MSE) 的重建误差

$$L(X, Y) = ||X - Y||^2$$

在训练时，使用随机梯度下降优化训练数据集上的 MSE 损失。解码器将误差差异反向传递到其输入，即编码器，允许我们使用未标记的输入线框来训练线框编码器。

UI 设计空间中的 kNN 搜索

每个 UI 屏幕截图*uis*都表示为一个潜在的此 UI 设计空间中的向量*V (uis)*， 然后进行搜索



03 方法概念验证

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer dolor quam, pretium eu placerat eu semper et nunc. Nullam ut turpis dictum luctus mi quis luctus lorem.

THREE

01.数据收集

作者使用自动GUI探索方法从这些Android应用程序中构建了一个大型UI屏幕截图数据库。总共从Google Play中抓取了8,000个安装数量最高的Android应用，并成功运行了7,748个Android应用并收集了54,987个UI屏幕截图。

文件放在GitHub‘/results’中

02. Model Hyperparameters

工具的线框自动编码器配置:

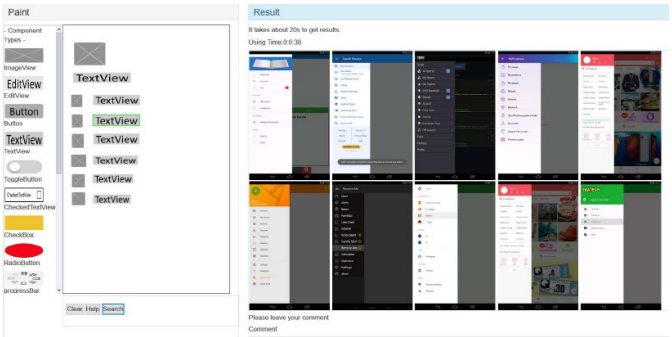
- 输入线框是 RGB 彩色图像并缩放到 180 X 228
- 编码器使用了四个卷积层，分别使用了 16 个 3x3x3 内核、32 个 3x3x16 内核、32 个 3x3x32 内核和 64 个 3x3x32 内核。
- 解码器将输入内核映射中的值上采样为 2 X2 该值的区域。它有四个上采样层。在每个上采样层之后，解码器使用转置卷积层，分别使用32个3x3x64内核、32个3x3x32内核、16个3x3x32内核、3个3x3x16内核。

THREE

03.工具实现

工具使用步骤:

1. 用户在左侧画布上绘制UI线框。目前支持16种最常用的线框组件类型，用户铜鼓拖拽组成自己需要的组件样式。
2. 用户点击搜索按钮，系统就会返回UI设计空间中的前10个（即KNN的k=10）UI设计



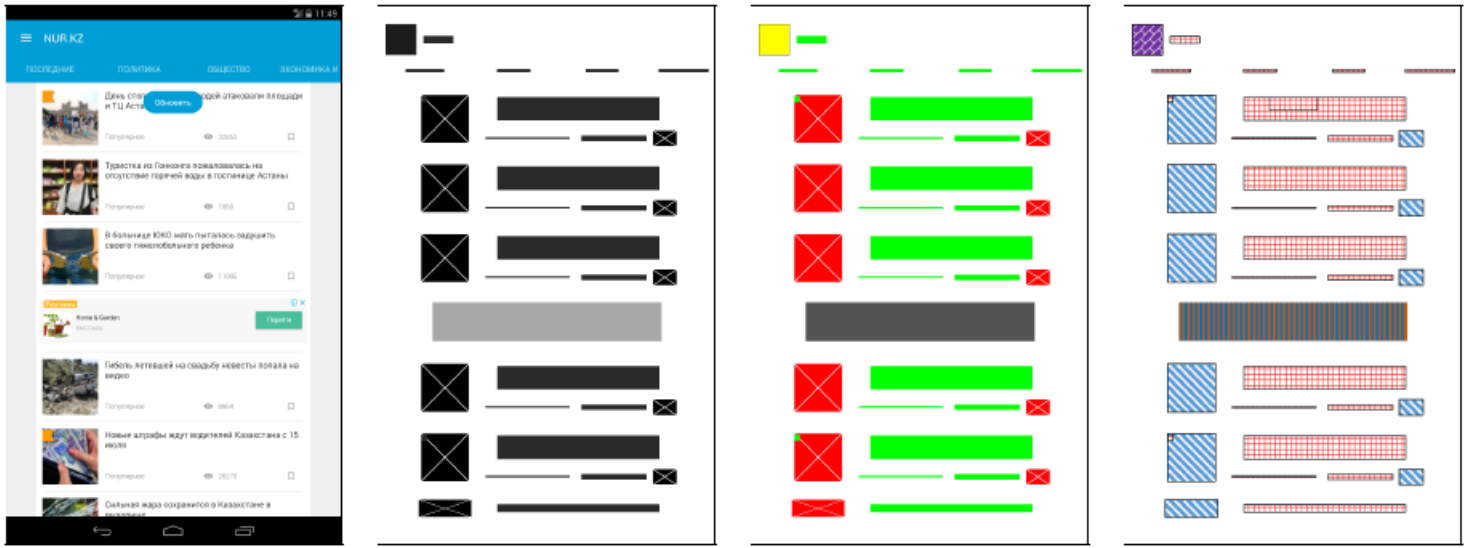
04 实验研究问题与结果

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer dolor quam, pretium eu placerat eu semper et nunc. Nullam ut turpis dictum luctus mi quis luctus lorem.

RQ1: 线框的不同表示的有效性

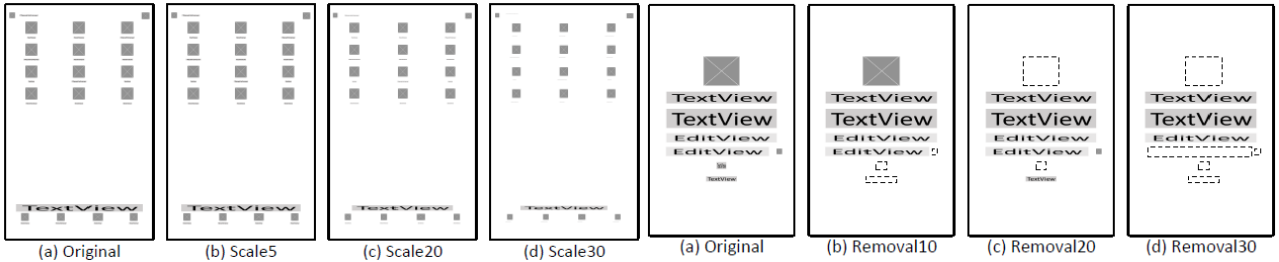
用于表示线框的哪种色觉表现最好？为什么表现不一样？

下图分别为原图、灰度级线框、颜色级线框、纹理级线框



实验过程

1. 作者研究了三种类型的视觉组件表示，包括不同的灰度值、不同的颜色和具有不同纹理的不同颜色
2. 其次，为了根据线框的不同表示来评估 UI 设计搜索方法的性能，作者更改了 Android UI 设计数据库中的 UI 屏幕截图，以人为地创建相关但不同的 UI 设计对。
3. 根据 UI 截图中组件的位置/大小，作者执行两种适用于 UI 设计的更改操作：*组件缩放*和*组件移除*。（下图所示）



(a) Original (b) Scale5 (c) Scale20 (d) Scale30 (a) Original (b) Removal10 (c) Removal20 (d) Removal30

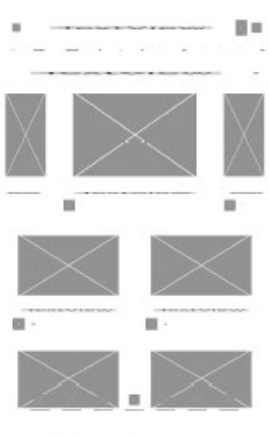
(b) Examples of component-removal treatment

实验结果 (1)

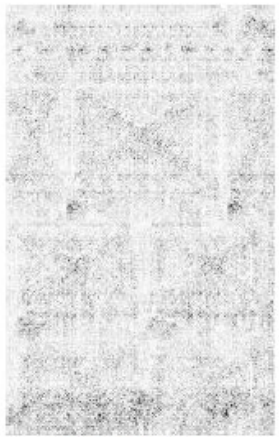
1. 总体而言，颜色级别模型在组件缩放和组件移除处理方面仍然比灰度模型略有优势
2. 作者使用下页图中的反向传播显着性将这些模型可视化。可以发现颜色级别模型的热图最清晰，而纹理级别模型的热图最模糊，噪点较多。灰度热图比颜色一级更模糊，因为灰度一级的组件和背景之间的差异很小。总之，颜色级别模型表现最好，最后作者选择它作为线框数据集的表示

实验结果 (2)

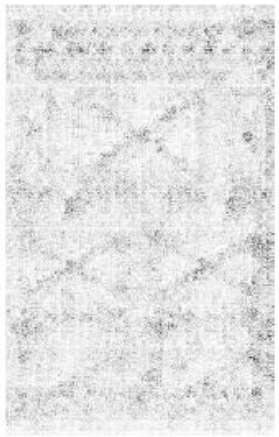
(a) Original screenshot



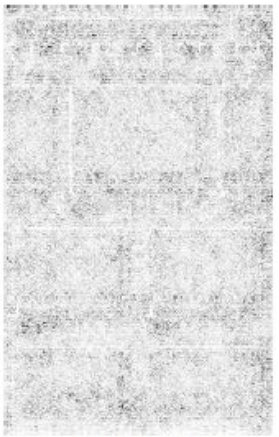
(b) Wireframe



(c) Grey-level



(d) Color-level



(e) Texture-level

RQ2: 我们的UI搜索引擎人工数据集的准确度表现

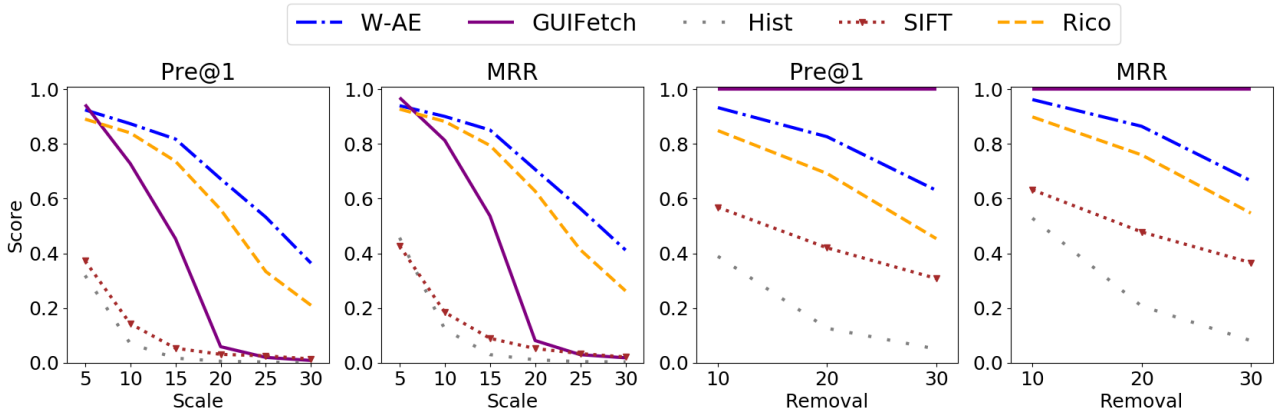
- 面对UI设计的巨大变化，WAE在多大程度上实现了找到相关UI设计的目标？
- 它与基于图像相似性、基于组件匹配或基于朴素神经网络的UI设计搜索相比如何，其原因是什么？

实验过程

- 使用数据集：RQ1中经过原始处理被认为不相关的IU
- 使用方法：
  - 基于图像特征的相似性：定位图像中的关键点，并利用关键点的局部特征来表示图像。两个基线通过图像特征相似度返回前k个最相似的UI设计
  - 基于启发式的组件匹配：通过从两个GUI之间的匹配组件计算出的相似性度量来搜索相似的GUI
  - 基于神经网络的匹配：Rico

实验结果（1）

- 运行性能：一般来说，WAE 比 GUIFetch 和 SIFT 基线快 12 倍和 6 倍，并且颜色直方图和 Rico 基线一样快
- 检索性能：相比之下，的 WAE 在面对较大的组件缩放和组件移除变化时更加稳健



实验结果（2）

检索故障分析

- 还有有一些UI界面的相关性需要人工评判
- 由于模型没有等效地处理来自小或大视觉组件的特征，可能导致大组件的相似性掩盖了小组件的相似性。
- 当查询UI包含一些与大背景组件重叠的前景UI组件时会发生一定的错误，这时候删除背景就可以解决问题，说明还没有消除背景对于组建的影响

RQ3：对现实世界用户界面的泛化能力

- 从开发人员的角度来看，WAE模型表现如何？
- WAE与 RQ2 中的最佳基线相比如何？

实验过程

人工评估UI设计相关性: 招募5名从事相关设计的工作者。两个注释者独立检查UI设计搜索结果。对于每个查询UI线框，他们将20个UI设计中的每一个分类为与查询UI相关或不相关。
评估指标: 第一个指标，计算Cohen的kappa统计量，衡量两个评估者访问两个类别的多个项目之间的一致性。第二个指标，计算Fleiss的kappa统计量，评估多个评估者之间的一致性。
同时通过三种策略将返回的UI设计视为相关:

- 严格（两个注释者都将其标记为相关）
- 中等（大多数注释者将其标记为相关）
- 宽松（至少一个注释器将其标记为相关）

然后计算Precision k（k=1, 5,10）和MRR。.

实验结果

所有参与者都花了大约120分钟来评估1000个UI设计与其相应的查询UI线框的相关性。
基于CNN的方法可以为一组不同的、看不见的实际应用程序查询UI稳健地检索相关的UI设计。相比之下，基于单个组件匹配的启发式方法为这些实际应用程序查询UI找到的相关UI设计要少得多。

LOGO

工具复现

感谢您的观看

汇报人：李若璇

时间：11-30