BACHELOR THESIS EXPOSÉ

Optimizing perceived aesthetics of UIs using metric guided generative pipelines

MORITZ WÖRMANN

May 2, 2024

Variational Autoencoder Optical Character Recognition

1 MOTIVATION AND BACKGROUND

Through User Interface (UI) design tools like Figma [1], it gets easier to create UIs for apps and websites for users to interact with. However, designing visually pleasing UIs still proves to be a complicated task, especially since these are highly subjective categories [21]. This challenge becomes even more significant when considering the impact initial impressions of a UI can have on the users perception and on the willingness to stay on the website or the mobile app [5]. Currently, designing a new UI is often a task for multiple teams with different professions, like graphic design and software engineering. Automating the task of creating UIs or providing assistance through automatic algorithms is therefore a worthwhile topic. An "end-to-end" process of creating UIs, or at least optimizing existing ones, can reduce time and effort.

One of the most important tasks in the UI design process is the task of layouting. Layout generation tasks describe the challenge of aligning different elements and components of user interfaces as well as controlling other parameters like font, font size and color in a visually pleasing way. While this task is evidently sufficiently difficult for humans, assigning this task to algorithms proves to be even more difficult as the challenge of defining what is considered visually pleasing for humans is not straight forward. [12]

2 PROBLEM DESCRIPTION

The objective of this research is to develop a comprehensive methodology that can be utilized by a potential user to input an image of an existing UI and provide additional instructions. This automated pipeline should be capable of segmenting the UI into its components and rearranging them in a more visually appealing manner. This segmentation process can be conceptualized as a transformation or mapping of the UI elements into a distinct space, which might be referred to as a latent space. An algorithm operates in this space and retrieves feedback from a model, which predicts and assesses visual appearance. This model has been pretrained on a dataset in which users have been interrogates for their perceived aesthetics of user interfaces.

It remains to be shown, if this classifier model can predict directly from this latent space or if the user interface has to be transformed out of this latent space again first. Clearly, this transformation proves to be a an additional challenge as well as deciding the size and dimension of the latent

1 https://www.figma.com/.

space. This research aims at exploring different approaches as to how these latent spaces can look with a focus on them being able to be used in a pipeline from the latent space to an aesthetics predictor, while ensuring differentiability in order to leverage common gradient-descent patterns for this task. As past research has shown, having automated models that try to optimise a function can lead to accidental adversarial attacks, also known as "Reward Hacking" [19]. It is probable that a similar issue will arise in this research project. Therefore, it is necessary to devise a strategy to circumvent this problem.

To solve these questions the following research questions will be answered:

RQ 1 How to best segment UIs in a way such that the layout can be optimized and reassembled?

RQ 2 What characteristics should a suitable latent space possess in order to be utilized for the projection of user interfaces (UIs), which can then be optimized and graded by an automatic classifier?

RQ 3 How can (accidental) adversarial attacks by the optimizer against the Aesthetic Predictor be avoided?

RQ 4 Do Diffusion Models provide advantages (like more degrees of freedom, measured by more complex generations), either via Pix2Pix optimization or via a different latent space which represents the UI?

RQ 5 How can the generation be constrained with user supplied input on high-level positional relationship between UI elements?

RQ 6 How can a finished tool look which supports a potential user in a meaningful way during the layouting and design process?

3 STATE-OF-THE-ART

In the following, we divide the current state-of-the-art intro three main parts: (1) Segmentation of UIs (2) Efforts to optimize UIs and their layout using non-diffusion based approaches (3) Efforts to optimize UIs using diffusion based approaches

3.1 *Segmentation of UIs*

Multiple different approaches to segmenting user interfaces have been explored and proven to be viable. For the training and evaluation part of this research, a pre-segmented dataset is useful. A dataset that serves this purposes has been presented by RICO [4]. Here, the segmentation is done via directly processing the semantics of the source code of UIs. Buttons, images and other elements are defined somewhere in the source of an app and through automatic agent based exploration of android apps, this segmentation is relatively straight forward. Purely optical based segmentation is also a thoroughly explored area. Approaches that leverage classic OCR (ocr) and combine it with modern object detection algorithms like YOLO [20] show promising results in segmenting user interfaces into their individual components [2].

3.2 *Efforts to optimize UIs using non-diffusion based approaches*

While modern generative models often leverage diffusion based approaches, a number of non-diffusion based approaches also exist. Recent research has shown large advancements while not (only) relying on diffusion based approaches like in 2022: Kong et al. [11] which show how a layout transformer model can be used to reliably generate missing attributes from their latent space. This space is comprised of different elements, labeled with their category and their respective positioning on user interfaces. A similar approach called LayoutTransformer is presented in LayoutTransformer: Gupta et al. [7]. This approach leverages self-attention to assist with the generation and even allows for generation in the 3D space. While these approaches may show competitive results in unguided layout u8generation, the topic of allowing the user to give high-level constraints such as predefined relationships between UI elements (e.g. ensuring the company logo is always on top) remains challenging and not explored in the same depth.

A different approach is the usage of a VAE (vae) like in Jiang et al. [10]. In their work, the authors propose a novel approach to segmenting the user interface into different regions. This approach involves "filling out" these regions with other user interface segments in order to combat the challenges of high-level relationships in user interfaces, which are difficult for these models to process. This research builds on Arroyo et al. [1] which initially proposed the usage of vaes for layout generation tasks. Such VAE approaches have also been explored in Xie et al. [22] and Patil et al. [16].

Still, non-VAE approaches also exist, e.g. leveraging advantages of Graph neural networks which allow for refinement of initial user controlled relationship definement like in H.-Y. Lee et al. [13].

3.3 *Efforts to optimize **uis!** (**uis!**) using diffusion based approaches*

Although not directly related to **uis!**, research has already been conducted in the field of metric-based optimization for Pix2Pix approaches. These approaches explore modifying images using diffusion models[17], effectively using an already existing image as the starting point in the latent space. Multiple research efforts have demonstrated that a simple gradient descent pipeline with a classifier directly attached to it can optimize a prompt embedding that is passed to a stable diffusion model, as shown in their diffusion models [3, 23].

Another approach is to not rely on Pix2Pix models, but instead use a different autoencoder to transform the UIs into the latent space.

Deka et al. [4] already showcased an autoencoder which reduces the dimensions of a user interface layout to a 64-dimensional vector which can later be used to retrieve the layout representation again. While they did not add diffusion to their latent space, it still is a promising approach. To close the gap from this reasearch to a finished pipeline, like it is the goal of this thesis, the component which transforms the generated layout (e.g. a picture) into a real UI (e.g. markup) is however still missing. Nevertheless, this approach could provide useful insights if be possible to use this autoencoder directly in a diffusion model.

However, all of the approaches presented that operate on user interfaces do not use a true aesthetic predictor as their metric, but rather more technical metrics that measure details such as overlap on design components ([24]).

4 PROPOSED APPROACH

As the overall goal of this research is to optimize perceived aesthetics, a clear way to measure this metric is needed. As all of the approaches will rely on the usage of a common gradient descent pipeline on one way or another, this metric needs to measured in a way which is differentiable. For simplicity, the same aesthetics predictor model will be used for all of the different approaches, which is the one presented in 20203: Leiva et al. [14].

The new research in this work is that we are seeing the aesthetics predictor as part of the generation instead of using technical metrics to assess the goodness of created user interfaces after the creation has already been finished.

Table 1: Overview of the Proposed Approach

Research Questions & Related Study Phase		
RQ 1: Segmentation	RQ 2: Latent space	RQ 3: Adversarial attacks
Tasks		
○ Assess state-of-the-art ○ Evaluate necessary changes & adaptations	○ Comparison of common gradient-descent pipeline with naive positioning vector as latent space to SOTA approaches	○ Evaluation if these (accidental) attacks do in fact prove to be a challenge ○ Exploration of different mitigation approaches
Expected Results		
○ Assessment of ”Fitness” of SOTA ○ Finished Segmentation pipeline able to be differentiated in reassembling phase w.r.t. positioning vector	○ usable decision for following quesitons	○ Selection of specific mitigation tactic if applicable
Research Questions & Related Study Phase		
RQ 4: Introduction of diffusion	RQ 5: User Input constraints	RQ 6: Usable e2e pipe
Tasks		
○ Assess SOTA layout diffusion approaches ○ Assess Pix2Pix with common gradient descent pipeline	○ Exploration of different constraining approaches	○ Incorporation of prior results into usable product
Expected Results		
○ Evaluation of diffusion based approaches ○ Pivot to explored approaches in case of superiority	○ Incorporation in prior results if applicable	○ Figma plugin or ○ demo app

4.1 *Research Question 1*

Past Research has shown that UI segmentation is a task which is manageable by state of the art algorithms [4]. It has been proven that optical segmentation into text and non-text elements (by mere masking of the affected regions) can be used to train an AutoEncoder architecture which reliably reduces the dimension of the information in a user interface [4]. As this research is exploring a similar question (transforming a user interface into a latent space), a similar approach might provide good results for this task. It remains to be shown how big the influence different segmentation approaches are on the final pipeline. The maturity and reliability of models like the one presented by Deka et al. [4] suggests that this effect may be minimal.

4.2 *Research Question 2*

As previously described, the main challenge in this domain will be developing an entire pipeline, that includes a latent space, a function which retrieves the user interface out of this latent space and a predictor that determines the aesthetics for this retrieved and modified user interface. One such space might just be a vector of coordinates where the segments are placed on the user interface.

Once such a space and pipeline has been found, the representation of the user interface can be improved by utilizing common gradient descent patterns provided by major machine learning frameworks, provided that the whole pipeline actually converges.

4.3 *Research Question 3*

This part of the research is arguably the most important one. Keeping the pipeline from becoming too volatile or quickly "learning" how to exploit the aesthetics predictor and thus creating an adversarial attack is a complex task. These exploits might lead to undesirable results, where user interfaces may show extreme or small changes for no apparent reason which might lead to a higher predicted aesthetic score but, do in fact not show the same favorability during interrogation through humans. Adversarial attacks (also known as adversarial examples) have long been documented, especially in regards to humans being unable to detect them [8].

The research area of preventing adversarial attacks has been thoroughly studied due to the ubiquitous nature of classifying neural networks [15].

4.4 *Research Question 4*

This research question can be divided into two distinct tasks. The first task involves identifying an appropriate latent space in which the user interface can be projected. The second task assumes that satisfactory results can be achieved by solely relying on Pix2Pix approaches, such as StableDiffusion. [17]. This would necessitate the finetuning of the AutoEncoder in the StableDiffusion model to adapt to UIs [18]. However, this approach may prove challenging, as these models are notoriously difficult to control, which has led to the emergence of a distinct research field, prompt engineering [6]. It is perhaps overly optimistic to suggest that Pix2Pix optimization can be used to create a user interface that is both functional and aesthetically pleasing.

Thus, the first approach, (finding a latent space that has been adapted to the UI usecase) could show improved results. For this, some research has already been done, for example by 2023: Hui et al. [9] who designed the latent space such that it only holds information about the layout of a user interface.

4.5 *Research Question 5*

Constraining the generated layouts and **uis!** has been the subject of past research [13]. While most of these efforts rely on giving the constraints at the start of the pipeline, e.g. developing relationships and going from there on to the user interfaces, another approach could be to penalize a model/pipeline for a undesirable results which may include user defined constraints. It would then be entirely up to the model to grasp these constraints and work them into the predictions.

4.6 *Research Question 6*

As soon as a viable solution for the described problem has been found, integrating this solution in an appealing way for potential users becomes an additional challenge. The question then arises about how a finished user interface, which is still in a graphic representation at this stage, can be transformed back out of the latent space.

As the main objective is to optimise existing user interfaces, the main challenge will be to transform such an existing user interface into a representation which can be used by the pipeline and, arguably more important, retrieving the user interface back from the pipeline. One such "starting point" might be rendered markup code in a tool like Figma. For a segmentation and rearrangement task, this might mean the association between rendered elements and the respective code parts which produce these segments. For this taks, the final extraction stage would have to have an understanding of how layering and ordering works in these kinds of markup languages.

To measure the results of this finished pipeline, already finished user interfaces could be rearranged in a way that is subjectively not aesthetic. The modified UI can be fed into the pipeline and differences between the original user interface and what the pipeline came up with can be used to grade the performance. Of course, this relies on the assumption that the original UI was already achieving a relatively high aesthetic score.

5 PROPOSED EXPERIMENTS

The following section presents the proposed experiments designed to address the aforementioned questions.

5.0.1 *Experiment 1*

1. Use a state of the art UI dataset containing screenshots like Rico [4] and segmenting technique to segment a picture of a user interface into its elements
2. Store coordinates of the individual elements in the user interface in a vector ($\rightarrow$ latent space)
3. Reassemble the user interface according to the coordinates
4. Pass the user interface in a state of the art aesthetics predictor ([14])
5. Ensure that the resulting score is differentiable w.r.t to the coordinate vector

5.0.2 *Experiment 2*

1. Extend the Setup from Exp. 1 to include an optimiser which tries to increase the score by optimising the coordinate vector

5.0.3 *Experiment 3*

1. Apply the process from Exp. 1 to a meaningful number of user interfaces to check for potential issues, such as the described accidental adversarial attacks

5.0.4 *Experiment 4*

1. Add additional parameters to the latent space which can be optimised, such as
   - Background color, size of the elements

5.0.5 *Experiment 5*

1. Package Exp. 1-4 into a Proof of concept pipeline, which just receives a user interface and outputs a new, optimised, user interface

5.0.6 *Experiment 6*

1. Fine-tune an Autoencoder to be part of a Diffusion model/pipeline for User interfaces
2. Use the same predictor as in Exp. 1 to optimise an existing user interface using a Pix2Pix approach
3. Compare the results to the ones from Exp. 1-4

Following the described approach, the structure shown in fig. 1 is proposed for the thesis.

Figure 1: Proposed Structure

1. Introduction
   1.1 Motivation
   1.2 Problem Statement
   1.3 Structure of the Work
2. Background
   2.1 Current approaches & challenges
3. State-of-the-Art and Related Work
   3.1 Non-Diffusion based approaches
   3.2 Diffusion based approaches
   3.3 Current shortcomings
4. Proposed Method and Implementation
   4.1 Implementation of described approaches
   4.2
   4.3 Final Application
5. Evaluation
   5.1 Evaluation Method
   5.2 Comparison to prior work
   5.3 Use Cases
6. Results
7. Discussion
   7.1 Threats to Validity
   7.2 Future Work

[1] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation, 2021.

[2] Jieshan Chen, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li. Object detection for graphical user interface: old fashioned or deep learning or a combination? In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 1202–1214, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Niklas Deckers, Julia Peters, and Martin Potthast. Manipulating embeddings of stable diffusion prompts, 2023.

[4] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, page 845–854, New York, NY, USA, 2017. Association for Computing Machinery.

[5] Maria Douneva, Rafael Jaron, and Meinald T. Thielsch. Effects of Different Website Designs on First Impressions, Aesthetic Judgements and Memory Performance after Short Presentation. *Interacting with Computers*, 28(4):552–567, 06 2016.

[6] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models, 2023.

[7] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention, 2021.

[8] Jan Philip Göpfert, André Artelt, Heiko Wersing, and Barbara Hammer. *Adversarial Attacks Hidden in Plain Sight*, page 235–247. Springer International Publishing, 2020.

[9] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model, 2023.

[10] Zhaoyun Jiang, Shizhao Sun, Jihua Zhu, Jian-Guang Lou, and Dongmei Zhang. Coarse-to-fine generative modeling for graphic layouts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):1096–1103, Jun. 2022.

[11] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. Blt: Bidirectional layout transformer for controllable layout generation, 2022.

[12] Talia Lavie and Noam Tractinsky. Assessing dimensions of perceived visual aesthetics of web sites. *International Journal of Human-Computer Studies*, 60(3):269–298, 2004.

[13] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints, 2020.

[14] Luis A. Leiva, Morteza Shiripour, and Antti Oulasvirta. Modeling how different user groups perceive webpage aesthetics. *Universal Access in the Information Society*, 22(4):1417–1424, Nov 2023.

[15] Qi Liu and Wujie Wen. Model compression hardens deep neural networks: A new perspective to prevent adversarial attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):3–14, 2023.

[16] Akshay Gadi Patil, Omri Ben-Eliezer, Or Perel, and Hadar Averbuch-Elor. Read: Recursive autoencoders for document layout generation, 2020.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[18] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023.

[19] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 9460–9471. Curran Associates, Inc., 2022.

[20] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, November 2023.

[21] Christiane G. von Wangenheim, João V. Araujo Porto, Jean C. R. Hauck, and Adriano F. Borgatto. Do we agree on user interface aesthetics of android apps?, 2018.

[22] Yuxi Xie, Danqing Huang, Jinpeng Wang, and Chin-Yew Lin. Canvasemb: Learning layout representation with large-scale pre-training for graphic design. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4100–4108, October 2021.

[23] Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jia Chen, and Shaoping Ma. Capability-aware prompt reformulation learning for text-to-image generation, 2024.

[24] Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models, 2023.