Learning Structural Similarity of User Interface Layouts using Graph Networks

Supplementary Materials

Dipu Manandhar[1], Dan Ruta[1], and John Collomosse[1,2]

[1] CVSSP, University of Surrey — Guildford, UK.
[2] Adobe Research, Creative Intelligence Lab — San Jose, CA.
{d.manandhar, d.ruta}@surrey.ac.uk, collomos@adobe.com

This supplementary document provides:

– A detailed description of the new GoogleUI dataset, which is included in the supplementary material as a compressed archive and will be publicly released on acceptance.
– Comprehensive results presenting performance of the proposed method under various configurations and settings, to extend the configurations reported in Table 2 of the main paper.
– Further qualitative retrieval results for user interface (UI) layouts, to complement the quantitative and qualitative results included in the main paper.

Sec. 1 discusses data collection, cleaning, and annotation procedure including the format of the available UIs and annotations in GoogleUI dataset.

Sec. 3 provides qualitative comparisons between the proposed method and baselines, and then provides additional retrieval results visualizing both screenshot images and their semantic UIs.

1 GoogleUI Dataset

We collected the GoogleUI Dataset which contains diverse data in the wild. Different from RICO which has clean screenshots captured by crowd-sourcing Android apps downloaded from PlayStore, GoogleUI contains noisy images collected from Google Images which often includes captured photos of mobile phones with diverse views and various backgrounds. GoogleUI dataset serves as a benchmark to study the transferability of UI desgin search models as discussed in Section 4.6 of the main paper. In the following, we discuss its collection, cleaning, annotation, and format.

1.1 Collection

The dataset is collected from the public internet via the Google Image search engine, using textual queries. A pool of search keywords was first created such as “mobile ui design”, “mobile ux android”, “smartphone ui iphone”, “phone ux template", etc. These keywords were further combined with words such as "shopping", "social apps", "travel" etc. to retrieve screenshots from various app categories.

We developed a web image crawler to scrape images. Specifically, our web crawler makes use of *Selenium*, an automated browser testing tool with python bindings. This process yielded approximately 21K images.

1.2 Data Cleaning and Annotation

The initial set of data often contained unwanted backgrounds such as phone frames, human hands holding the device, digital graphics around the interface, and multiple UIs per image. Moreover, some images contained arbitrarily oriented UIs. We used Amazon Mechanical Turk in order to obtain bounding boxes that are subsequently used to crop out UIs as screenshots. We excluded the phone frames and ignored the images that do not contain portrait UIs. Finally, we obtained 18.5K samples of cleaned images ready for further annotation.

Next, we automatically parsed the clean images into UI components using a UI component detector. In particular, we trained a VGG-16 based Faster-RCNN [3] trained using the annotated UI components in the RICO dataset; we use the training partition of RICO to detect 25 classes of UI components. The confidence threshold for detection was empirically set to 0.5 upon qualitative inspections, and class labels and corresponding bounding boxes for all the detected components were obtained. Fig. 1 shows sample images from GoogleUI dataset together with component detection results. We can observe that most the components are correctly detected. Note that our aim is to obtain estimates of layouts to evaluate the transferability of layout search models. Full detection results are included in the compressed archive accompanying this supplementary document.

1.3 Annotation Format

Annotations for GoogleUI dataset are provided in two files:
(a) dataset.csv
This csv file containing image names, their URLs, and the bounding box parameters for cropping out UI viewport from the original images. Each row of the file contains annotation in the following format.
`<img_name> <height> <left> <top> <width> <url>`

(b) gogui_boxinfo.pkl
A Python pickle file that contains UI component label annotations. We use the 25 classes of UI components from RICO to annotate GoogleUI images. The list of the component names is shown in Table 1. The pickle file consists of a dictionary with `img_name` as the dictionary keys and the values have corresponding UI component annotations. In each key-value pair, the value is a dictionary with following attributes:

– `nComponent`: number of UI components in the image

- `class_id`: a list of class ids of UI components
- `bbox`: an array of shape `nComponent`×4; each row is bounding box in format <`left`> <`top`> <`width`> <`height`>
- `imgW`: width of the image
- `imgH`: height of the image

Table 1. List of 25 classes of UI components in RICO dataset. The same labels are used for GoogleUI annotations. Please refer to [2] for more details on UI categories.

ID	**Name**	**ID**	**Name**	**ID**	**Name**	**ID**	**Name**
1	Toolbar	8	Card	15	Modal	22	Slider
2	Image	9	List Item	16	Button Bar	23	Number Stepper
3	Icon	10	Advertisement	17	Pager Indicator	24	Video
4	Web View	11	Background Image	18	On/Off Switch	25	Date Picker
5	Text Button	12	Drawer	19	Checkbox		
6	Text	13	Input	20	Map View		
7	Multi-Tab	14	Bottom Navigation	21	Radio Button		

2 Performance of variants of the proposed method

Table 2 shows detailed ablation studies for the proposed method using different graph representations, decoder architectures and various embedding dimensionalities with/without triplet supervision.

3 Retrieval Results

3.1 Qualitative comparison between baselines

Fig. 2, 3 and 4 show retrieval comparisons between AE method [1], CAE [2] and the proposed GCN-CNN-TRI. The results clearly show the performance gain of the proposed framework for UI layout search.

3.2 Additional retrieval results with semantic UIs

We provide additional retrieval results for various layouts using the proposed method in Fig. 5, 6, 7 and 8. For each screenshot image, we also show the semantic UIs where different color represent various UI component classes. These results further illustrate the effectiveness of the proposed method for layout search.

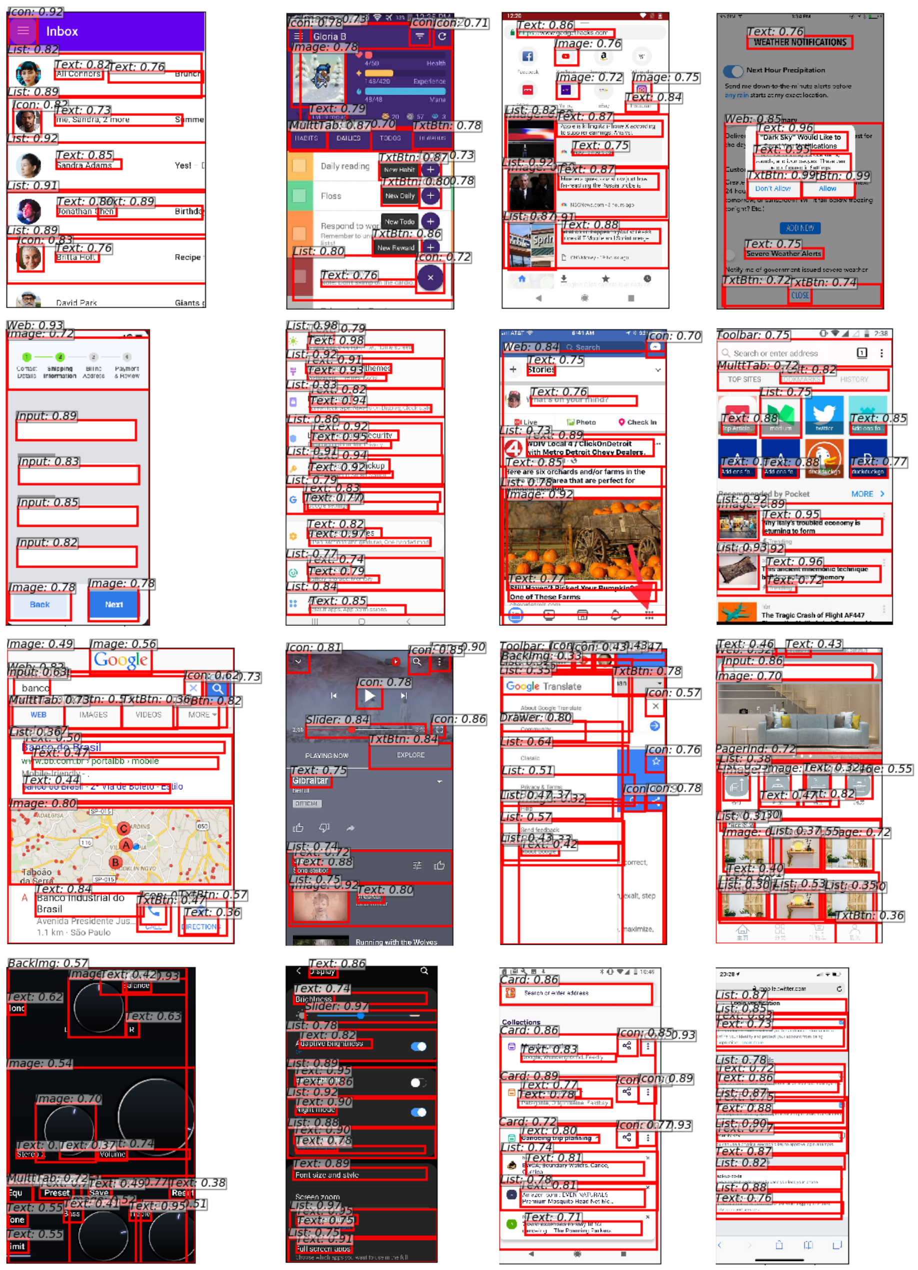

Fig. 1. Sample images from the newly constructed GoogleUI dataset. The UI components detected by Faster-RCNN are overlaid on the images.

Table 2. Performance of variants of the proposed method GCN-CNN for (D)irected vs. (U)ndirected graph representation, and (Str)ided vs. (Ups)ampling (dec)oder stage for several (dim)ensionalities of embedding. Unsupervised and (tri)plet supervision are evaluated at $k = [1, 5, 10]$ over RICO. Numbers in parentheses indicate triplet supervision.

Method	Dec.	Dim	MIoU (%)			MPixAcc (%)		
top-k			1	5	10	1	5	10
U(+tri)	Ups	128	59.5(59.6)	49.2(50.7)	45.9(48.8)	66.7(67.4)	58.1(59.6)	55.2(58.4)
U(+tri)	Ups	256	56.6(59.7)	49.2(51.5)	46.7(49.1)	64.1(67.6)	57.4(60.0)	55.7(57.9)
U(+tri)	Ups	512	57.7(58.4)	50.1(52.2)	47.4(49.8)	65.4(66.0)	59.0(61.7)	56.8(59.3)
U(+tri)	Ups	1024	58.3(60.2)	50.4(52.8)	47.5(50.8)	64.7(68.2)	59.3(61.4)	56.8(60.1)
U(+tri)	Ups	2048	58.0(59.0)	50.4(51.6)	48.0(49.4)	65.5(66.6)	59.4(60.7)	57.7(59.3)
U(+tri)	Str	128	59.2(58.1)	50.0(51.2)	47.4(49.4)	66.1(65.9)	58.7(60.2)	56.6(58.7)
U(+tri)	Str	256	57.5(59.3)	50.7(51.8)	47.7(49.6)	65.1(65.9)	59.9(61.1)	56.9(59.5)
U(+tri)	Str	512	59.3(61.0)	50.8(52.9)	48.2(50.5)	66.2(69.8)	59.9(62.0)	57.3(60.0)
U(+tri)	Str	1024	58.7(59.3)	50.5(52.0)	47.6(50.1)	66.7(67.8)	59.4(61.0)	57.1(59.5)
U(+tri)	Str	2048	58.9(61.6)	50.9(53.4)	48.1(51.0)	66.3(70.2)	59.7(62.5)	57.6(60.6)
D(+tri)	Ups	128	55.5(58.6)	46.5(48.7)	43.7(46.0)	63.5(65.8)	55.4(57.6)	52.6(54.8)
D(+tri)	Ups	256	57.4(58.4)	49.1(50.1)	46.9(47.8)	66.1(66.7)	58.5(59.1)	56.5(57.2)
D(+tri)	Ups	512	57.3(60.4)	51.0(52.2)	48.1(49.9)	65.9(68.5)	60.6(61.3)	57.5(59.5)
D(+tri)	Ups	1024	58.2(60.1)	50.3(53.1)	47.3(51.0)	65.1(67.4)	57.7(61.9)	55.5(60.1)
D(+tri)	Ups	2048	59.0(60.4)	50.2(52.9)	47.1(50.3)	66.4(69.3)	59.2(62.4)	56.3(59.7)
D(+tri)	Str	128	58.8(59.8)	50.1(51.6)	46.6(49.7)	67.5(68.4)	59.7(61.5)	56.2(60.0)
D(+tri)	Str	256	58.1(60.3)	50.0(52.0)	47.2(50.2)	66.3(61.0)	59.1(61.9)	56.2(59.8)
D(+tri)	Str	512	59.4(62.8)	49.3(53.3)	47.1(50.5)	68.3(71.0)	58.9(62.7)	56.5(60.0)
D(+tri)	Str	1024	59.5(**62.8**)	51.3(54.1)	**48.6**(**51.8**)	68.0(**71.9**)	61.1(63.5)	**58.2**(**61.9**)
D(+tri)	Str	2048	**60.0**(61.7)	**51.6**(**54.1**)	48.3(51.3)	**68.1**(70.1)	**61.4**(**64.0**)	58.0(61.0)

References

1. Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., Kumar, R.: Rico: A mobile app dataset for building data-driven design applications. In: Proceedings of the 30th Annual Symposium on User Interface Software and Technology. UIST '17 (2017)
2. Liu, T.F., Craft, M., Situ, J., Yumer, E., Mech, R., Kumar, R.: Learning design semantics for mobile apps. In: The 31st Annual ACM Symposium on User Interface Software and Technology. pp. 569–579. UIST '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3242587.3242650, http://doi.acm.org/10.1145/3242587.3242650
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proc. NIPS (2015)

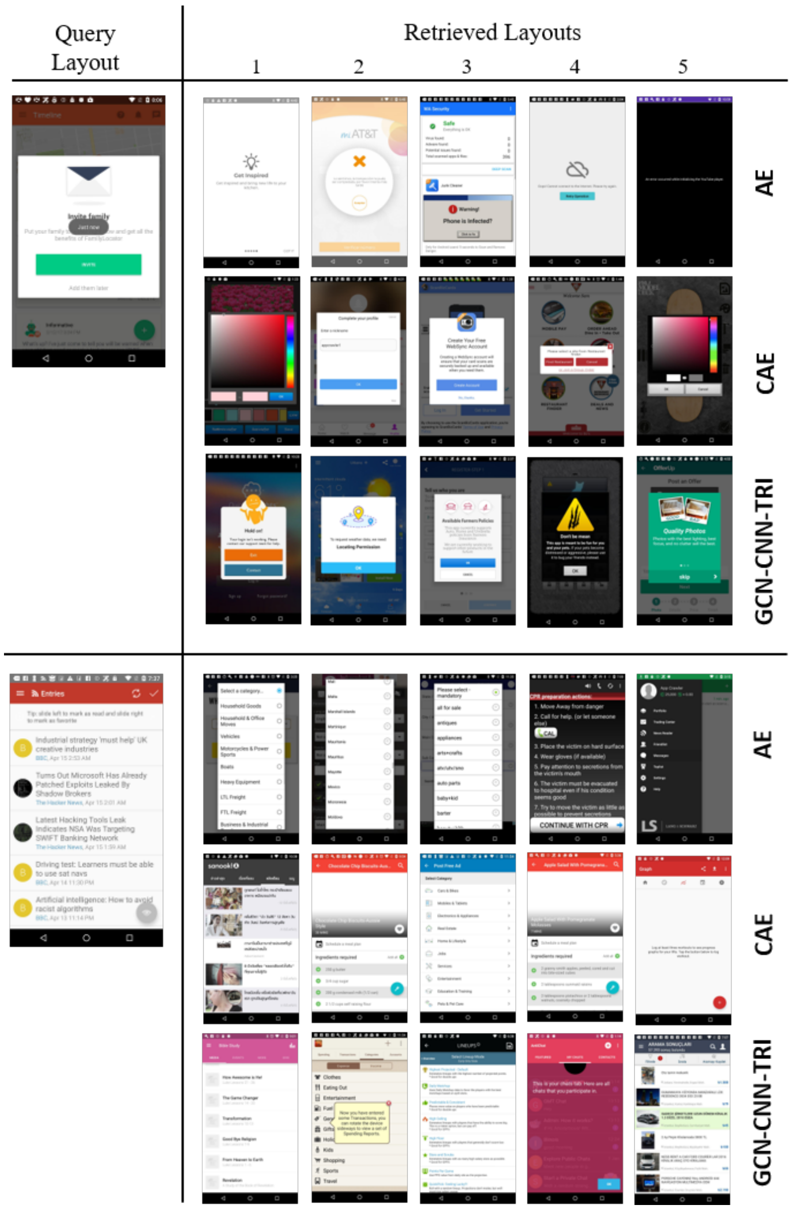

Fig. 2. Retrieval comparison-1: Proposed GCN-CNN-TRI with AE [1] and CAE [2] on RICO

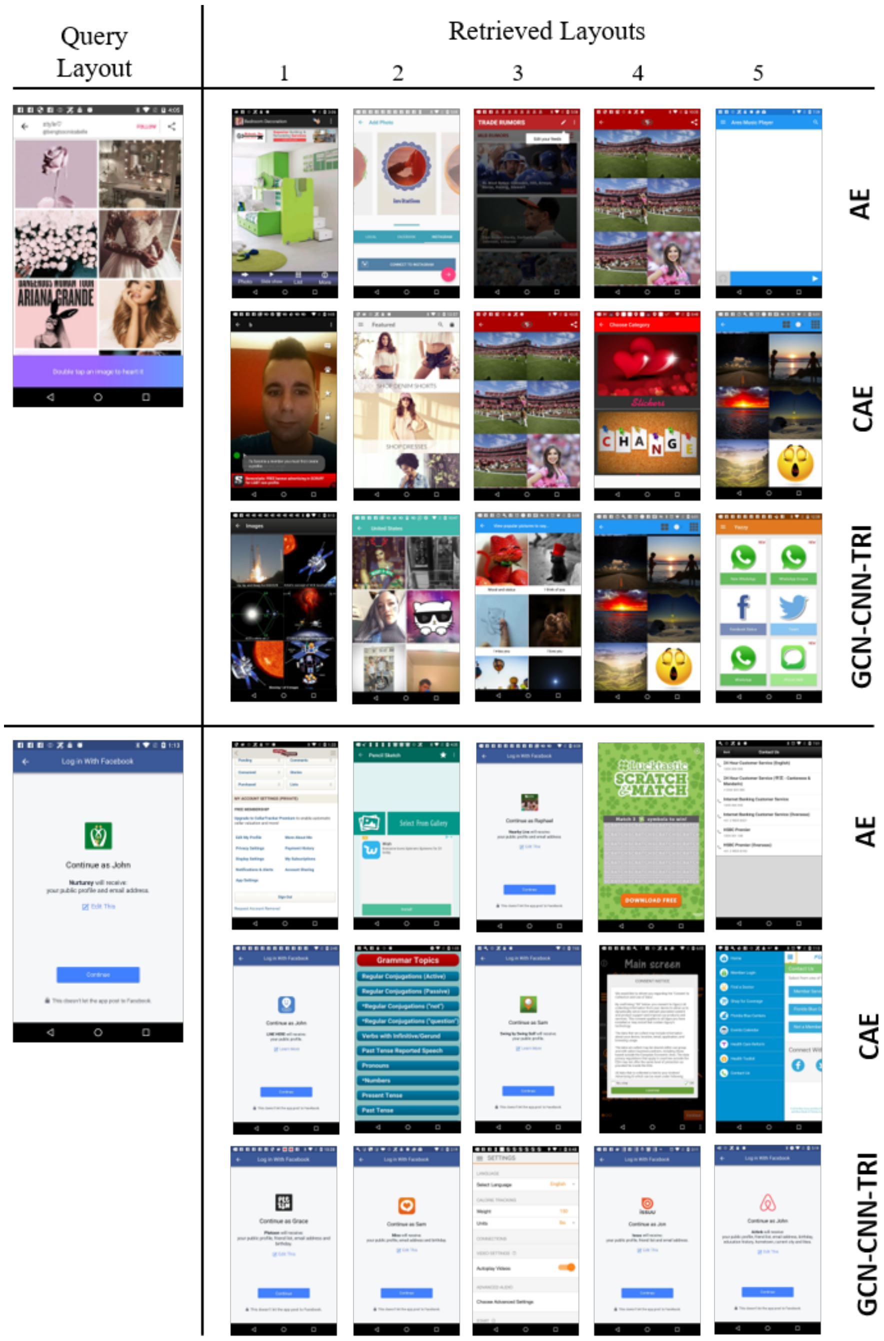

Fig. 3. Retrieval comparison-2: Proposed GCN-CNN-TRI with AE [1] and CAE [2] on RICO

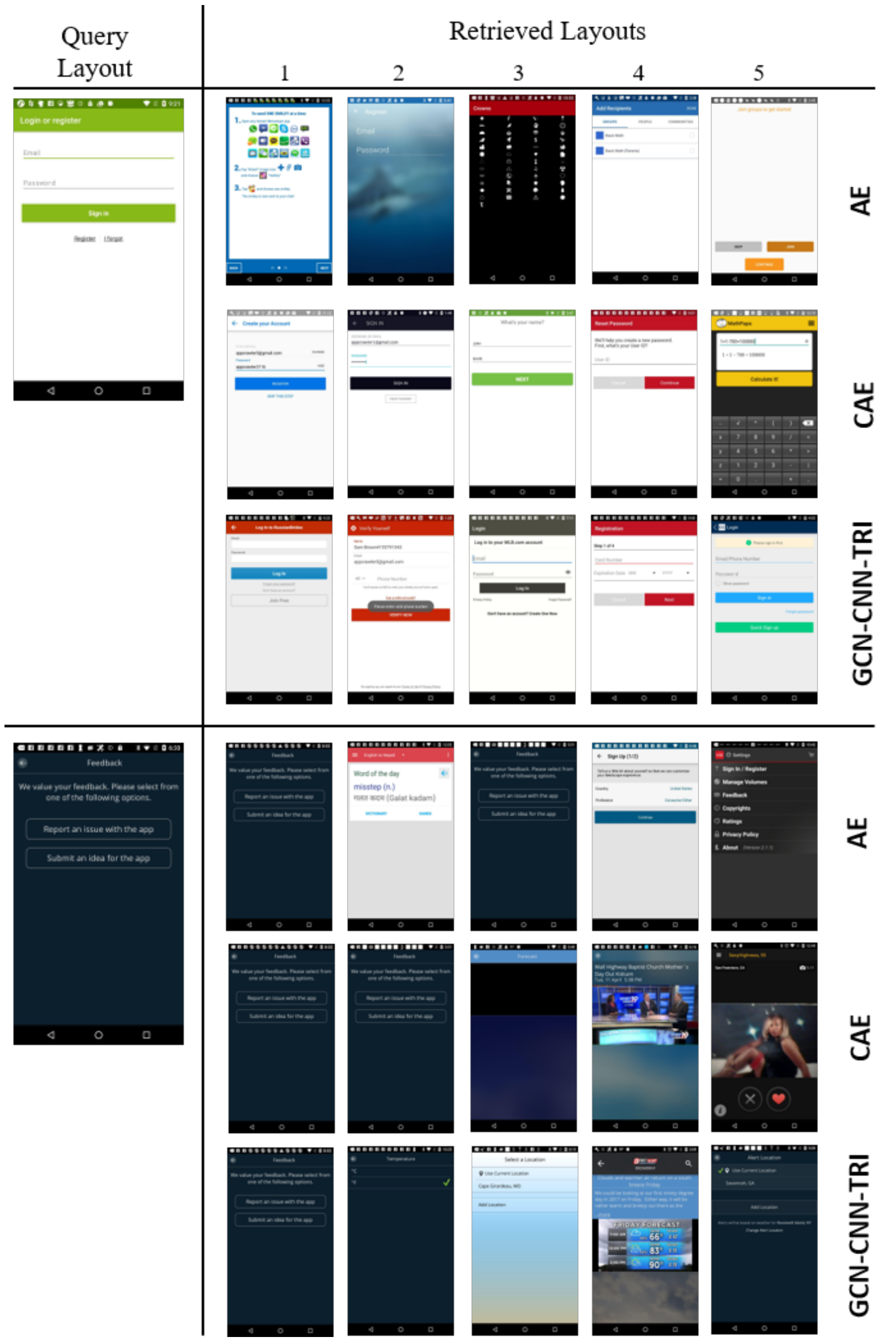

Fig. 4. Retrieval comparison-3: Proposed GCN-CNN-TRI with AE [1] and CAE [2] on RICO

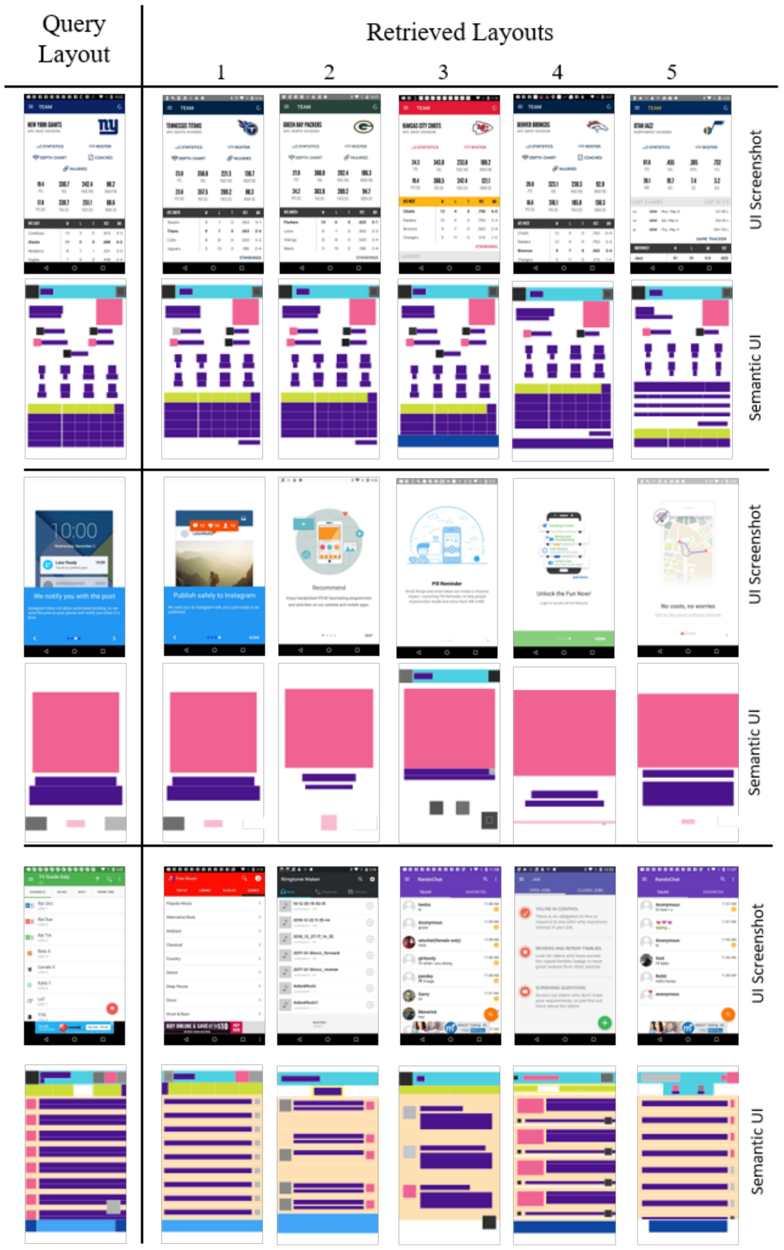

Fig. 5. Retrieval results-1: Additional retrieval examples from RICO dataset with both screenshots and corresponding semantic UIs showing the layout.

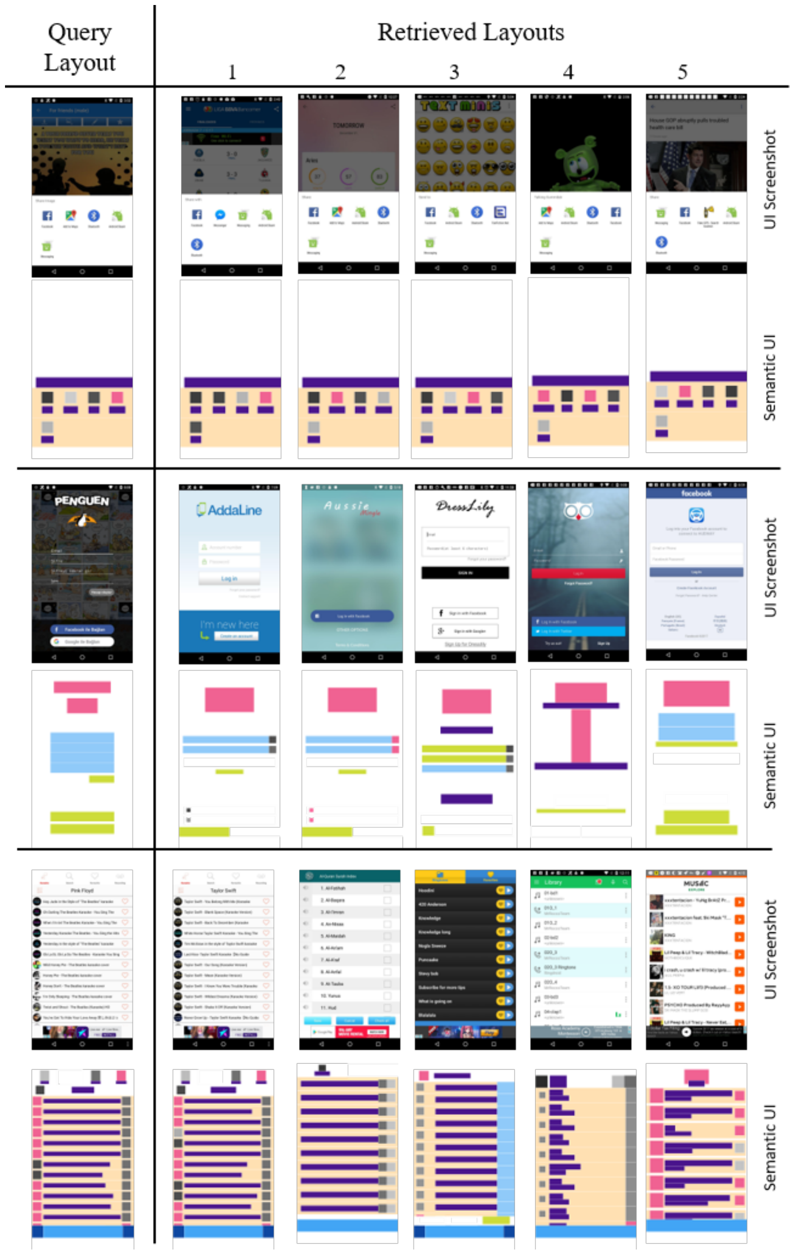

Fig. 6. Retrieval results-2: Additional retrieval examples from RICO dataset with both screenshots and corresponding semantic UIs showing the layout.

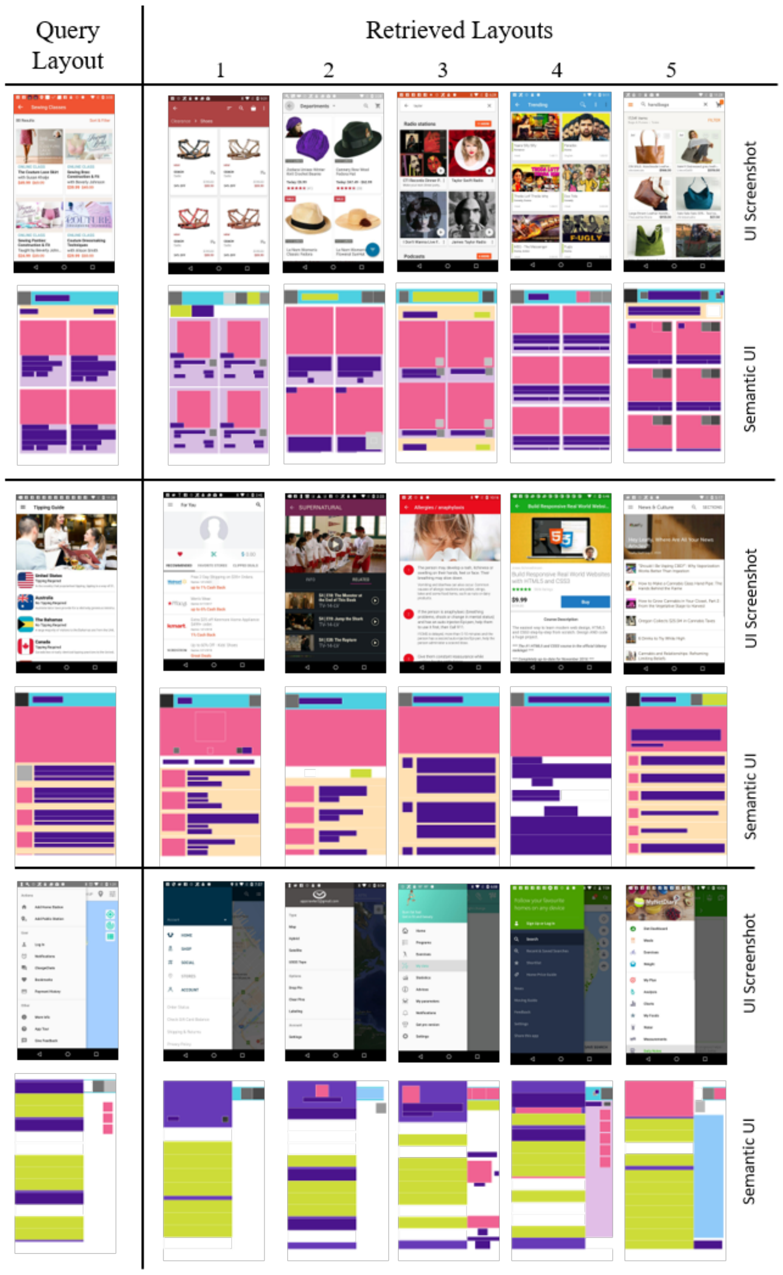

Fig. 7. Retrieval results-3: Additional retrieval examples from RICO dataset with both screenshots and corresponding semantic UIs showing the layout.

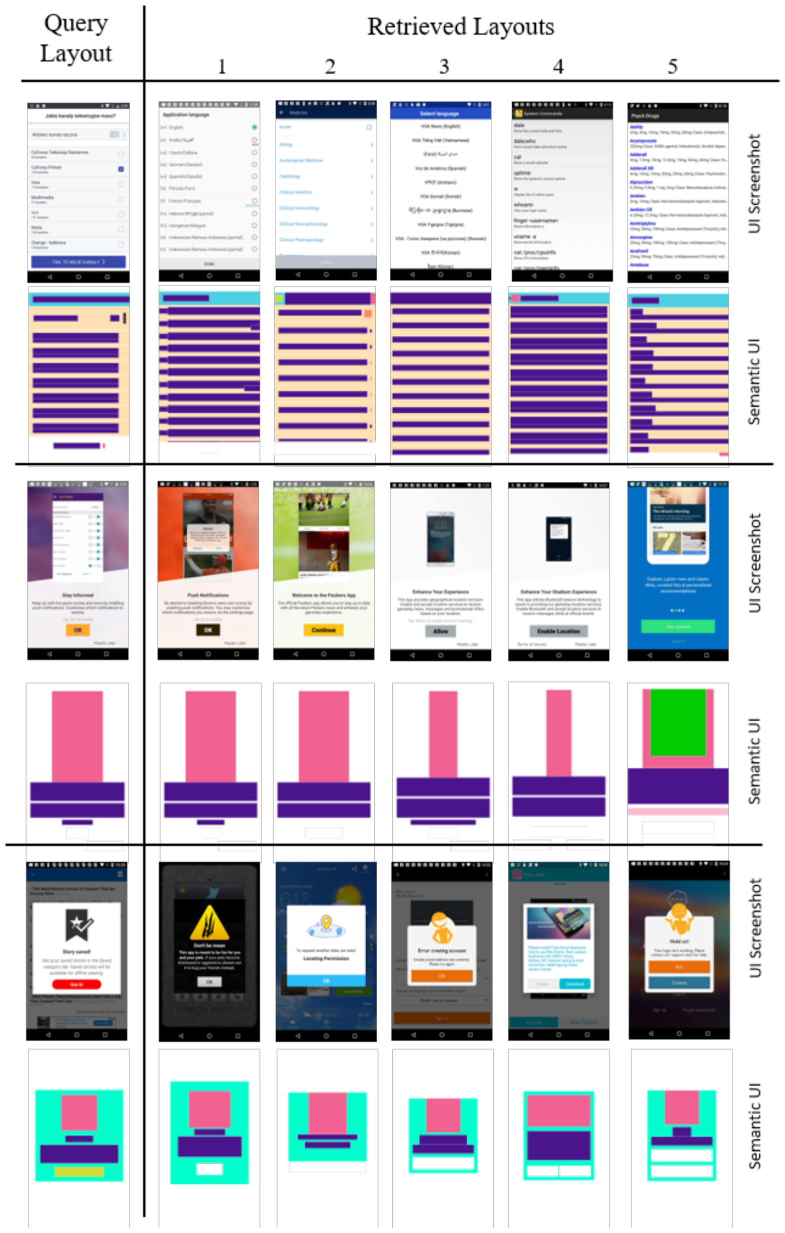

Fig. 8. Retrieval results-4: Additional retrieval examples from RICO dataset with both screenshots and corresponding semantic UIs showing the layout.