Exemplar Based Experience Transfer

Paridhi Maheshwari
Adobe Research
Bangalore, India
parimahe@adobe.com

Nitish Bansal
Indian Institute of Technology
Roorkee, India
nitishbansal2297@gmail.com

Surya Dwivedi
Indian Institute of Technology
Kharagpur, India
surya2191997@gmail.com

Rohan Kumar
Indian Institute of Technology
Kanpur, India
rohank@iitk.ac.in

Pranav Manerikar
Ohio State University
Columbus, OH, USA
maneriker.1@buckeyemail.osu.edu

Balaji Vasan Srinivasan
Adobe Research
Bangalore, India
balsrini@adobe.com

ABSTRACT

Banners are present in several forms and a person might be inspired by one or more of these. However, designing banners is a non-trivial task, especially for novices. Starting from a blank canvas can often be overwhelming, and exploring alternatives is time-consuming. In this paper, we propose an automatic approach to transfer a novice user's content into an example banner. Our algorithm begins with extracting the template of the example banner via a semantic segmentation approach. This is followed by an energy-based optimization framework to combine multiple design elements and arrive at an optimal layout. A crowd-sourced experiment comparing our automatic results against banners designed by creative professionals indicates the viability of the proposed work.

CCS CONCEPTS

• **Applied computing** → **Graphics recognition and interpretation**; • **Human-centered computing** → *HCI design and evaluation methods*;

KEYWORDS

style transfer, single page graphic design, banner design, design automation

ACM Reference Format:
Paridhi Maheshwari, Nitish Bansal, Surya Dwivedi, Rohan Kumar, Pranav Manerikar, and Balaji Vasan Srinivasan. 2019. Exemplar Based Experience Transfer. In *24th International Conference on Intelligent User Interfaces (IUI '19), March 17–20, 2019, Marina del Rey, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3301275.3302300

1 INTRODUCTION

In the digital era, there exists a plethora of designs and banners in the form of flyers, advertisements, hoardings etc. These banners have become important means of visual communication to mass audiences. Designing and refining banners requires conveying the content in a succinct and visually pleasing manner. This is a non-trivial task even for experienced designers and creating banners from scratch is a time-consuming process. The traditional approach is to leverage existing designs as building blocks to develop new ones. Therefore, creative professionals often use curated example galleries that serve as inspirations for designing alternatives [7, 13].

Given the ubiquitous nature of such designs, a novice user might be inspired by the designs of banners they interact with and would want to use them with their own content. However, this becomes a tedious task if the corresponding template is unavailable. Even if the template is available, transforming them based on his content while simultaneously satisfying aesthetic intent requires professional skills that involve an understanding of the factors that contribute to banner aesthetics. In the absence of appropriate automation tools, such processes are quite challenging for novices.[1]

In this paper, we propose a two-fold technique to automate the transfer of user's content into the design(s) of exemplar banner(s). The proposed algorithm extracts the design template by performing a semantic segmentation of the input banner image. These segments are then used to identify the underlying design elements (such as the image/text/shape placeholder locations, size, etc.) using a region-based clustering approach yielding an editable template where the user can insert his content to achieve the experience transfer. The final step of our algorithm is to identify the salient aspects of a banner from a set of sample corpus and utilize them to fine tune the layout for better overall aesthetics via an energy-based optimization.

2 RELATED WORK

One line of work that is relevant to our problem deals with **extracting templates** from images. Betramelli et al. [4] is aimed at extracting templates of interfaces from its screenshots using an RNN-based architecture. However, their approach was restricted to simpler interfaces with lesser degrees of freedom and obtains only approximate estimates of the elements' locations. To be able to produce an editable template, accurate locations of all design elements of the banner is required.

Closely related to template extraction are works on **segmentation and object detection**. Semantic segmentation is the task of classifying each pixel of an image into various types/categories. Long et al. [20] proposed Fully Convolutional Neural Networks (FCNN) for semantic segmentation and showed that convolutional

IUI '19, March 17–20, 2019, Marina del Rey, CA, USA

ACM ISBN 978-1-4503-6272-6/19/03…$15.00
https://doi.org/10.1145/3301275.3302300

[1] Work done when authors were at Adobe Research

networks trained end-to-end exceed the state-of-the-art in semantic segmentation. Semantic segmentation, however, is limited to identifying the semantics of the image at a pixel level and does not directly yield templates requiring additional processing to extract design element information from the identified segments. Object detection algorithms, on the other hand, directly yield bounding boxes and can be useful in finding the locations of the detected design elements. Notable works in object detection include Regions with CNN features [12], its variants like Fast R-CNN [10], Faster R-CNN [26], Mask R-CNN [12] and R2CNN [16]. Mask R-CNN [12] first identifies the semantic regions of various objects followed by identifying the bounding boxes (masks) corresponding to these regions. Due to the recent successes of the Mask R-CNN and FCNN frameworks as strong baselines in segmentation and detection tasks, we explore both these frameworks for extracting the templates of banners.

Another line of related works involve **optimization of layouts**. A growing body of recent work focuses on automating this process in the context of website and documents. Bricolage [17] automates the website creation process by transforming the content of one webpage into the style and layout of another. This is achieved by creating correspondences between the Document Object Model (DOM) elements of two webpages. Such a technique cannot be employed for banners due to the absence of a DOM tree structure. In interface designing, Gajos et al. [9] specify the positions and categories of widgets and use a margin-based approach to set weights of the objective functions to learn model parameters. Swearngin et al. [27] introduce Rewire, a system that automatically reconstructs vector representations from screenshots of interfaces and leverages them to provide assistance to designers in creating new designs. Their reconstruction process involves low-level image processing based on UI-specific techniques and can not be generalized. Todi et al. [28] propose a layout restructuring of websites to make it more familiar to users and aid navigation. Adaptive layouts for documents have used grid-based templates for individual elements [14], probabilistic models for document composition [8] and genetic algorithm framework to personalize documents [25]. However, they deal with simple documents primarily involving text that support a linear read-order and easily conform to templates. Banners, however, are less structured and composed of free-form placement of text and image elements. Such semantic and structural differences between banners and the experience synthesized in these explorations make these approaches not trivially extendable for banner re-purposing.

The sub-field of layout problem most similar to our problem is the **optimization of single-page graphic designs**. Early works in this domain was automatic designing of magazine covers by analyzing visual saliency of photographs to position text [15]. A common approach in layout structuring is to define measures of aesthetic requirements and optimize layouts to minimize these measures. Various energy functions to quantify alignment of documents [2, 3], visual weight and balance of layouts [19], aesthetics of interfaces [22] and heuristics like balance and uniformity of documents [11] have been put forth. Vollick et al. [29] proposed an energy-based approach to model layouts of labels in technical diagrams. Non-linear inverse optimization [18] have been commonly explored to learn model parameters from training data. Label layout, however, is a relatively simpler problem where text elements are distributed around a background image. Zhang et al. [32] proposed techniques to automatically generate banners of different sizes adhering to learned style parameters. But they primarily work on reflowing the banner content into different sizes and do not deal with optimization for news content. Along these lines, Donovan et al. [24] introduced an energy-based model derived from design principles to synthesize different versions of single-page graphic designs and have extended their method to a user-interactive mode [23]. We deploy the energy-based framework of [24] for defining energy functions in the context of banners and optimizing elements from multiple banners into a single layout.

3 EXEMPLAR BASED EXPERIENCE TRANSFER

Our proposed algorithm of transferring a user's target experience into an exemplar banner consists of two key steps. The underlying design elements (text boxes, shapes and images) are first extracted from the banner images to build an editable template where the user can add his own content. For this, we explore two alternate configurations. The first configuration is based on an initial *semantic segmentation* of the banner image into salient regions via a fully convolutional neural network (FCNN) [20] followed by utilizing the segmentation to *extract key design elements*. Our alternative configuration employs the Mask R-CNN [12] framework to extract the key elements in the input banner. Similar to FCNN, Mask R-CNN starts with a Region Proposal Network that segments the different regions in the input. In addition to identifying the class of each region, the Mask R-CNN also identifies the bounding boxes for the identified region, thus yielding the design elements directly. Both these configurations facilitate the extraction of the underlying template of the banner image and enable replication of an exemplar banner with different content.

Once the template is extracted, it is possible to insert the user content into the template by matching the type of content (image to image fields, and so on). However, since the content is different, it might result in misalignment and overflows thus affecting the overall aesthetics of the banners. To address this, the second step of our algorithm tunes the content-filled-elements to produce a banner that satisfies various aesthetic goals. We formulate this as an optimization problem where a set of energy functions are introduced to quantify the aesthetics of a template. The energy functions are smoothed and a weighted sum is used towards defining the overall 'goodness' of a banner. The weights of the energy functions are learned via a non-linear optimization [18] based on the corpus of exemplar banners. The energy function thus learned is used in a simulated annealing setup to optimize the new layout of the banner with replaced content. Figure 3 shows the complete schematic of the sequence of steps in the proposed framework.

4 TEMPLATE EXTRACTION

Given an image of the banner, template extraction identifies the design elements of the banner along with their positional details that can be used to transfer the user's content into the banner. We explore two state-of-the-art frameworks in image segmentation (Fully Convolutional Neural Networks) and object detection (Mask R-CNN) for identifying the elements of the input banners.

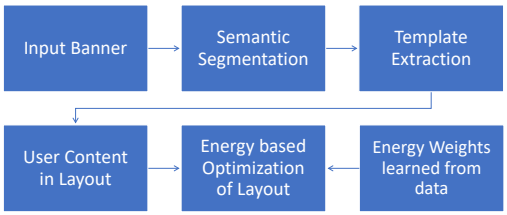

Figure 1: Schematic of the proposed approach

4.1 Fully Convolutional Neural Network (FCNN)

Unlike a fully-connected network, FCNN is capable of retaining spatial information and can easily extend to inputs of arbitrary sizes. FCNN layers are of the form,

$$y_{ij} = f_{ks}\left(\{x_{s_i+\delta_i, s_j+\delta_j}\}_{0\leq\delta_i,\delta_j\leq k}\right)$$

Where x is the input to the corresponding layer and y is the output of the layer, k is the kernel size and s is the stride/subsampling factor. f_{ks} determines the type of the layer. f_{ks} could be matrix multiplication for convolution, or average pooling and spatial max for max pooling, or an element-wise nonlinear function for activation, depending on the type of the layer. The end-to-end network is optimized to minimize the pixel-wise cross-entropy loss between the input banner and the corresponding output annotation.

To accurately represent the design elements, the extracted template should specify the rectangular bounding box for each of the elements. Since the output of FCNN does not directly yield this, the semantic segmentation output is processed to extract the bounding boxes for each identified element. Moreover, the layout of banners may have overlapping elements (for example, text on top of shape) while semantic segmentation assigns a single label to each pixel. We, therefore, devise a mechanism based on connected components to process the segmentation output and extract the exact position, size and other meta details of the underlying design elements.

From every segmented pixel, we deploy a Depth-First Search (DFS) around its neighbourhood to identify a bounding box for the corresponding design element encompassing the current pixel. Since the bounding box search does not limit the selection of a pixel in multiple elements, this allows for overlapping elements to be identified. Bounding boxes less than a threshold (in size) were rejected to control the noise in semantic segmentation. The algorithm outputs a set of elements in the banner along with the respective bounding boxes and location, yielding the final template with the required layout information. Algorithm 1 summarizes the sequence of steps to extract the template from the semantically segmented banners.

4.2 Mask Region-based Convolutional Neural Network

The Mask Region-based Convolutional Neural Network (Mask R-CNN) [12] extends from Faster Region-based Convolutional Neural Network [26] and adds an additional branch for predicting segmentation 'masks' on every region of interest (RoI), along with classifying the region. The mask layer is based on a Fully Convolutional

Algorithm 1 Design Element Extract

Input I = Image output of semantic segmentation
Initialize $L = \emptyset$
while there is an unvisited pixel **do**
    Run DFS from the unvisited pixel N to find a connected component C
    Maintain the 4 points of C closest to the 4 corners of I in **Box** while running DFS
    L.append(Box)
Filter L based on region size
return L

Neural Network applied to each RoI, predicting the segmentation mask in a pixel-to-pixel manner. Thus, Mask R-CNN yields the bounding boxes for every identified segment and unlike FCNN, additional processing is not required to extract these boxes. To allow for overlapping elements, we followed [12] to independently classify the masks for each identified regions. Further, Mask R-CNN also decouples the mask-identification from the region classification, resulting in smoother masks (and hence bounding boxes).

4.3 Evaluating Template Extraction

In order to train our frameworks, we used a dataset of ~ 140k banner images that range over a wide variety of topics and styles created by non-expert users of a design software. Every banner has an associated manifest file that includes information about the design elements in the banner, their position and orientation in the banner, and other meta-data like font type and font size for text descriptions and colour for shapes. Although there are various components in a banner, we specifically focus on modelling the layout, i.e, the positions and scales of different design elements. To train the FCNN and Mask R-CNN models, we process the manifest files and generate pixel-wise ground truth annotations for the banners by marking various regions of the banners. The dataset is divided in train (80%), validation (10%) and test (10%) sets.

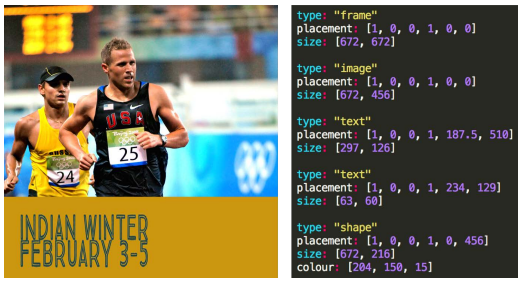

Figure 2: Sample banner from our dataset and its manifest file

Table 1 shows the performance of the various frameworks on our dataset. The FCNN based model outperforms the other settings in overall segmentation. However, since the plain FCNN model only computes the pixel-wise segmentation, it cannot be directly used as an editable template. For template extraction, it can be seen that the FCNN segmentation followed by the region-based

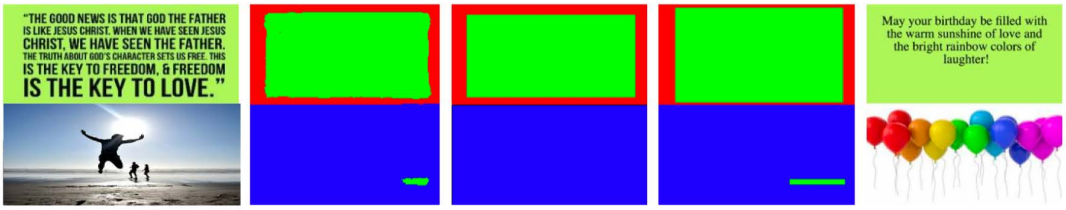

Figure 3: Example depicting the layout extraction framework (a) Input image (b) Semantic segmentation output (c) Design element extraction (d) Ground truth annotation (e) Transformation with user content

clustering via DFS outperforms the Mask R-CNN setting in terms of F1 scores although the accuracy scores are comparable between the two settings. This is perhaps due to the superior ability of FCNN in identifying the regions than the Mask R-CNN cascading to the template extraction.

	Accuracy	F1 Score
FCNN	0.83	0.61
FCNN + Template Extraction	0.80	0.58
Mask R-CNN	0.79	0.50

Table 1: Results of template extraction

Fig 3 shows the output from various stages in our template extraction and how an experience can be synthesized by transferring new content into the exemplar banners. While we have not considered extracting the font style here, existing work like [30] can easily be incorporated into our framework to recognize and transfer font styles from sample banners.

5 ENERGY-BASED OPTIMIZATION

While the element extraction yields the underlying templates for input banners, adding user's content in these template can offset the underlying design. This calls for a way to fine-tune the content-transferred banner. We extend the energy-based optimization framework of [24] for fine tuning the layouts. The framework optimizes a non-linear energy function that encapsulates various aspects of design via simulated annealing. We extended various energy functions from [24] for our purpose and learn their combination via a non-linear hyperparameter optimization [5] based on the samples in our corpus.

5.1 Energy Functions

Peter et al. [24] define a set of energy functions to measure the goodness of a layout on various dimensions. We adapt different components of these energy functions E_i for our problem. The composite energy function's hyperparameters θ comprises of $[\mathbf{w}, \alpha]$. $\mathbf{w}$ are the weights given to different energy functions in the weighted sum that yields the final energy. Each component energy function is smoothed via a sigmoid function $S(\cdot;\alpha)$ to map it to a value between 0 and 1. The parameter α is used to smoothen the sigmoid function and determine how the changes in energy should be mapped between 0 and 1. A smaller value of α makes the sigmoid function smoother, whereas a larger value takes it closer to a step function. Fig. 4 shows the mapping for different α values. The overall energy of a layout is given by a weighted sum of individual energies,

$$\mathrm{E}(\mathbf{X};\theta) = \sum_i w_i \mathrm{E}_i(\mathbf{X};\alpha_i).$$

The hyperparameters $\mathbf{w}, \alpha$ are learned via a non-linear optimization as described in the next sub-section. Here, we outline the different terms that constitute our energy.

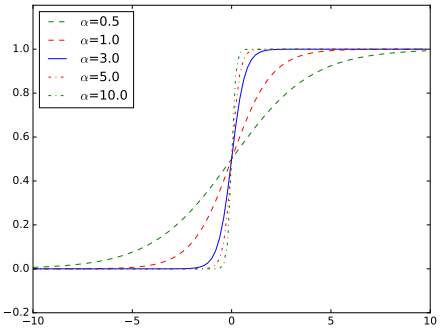

Figure 4: Sigmoid function mapping for different α

Correct **alignment** is an important aspect that portrays an organized representation, which in turn adds to the aesthetic value. We considered *Left*, *Right*, *X-center*, *Y-center* and *Bottom* alignments, which are depicted in Figure 5. The design elements of typical banners can be grouped such that the elements within a group are aligned together. The alignment energy term measures the fraction of element pairs that can be bracketed together under the same alignment type,

$$\mathrm{E}^a_{align} = -\mathrm{S}\left(\frac{1}{n^2}\sum_{i\in(\text{all})}\left(\sum_{j\in(\text{all})} I^a_{ij}\right); \alpha^a_{align}\right)$$

where n denotes the total number of elements, I^a_{ij} indicates if elements i and j are aligned by type a.

The **group energy** encourages aligned element pairs to be clustered together under a common type of alignment. This promotes symmetry in the banner and is visually appealing.

$$\mathrm{E}^a_{group} = -\mathrm{S}\left(\frac{1}{nm}\sum_g \sum_{i\in(\text{all})} I^i_g; \alpha^a_{group}\right)$$

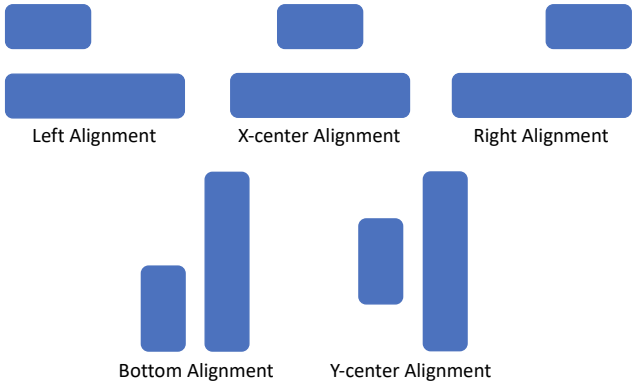

Figure 5: Types of alignments considered

where n and m are the number of elements and alignment groups respectively and I_g^i indicates if element i belongs to alignment group g.

Minor **misalignment** between two elements is visually discomforting as well as distracting. To accommodate for this, we use a misalignment energy term that heavily penalizes element pairs that are slightly off from alignment.

$$\mathrm{E}^a_{misalign} = \frac{1}{3n^2} \sum_a \sum_{i \in (\mathrm{all})} \sum_{j \in (\mathrm{all})} I^a_{ij} C(d^a_{ij})$$

Here, d^a_{ij} is a measure of the misalignment, i.e, minimum distance to align the elements and $C(\cdot)$ is the cost function. We used $C(d) = 5\arctan\left(\frac{d}{0.015}\right)$ heavily penalizing even minor misalignments.

A good layout finds a good mix of **whitespace** and limited **spread** [31]. To account for these in our energy functions, we encourage whitespace via a negative fraction of the total pixels that are occupied, given by,

$$\mathrm{E}_{whitespace} = -\mathrm{S}\left(\frac{\sum_p I_p}{wh}; \alpha_{whitespace}\right)$$

where I_p is an indicator of whether a pixel p has any element or not and (w, h) are the width and height of the image respectively.

While whitespaces are good, too much white space can be visually displeasing. We, therefore, penalize the spread via another energy component given by,

$$\mathrm{E}_{spread} = \mathrm{S}\left(\frac{1}{n^2} \sum_{i \in (\mathrm{all})} \min_{j \in (\mathrm{all})} D_{ij}; \alpha_{spread}\right)$$

where D^a_{ij} is the euclidean distance between elements i and j.

Our final energy component penalizes overlap between elements and is given by the sum of overlapping pixel area across all combination of elements.

$$\mathrm{E}^a_{overlap} = \mathrm{S}\left(\frac{\sum_p A_p}{wh}; \alpha^{overlap}\right)$$

where A_p is an indicator of any overlap at pixel p and (w, h) are the width and height of the image respectively.

5.2 Layout Learning

The objective of defining the energy functions is to measure various aspects of the layouts. While the proposed energy functions are not exhaustive, it measures various visual aspects of the banner. Due to the non-linearity of the energy function, the parameters $\theta = [\mathbf{w}, \alpha]$ can neither be estimated easily nor can be set empirically. We, therefore, use a non-linear hyper-parameter optimization framework based on randomized search [5] to learn the appropriate θ from our training corpus of designs. These parameters are updated iteratively based on a non-linear inverse optimization [18] framework.

It is assumed that $\mathbf{X}_T$ is optimal for some unknown θ and in order to estimate this parameter θ, we minimize the energy difference between the example layouts $\mathbf{X}_T$ and the optimal layout for an unknown θ. This is given by the energy function,

$$\mathrm{G}(\theta) = \mathrm{E}(\mathbf{X}_T; \theta) - \min_{\mathbf{X}} \mathrm{E}(\mathbf{X}; \theta)$$

Since the optimization is non-linear, we follow an alternating minimization approach where θ and $\mathbf{X}_T$ are updated iteratively. At each iteration, the previous θ is used to determine the optimized layout (using the method described in the next subsection) followed by determining the new θ that minimizes the $\mathrm{G}(\theta)$. The final θ is obtained by repeating this over several iterations and is used for energy computation in subsequent layout optimization.

5.3 Layout Optimization

Given a set of design elements $\mathbf{X}$ and the learned weights θ, the task is to arrange these elements in a layout with the least total energy. For this multi-variable problem involving highly coupled constraints, we use a simulated annealing approach [1, 21] since our search space is discrete. Though there are many optimization algorithms, including hill climbing, gradient descent, etc., the advantage of simulated annealing is that it avoids getting stuck in local minima/maxima even with complex optimization functions.

Zhang et al. [32] noted that initialization in such inverse optimizations play a key role and bad initialization can seriously impact the optimization. Therefore, the layout is first initialized based on tree-of-parzen-estimator [6] based optimization and its energy is computed. This gives a good initialization for faster convergence and is better than a random initialization. The algorithm then proceeds to explore various *proposals* and a move is considered to be a good one if the energy decreases from the current layout to the proposed layout. All good transitions are accepted and bad transitions are accepted with some probability. The process is repeated across several iterations to reach the layout with optimal energy. In the initial iterations, the probability of accepting higher energy layouts is more, thus avoiding local minima. This threshold is annealed (or decreased) as the optimization progresses so that only changes that lower the energy are preferred.

The following proposals were executed to adjust the elements and decrease various components of the energy function:

- Alignment: This proposal picks up two design elements randomly and aligns them on one of the alignment axes.
- Overlapping Elements: It picks up two elements and checks whether these elements have a common area and ensures that they are separated.

- Update Height/Width: This proposal alters the height/width of a randomly chosen design element.
- Update Single Element Position: This proposal randomly picks up any design element and shifts its position by a certain distance in both x and y directions.
- Swap Elements: This proposal randomly swaps the location of two elements.

5.4 Evaluating the layout optimization

Figures 6d, 6e and 6f show some sample banners from our algorithm. It can be seen that the proposed algorithm produces reasonable layouts. However, the text on top of images are not visually pleasing, since we do not consider image saliency in our energy computation - e.g. the font colour in Fig. 6d, the text overlap in Fig. 6f. While this can be addressed via our framework, defining an appropriate energy metric for saliency and corresponding proposals are a subject of further research.

To formally evaluate the proposed framework, we create a large number of machine-generated banners. We begin with identifying banners with the same number of design elements in our dataset. Retaining the content of the original banner, we optimize the other banners to arrive at different layout combinations resulting in different variants of the input banner. We deployed a human experiment to compare the goodness of the optimized banners against human-generated banners.

Every user is presented with a random banner image - either machine-generated or designer-generated. The annotators are asked to rate the alignment, overlap, spread and overall aesthetics of the banners on a 5−point Likert scale with the items *Very poor*, *Poor*, *Neutral*, *Good* and *Very good*. To maintain the quality of the annotations, workers are restricted to those with 95% acceptance over a minimum of 200 annotations. Every annotator rated a single image and are asked to describe the banner in order to further filter out arbitrary annotations. This crowd-sourced experiment is performed on 30 machine-generated and 30 designer banners. Each banner is evaluated by 5¢ different AMT workers and every successful annotation is awarded 5. The ratings are summarized in Table 2. Sample designer and machine-generated banners from our survey are shown in Figure 6.

Aspect	Alignment	Overlap	Spread	Overall
Designer	3.705	4.093	3.899	4.046
Machine	3.854	3.806	3.806	3.733

Table 2: Average user ratings on a scale of 1-5

It is evident from Table 2 that banners generated from our algorithm, which received a mean rating of 3.73, perform comparable to designer banners with a mean rating of 4.05. There are considerable variations in the assessment of specific visual qualities like alignment and overlap. The alignment score improved from designer banners to our banners whereas overlap between elements was found to be higher for machine-generated banners. The reason for this is that alignment is primarily an objective metric which depends only on the locations and sizes of the design elements, and not the content. Overlap, on the other hand, is a more subjective metric that is not independent of the content. For example, consider the text boxes of figures 6a and 6f, both of which are placed on top of images, thereby resulting in complete overlap. But the overlap in 6a is not displeasing because it is situated over the image background. In 6f however, the text interacts with the image foreground and causes uneasiness. As mentioned before, our current framework is content agnostic and taking the image saliency into account requires defining an appropriate energy metric and proposals to optimize the metric and is a subject of further research. These comparative annotations indicate that the proposed method can help in providing compelling starting points for a novice designer.

6 CONCLUSION & FUTURE WORK

In this paper, we have studied the problem of transferring the user's content into a sample banner. Our algorithm uses a semantic segmentation based framework to first extract the underlying template. Once the user-content is entered into the layouts, we have energy-based optimization to fine-tune the layouts with the user content.

While we have outlined our algorithm to take a single banner into account, our algorithm can also be extended to optimize the design elements from multiple banners. Fig 7. shows an example where elements from 2 distinct banner images were combined and optimized using our algorithm - however, we manually selected the elements from individual banners. Automatically identifying the right subset of elements from each of the banners to be combined into an aesthetic banner is part of future research.

REFERENCES

[1] Maneesh Agrawala and Chris Stolte. 2001. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 241–249.

[2] Helen Y Balinsky, Jonathan R Howes, and Anthony J Wiley. 2009a. Aesthetically-driven layout engine. In *Proceedings of the 9th ACM symposium on Document engineering*. ACM, 119–122.

[3] Helen Y Balinsky, Anthony J Wiley, and Matthew C Roberts. 2009b. Aesthetic measure of alignment and regularity. In *Proceedings of the 9th ACM symposium on Document engineering*. ACM, 56–65.

[4] Tony Beltramelli. 2018. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, 3.

[5] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13, Feb (2012), 281–305.

[6] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*. 2546–2554.

[7] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann.

[8] Niranjan Damera-Venkata, José Bento, and Eamonn O'Brien-Strain. 2011. Probabilistic document model for automated document composition. In *Proceedings of the 11th ACM symposium on Document engineering*. ACM, 3–12.

[9] Krzysztof Gajos and Daniel S Weld. 2005. Preference elicitation for interface optimization. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 173–182.

[10] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[11] Steven J Harrington, J Fernando Naveda, Rhys Price Jones, Paul Roetling, and Nishant Thakkar. 2004. Aesthetic measures for automated document layout. In *Proceedings of the 2004 ACM symposium on Document engineering*. ACM, 109–111.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2980–2988.

[13] Scarlett R Herring, Chia-Chen Chang, Jesse Krantzler, and Brian P Bailey. 2009. Getting inspired!: understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 87–96.

[14] Charles Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. 2003. Adaptive grid-based document layout. *ACM transactions on graphics (TOG)*

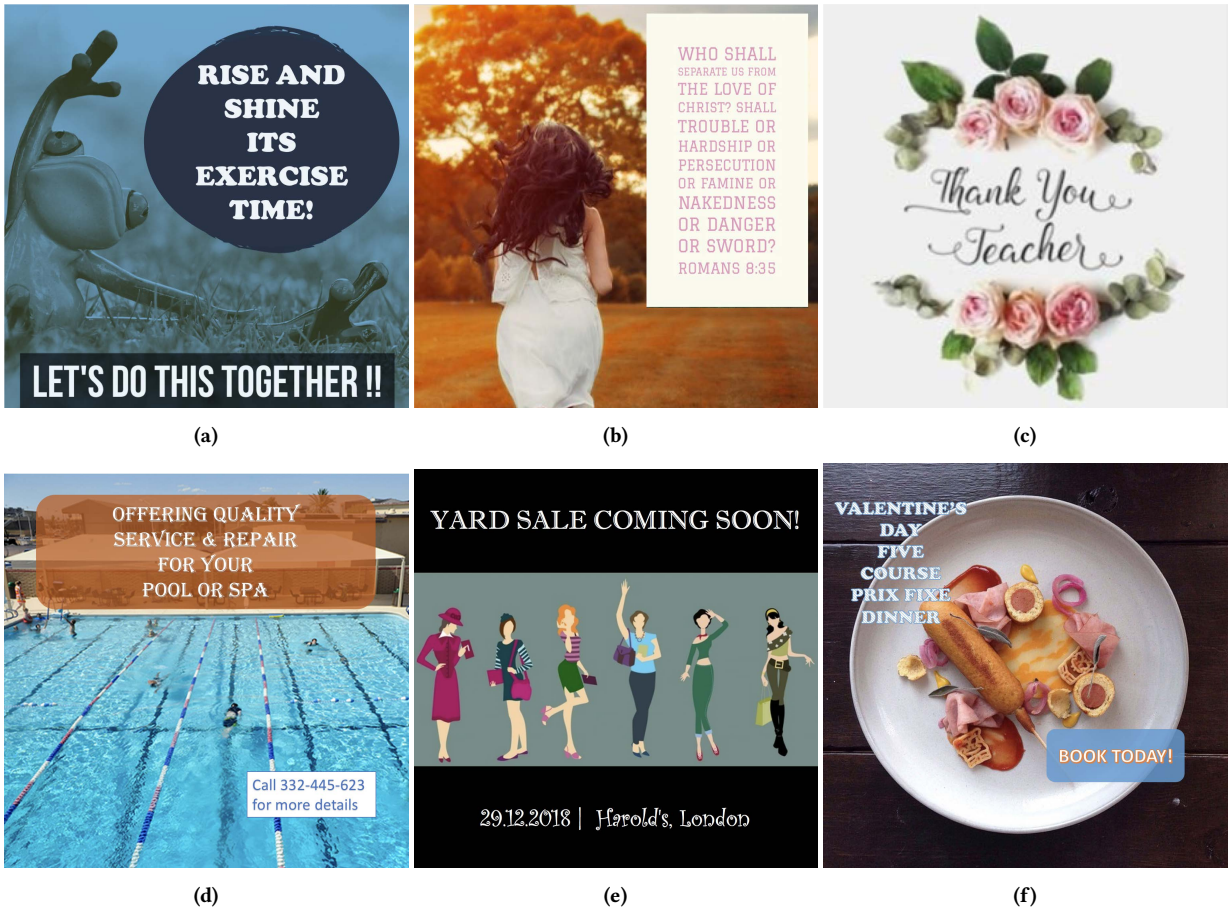

Figure 6: Examples of banners from user survey. (a-c) Banners created by designers. (d-f) Banners generated from our algorithm

(a) First input image **(b) Second input image** **(c) Combined layout with user content**

Figure 7: Example depicting the multiple banner combination

22, 3 (2003), 838–847.
[15] Ali Jahanian, Jerry Liu, Daniel R Tretter, Qian Lin, Niranjan Damera-Venkata, Eamonn O'Brien-Strain, Seungyon Lee, Jian Fan, and Jan P Allebach. 2012. Automatic design of magazine covers. In *Imaging and Printing in a Web 2.0 World III*, Vol. 8302. International Society for Optics and Photonics, 83020N.
[16] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. 2017. R2CNN: rotational region CNN for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579* (2017).
[17] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. 2011. Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems. ACM, 2197–2206.

[18] C Karen Liu, Aaron Hertzmann, and Zoran Popović. 2005. Learning physics-based motion style with nonlinear inverse optimization. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1071–1081.

[19] Simon Lok, Steven Feiner, and Gary Ngai. 2004. Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interfaces*. ACM, 101–108.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[21] Zbigniew Michalewicz and David B Fogel. 2013. *How to solve it: modern heuristics*. Springer Science & Business Media.

[22] David Chek Ling Ngo, Lian Seng Teo, and John G Byrne. 2002. Evaluating interface esthetics. *Knowledge and Information Systems* 4, 1 (2002), 46–79.

[23] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1221–1224.

[24] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics* 20, 8 (2014), 1200–1213.

[25] Lisa Purvis, Steven Harrington, Barry O'Sullivan, and Eugene C Freuder. 2003. Creating personalized documents: an optimization approach. In *Proceedings of the 2003 ACM symposium on Document engineering*. ACM, 68–77.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[27] Amanda Swearngin, Mira Dontcheva, Wilmot Li, Joel Brandt, Morgan Dixon, and Andrew J Ko. 2018. Rewire: Interface Design Assistance from Examples. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 504.

[28] Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2018. Familiarisation: Restructuring Layouts with Visual Learning Models. In *23rd International Conference on Intelligent User Interfaces*. ACM, 547–558.

[29] Ian Vollick, Daniel Vogel, Maneesh Agrawala, and Aaron Hertzmann. 2007. Specifying label layout style by example. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 221–230.

[30] Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S Huang. 2015. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 451–459.

[31] Alex W White. 2011. *The elements of graphic design: space, unity, page architecture, and type*. Skyhorse Publishing, Inc.

[32] Yunke Zhang, Kangkang Hu, Peiran Ren, Changyuan Yang, Weiwei Xu, and Xian-Sheng Hua. 2017. Layout Style Modeling for Automating Banner Design. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. ACM, 451–459.