Software Engineering Group
Department of Computer Science
Nanjing University
http://seg.nju.edu.cn

Technical Report No. NJU-SEG-2019-IW-001

2019-IW-001

Android GUI Search Using Hand-drawn Sketches

Xiaofei Ge

International Conference on Software Engineering: Companion Proceedings 2019

2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)

Android GUI Search Using Hand-drawn Sketches

Xiaofei Ge
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
xfge@smail.nju.edu.cn

***Abstract*—GUI design is crucial to mobile apps. In the early stages of mobile app development, having access to visually similar apps can help designers and programmers gain inspiration for revising their designs or even reuse existing GUI code. We propose an intuitive sketch modelling language to draw GUI sketches, and a deep learning based method to search for visually similar apps according to the sketches. Preliminary results show the potential of our approach.**

***Index Terms*—Graphical User interface, Android app, Code search and recommendation, Reverse engineering.**

I. Background and Motivation

Mobile apps are event-centric programs with rich Graphical User Interfaces (GUIs) [1]. Apps with good GUI designs can be more competitive, driving developers investing considerable effort in GUI developing. Meanwhile, as there are abundant successful apps in the market, the idea of reusing the high-quality GUI code is natural and attractive.

Just like searching for nearest shops in Google, searching for apps with similar GUI designs should be a simple and quick task as well. It is impractical to require users to provide high-fidelity images by a complex modelling tool. Searching tools should support *lightweight* and *low-fidelity* prototyping. However, existing approaches for GUI code search [2, 3] do not provide users with straightforward methods to model GUIs conveniently, which limits their usability.

We propose a novel approach that searches for visually similar apps using sketches. Sketching GUIs on paper is an intuitive method by which users can use a pencil to model app GUIs rapidly. To search for code that realizes the sketch design, We propose to leverage a deep learning (DL) based framework to translate sketches into GUI structures. Then, similarity scores are computed between translated structural GUI data with the ones in the app repositories, based on which search results are ranked before returning to users.

II. Related Work

As early as the era of widespread Java applications, there was much work [2, 4, 5] on Java GUI code search and generation. Reiss's work [3] transfers a Java-based GUI code search method [2] to mobile domain, however, the cumbersome modelling methods limit the feasibility.

Recently, mobile apps are gaining more attention. REMAUI combines OCR and computer vision algorithms to generate GUI code from app screenshots but only supports three widget types. The generated code positions widgets by absolute coordinates, which is inconvenient for programmers to reuse. Chen et al. [1] use deep-learning techniques to generate GUI skeletons from designed UI images. Our tasks are different as we focus on the initial stages of app development when users could not provide high-fidelity designed UI images.

Similarity computation for GUIs is also related to our work. DroidEagle [6] and ViewDroid [7] identify repackaged apps or phishing malware using GUI similarity. However, the relevance and difference of widget types are rarely considered in the existing work.

III. Approach

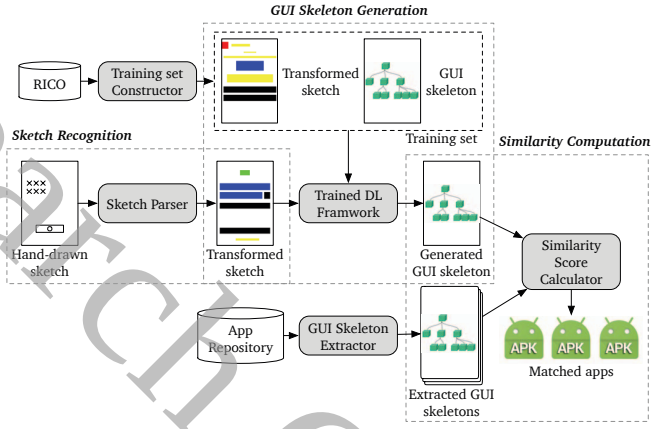

Fig. 1. Overview of our approach

Fig. 1 shows the overview of our approach. It consists of three parts: (I) Input hand-drawn sketches are processed and transformed into intermediate forms by a parser; (II) GUI skeletons of sketches are generated by a DL framework; and (III) Similarity scores are computed between the generated GUI skeletons and the ones in the skeleton dataset. Search results are ranked using the similarity scores.

A. Sketch Modelling Language

We propose a sketch modelling language (SML) to conduct rapid sketching of app GUIs. Striving for the goal of SML being intuitive and easy-to-use, we try not to include all Android widgets but the commonly used ones. We leverage an Android app UI dataset to decide which widgets to select. The dataset named *Rico* contains 72k UIs mined from 9.7k Android apps using a combination of human and automated exploration. GUI hierarchies are analyzed and converted into the form of `JSON`, along with the widget details used in each UI, e.g., widgets' implementing class name, superclass names, coordinates of the bounding box.

2574-1934/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSE-Companion.2019.00060

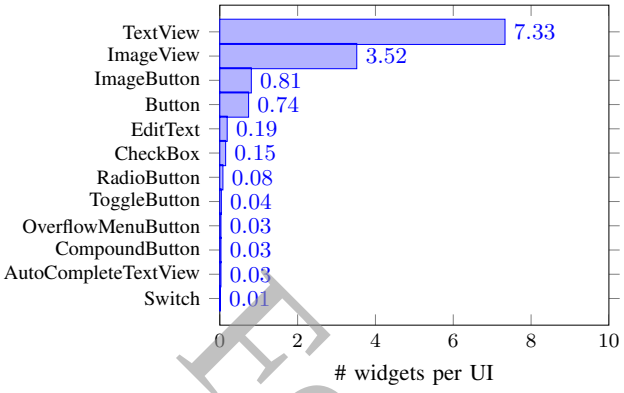

Fig. 2. Average number of the most frequent widgets per UI

We counted the occurrence times of official Android widgets (with class names prefixed with widgets). For a customized widget, we considered its most direct superclass prefixed with android.widget as its type. Fig. 2 depicts the widgets that have more than 0.01 usage times per page, on average. The 12 most frequently used widgets listed in the figure account for 99.8% of the total widget usages. For these widgets, we further merge the functionally and visually identical ones into a single category. For example, AutoCompleteTextView and EditText are both widgets used to get user inputs and the latter is a superclass of the former, and are therefore, merged as EditText.

Finally, seven kinds of widgets—TextView, EditText, ImageView, Button, RadioButton, Switch and CheckBox are selected. We designed sketch representations for each widget with different geometric shapes. We also developed a recognition tool (Sketch Parser in Fig. 1) to identify types and bounding boxes of widgets from sketches.

B. Deep-learning based GUI Skeleton Generation

Android GUIs, implemented in the form of tree-like structures, are called *GUI skeletons*. With each node representing a widget or a layout, a GUI skeleton defines what and how the components of layouts and widgets are composed for reproducing the UI elements and their spatial layout.

We employ a deep-learning method to generate GUI skeletons from sketches, following the idea of [1]. In [1], GUI skeletons are generated from UI images. Large quantities of pairs of UI images and GUI skeletons were collected for model training. Our task is different from theirs in that we need the skeletons and their corresponding sketches to train the DL model. Therefore we implemented a sketch generation method to construct the training set based on GUI details contained in *Rico*. To reduce training complexity and improve learning accuracy, we transformed hand-drawn sketches into colored images in which widgets are represented by rectangles with different colors (as shown in Fig. 1). Experiments show that our DL framework is able to generate GUI skeletons for sketches not included in the dataset.

C. GUI Skeleton based Similarity Computation

The similarity between a pair of GUIs are represented by the similarity between their corresponding GUI skeletons. As GUI skeletons are tree-like structures, we proposed a similarity computation method based on the algorithm of the largest weighted common subtree embeddings (LWCSE) with penalties. Specifically, we employ labeled trees to depict GUI skeletons where tree nodes can distinguish different widget types. The similarity score of two GUI skeletons, considering both the structural information of the tree and the types of widget nodes, is computed by applying LWCSE on a weight function on pairs of nodes.

The search is conducted by comparing the GUI skeleton of the sketch against all skeletons in the data set. Search results are ranked before returning to users. The largest common weighted GUI substructure embeddings are provided for reference as well. As the experiment shows, our approach can effectively evaluate the similarity between two GUI Skeletons.

IV. Preliminary Results

Our goal is to find apps that contain UIs similar to input hand-drawn sketches. We performed a preliminary experiment on 60 Android apps from different categories. These apps cover at least one or more of the six common UI scenarios: login & register, search, setting, list, picture, and detail. For each UI scenario, we selected an app and drew sketches for the concerned GUIs. We refer to the GUIs from which we drew sketches as *target GUIs*. Six hand-drawn sketches were processed by our tool, and the results are shown in Table I.

TABLE I
Target GUIs in Search Results

UI scenario	Target App package name	Similarity score*	Rank*
login & register	com.choiceoflove.dating	81.08%	1
search	com.parfield.prayers.lite	64.86%	5
setting	com.google.android.apps.messaging	72.97%	5
list	com.baviux.pillreminder	67.57%	6
picture	com.kudago.android	78.42%	1
detail	com.google.android.play.games	70.27%	1

* Scores and ranks of the six target GUIs in the final search result list, respectively.

As the result shows, our approach can find the target GUIs in the top six candidates, with three ranked top one. We also ran the returned apps and found that the other top-ranked apps in the search result were also visually similar to the sketches. We are currently working on collecting more Android apps, performing experiments on a larger-scale and conducting systematic evaluations.

V. Conclusion

The preliminary results show that given sketches, the proposed method can find visually similar apps. The search strategy is under improvement based on further experiments, and we are conducting empirical studies to improve the effectiveness and efficiency of the approach.

REFERENCES

[1] C. Chen, T. Su, G. Meng, Z. Xing, and Y. Liu, "From UI design image to GUI skeleton: a neural machine translator to bootstrap mobile GUI implementation," in *Proceedings of the 40th International Conference on Software Engineering*. ACM, 2018, pp. 665–676.

[2] S. P. Reiss, "Seeking the user interface," in *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*. ACM, 2014, pp. 103–114.

[3] F. Behrang, S. P. Reiss, and A. Orso, "GUIfetch: supporting app design and development through GUI search," in *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*. ACM, 2018, pp. 236–246.

[4] A. Caetano, N. Goulart, M. Fonseca, and J. Jorge, "JavaSketchIt: Issues in sketching the look of user interfaces," in *AAAI Spring Symposium on Sketch Understanding*, 2002, pp. 9–14.

[5] J. Seifert, B. Pfleging, E. del Carmen Valderrama Bahamóndez, M. Hermes, E. Rukzio, and A. Schmidt, "Mobidev: a tool for creating apps on mobile phones," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, 2011, pp. 109–112.

[6] M. Sun, M. Li, and J. Lui, "DroidEagle: seamless detection of visually similar android apps," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 9.

[7] F. Zhang, H. Huang, S. Zhu, D. Wu, and P. Liu, "ViewDroid: Towards obfuscation-resilient mobile application repackaging detection," in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*. ACM, 2014, pp. 25–36.