Yunnong CHEN, Yankun ZHEN, Chuning SHI, Jiazhi LI, Liuqing CHEN, Zejian LI, Lingyun SUN, Tingting ZHOU, Yanfang CHANG, 2023. UI layers merger: merging UI layers via visual learning and boundary prior. *Frontiers of Information Technology & Electronic Engineering,* 24(3):373-387.
https://doi.org/10.1631/FITEE.2200099

UI layers merger: merging UI layers via visual learning and boundary prior

Key words: User interface (UI) to code; UI design lint; UI layer merging; Object detection

Corresponding author: Liuqing CHEN
E-mail: chenlq@zju.edu.cn
ORCID: https://orcid.org/0000-0002-9049-0394

Motivation

- With the fast-growing graphical user interface (GUI) development workload, some research attempted to generate front-end code from GUI screenshots.
- It can be more suitable for using user interface (UI) design drafts that contain UI metadata.
- However, fragmented layers inevitably appear in the UI design drafts, which greatly reduces the quality of the generated code.
- None of the existing automated GUI techniques detects and merges the fragmented layers for improving the quality of the generated code.

Main idea

- The proposed UI layers merger (UILM) was divided into two steps: **merging area detection** and **layer merging**.
- A data preprocessing pipeline was constructed.
- The layer boundary information was incorporated into merging area detection as prior knowledge to improve the visual understanding of UI.
- A novel data augmentation and spatial feature fusion strategy is also introduced to boost the performance of our merging area detection.
- To retrieve the fragmented layers and merge them into UI components, we proposed a rule-based merging algorithm.

Method

The proposed method consists of three main steps:

- **Dataset construction,** for generating UI screenshots and implementing a data augmentation method
- **Merging area detection,** for locating the merging area of the UI components and determining the reliable boundary for the area
- **Layer mering,** for retrieving fragmented layers within the bounding boxes and merging them into UI components

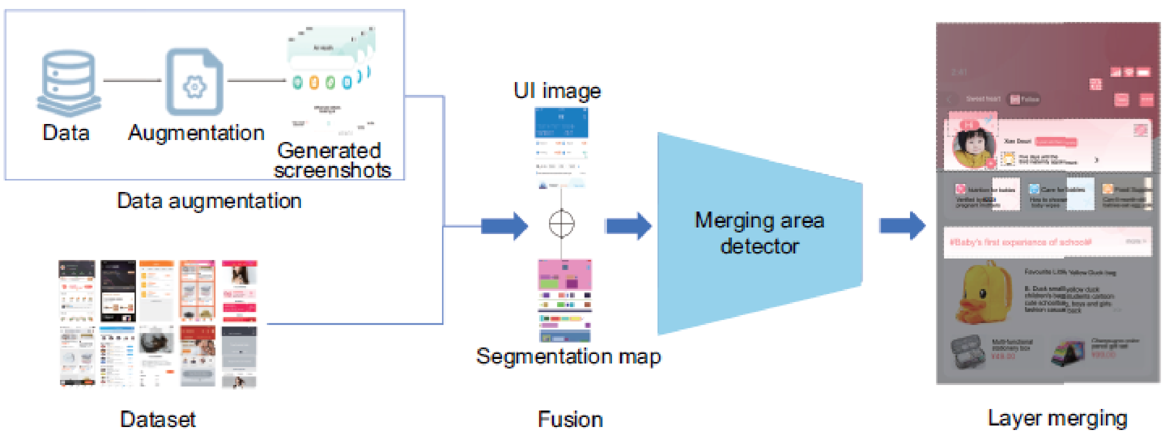


Fig. 3 Overview of the proposed method

Method (Cont'd)

Dataset construction: given an UI design draft (also called artboard), we parse it to a JSON file and generate the train labels using the COCO format. To incorporate the boundary information, the segmentation maps are generated by filling the layers with colors in traversing order.

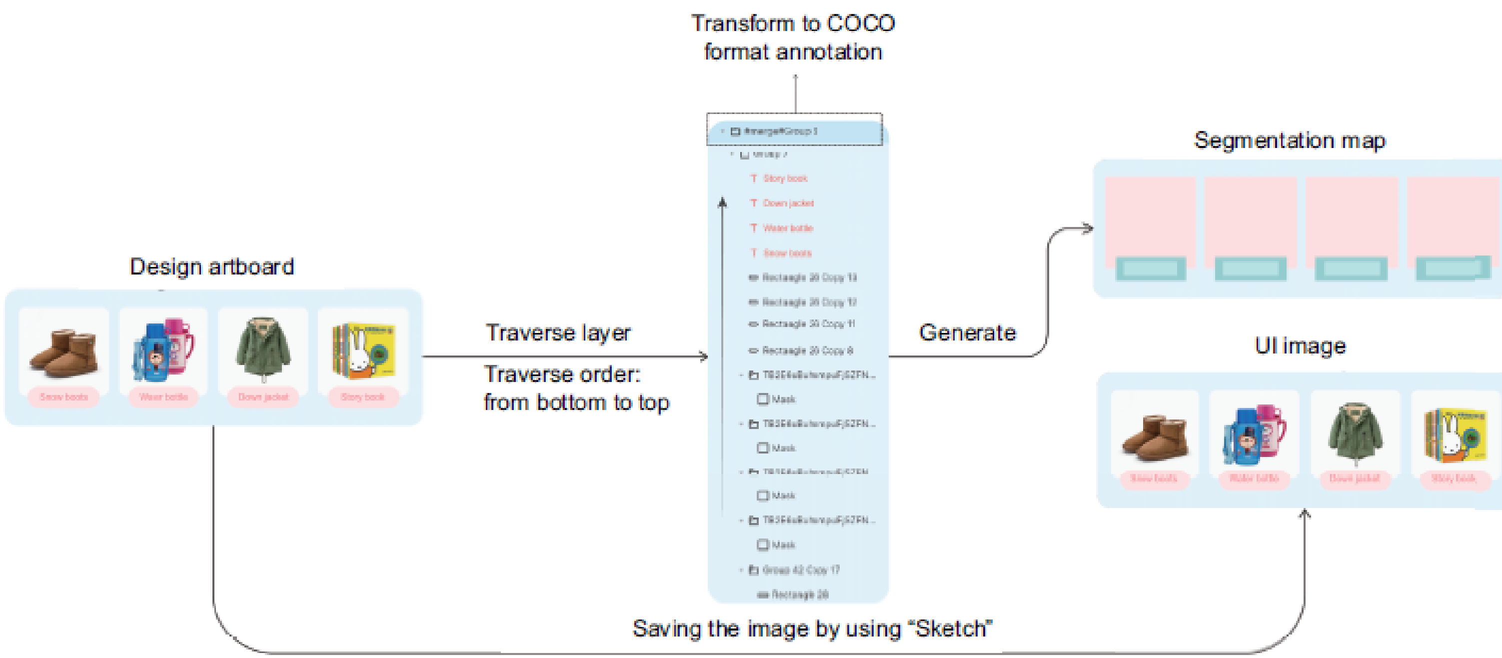


Fig. 4 Parsing pipeline of user interface (UI) draft preprocessing

Method (Cont’d)

Merging area detection: to determine reliable boundary for the merging area, we adopt adaptive convolution to learn an anchor with an adaptive shape and multi-stage box regression to improve the boundaries’ alignment accuracy in the context of complex UI layouts.

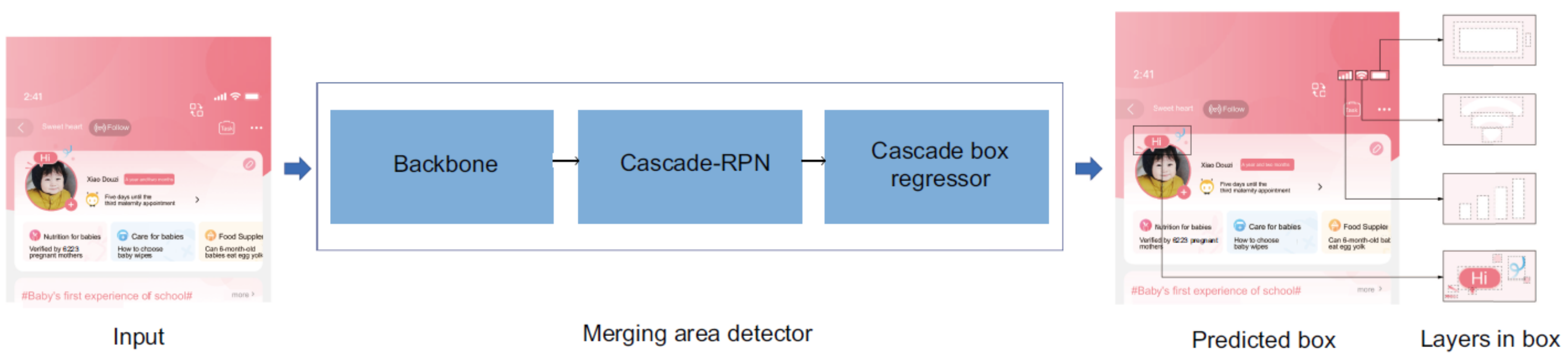


Fig. 5 Architecture of the merging area detector (RPN: region proposal network)

Method (Cont'd)

Feature fusion: we propose two fusion strategies as prior knowledge to improve the spatial understanding of the UI image.

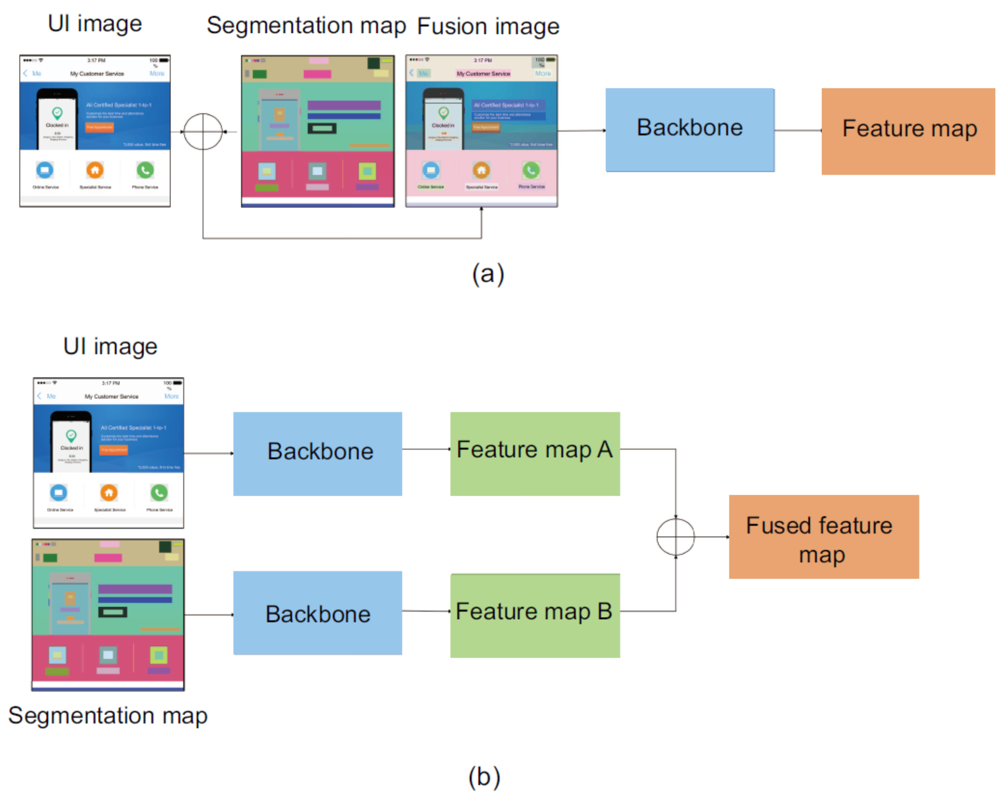


Fig. 6 Two strategies for feature fusion: (a) spatial fusion strategy; (b) feature fusion strategy

Major results

The performance of our MAD is improved by 26.61% and 18.15% in terms of mAP compared to RetinaNet and Faster-RCNN, respectively. Our approach also outperforms competing methods in terms of mAP compared to Cascade-RCNN and GA-Faster-RCNN by 12.75% and 15.58%, respectively.

Table 1 Performance comparison with baselines

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	0.545	0.708	0.602	0.523	0.634	0.395
Faster-RCNN	0.584	0.726	0.647	0.588	0.627	0.430
GA-Faster-RCNN	0.597	0.739	0.654	0.605	0.637	0.429
Cascade-RCNN	0.612	0.734	0.665	0.612	0.657	0.456
CRPN-Faster-RCNN	0.638	0.766	0.696	0.638	0.687	0.487
MAD	**0.690**	**0.801**	**0.753**	**0.677**	**0.752**	**0.536**

The best results are in bold. The subscripts "50" and "75" represent that the IoU thresholds are 0.50 and 0.75, respectively. The subscripts "S," "M," and "L" represent area$\leq 32 \times 32$, $32 \times 32 <$ area $\leq 96 \times 96$, and area$> 96 \times 96$, respectively

Major results (Cont'd)

The spatial fusion (SF) strategy enriches the original UI image with spatial features from layer boundaries at the pixel level, which appears to be more efficient for the backbone to extract the semantic features. The results show that the SF strategy on all three models improves mAP by at least 8.27% over the FF strategy.

Table 2 Comparison of two fusion strategies

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet+FF	0.556	0.745	0.633	0.553	0.636	0.272
RetinaNet+SF	**0.602**	**0.754**	**0.679**	**0.580**	**0.682**	**0.435**
Faster-RCNN+FF	0.495	0.690	0.583	0.505	0.563	0.170
Faster-RCNN+SF	**0.622**	**0.757**	**0.697**	**0.614**	**0.683**	**0.442**
MAD+FF	0.607	0.763	0.680	0.611	0.659	0.378
MAD+SF	**0.674**	**0.797**	**0.741**	**0.661**	**0.736**	**0.524**

The better results are in bold. The subscripts "50" and "75" represent that the IoU thresholds are 0.50 and 0.75, respectively. The subscripts "S," "M," and "L" represent area$\leq 32 \times 32$, $32 \times 32 <$ area $\leq 96 \times 96$, and area$> 96 \times 96$, respectively. FF: feature fusion; SF: spatial fusion

Major results (Cont'd)

Merging results of associative fragmented layers and an example of generated DOM tree with and without UILM.

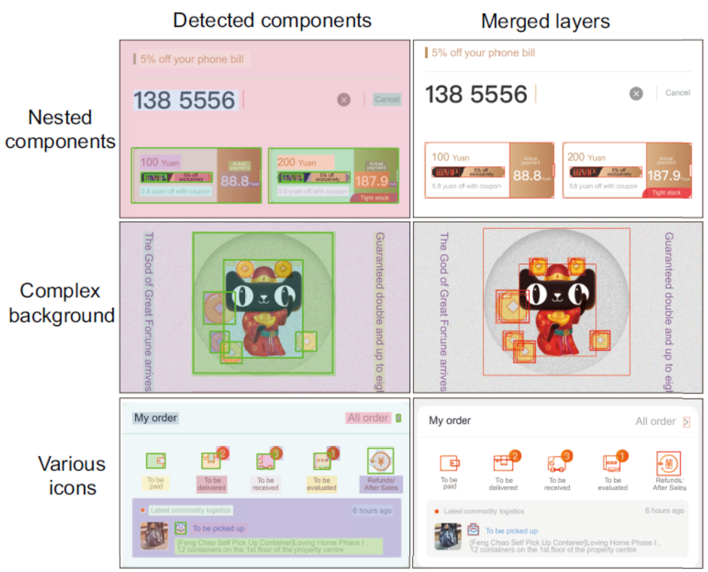


Fig. 10 Examples of associative layer merging results (the green box represents the detected components, and the red box represents the layers to be merged). References to color refer to the online version of this figure

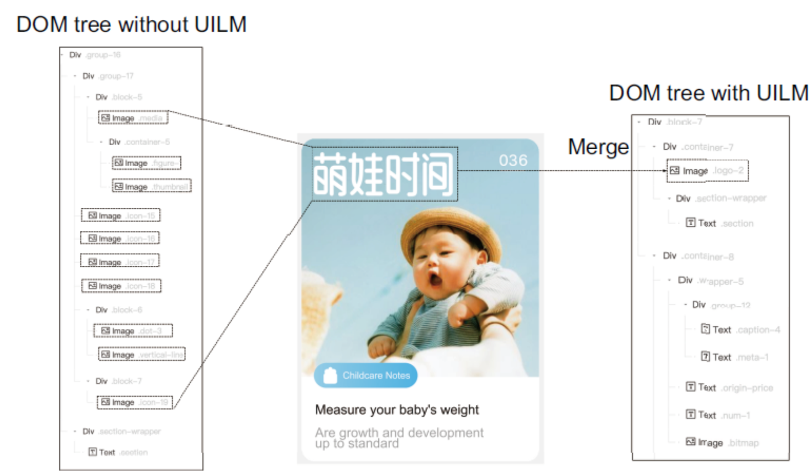
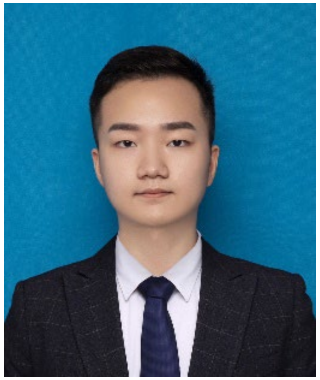


Fig. 11 An example of a generated DOM tree with and without UILM. In the DOM tree without UILM, the dashed boxes represent redundant containers. In the DOM tree with UILM, the single image container represents the "text-like" UI component (DOM: document object model; UI: user interface; UILM: UI layers merger)

Conclusions

- We investigated a novel issue concerning layer merging in an automatic design draft to the UI view code process.
- To solve this issue, we proposed UILM by detecting the areas of UI components and merging the fragmented layers into UI components.
- The evaluation results showed that our method significantly outperforms all baselines on UILM datasets. A human evaluation study also confirmed that the generated code by UILM tends to be more readable, maintainable, and useful.



Yunnong CHEN received his BS degree in mechanical engineering from Chongqing University of Posts and Telecommunications, Chongqing, China in 2021. He is currently pursuing the PhD degree in computer engineering at Zhejiang University, Hangzhou. He participated in several research projects, including National Natural Science Foundation of China and National Key R&D Program of China. His current research interests include intelligent front-end, computational creativity, and human-computer interaction.

Dr. Liuqing CHEN is the ZJU-100 Young Professor at the College of Computer Science and Technology, Zhejiang University, since September 2020. He is the principal investigator or investigator of several research projects, including National Natural Science Foundation of China and National Key R&D Program of China. Before joining ZJU, he was a Chief Data Scientist in Verchable Ltd. (UK). He is a UK's Exceptional Talent visa holder. His research Interests include: design intelligence, computational creativity, UI design, human-computer interaction, and AR/VR/XR.