# WordBlender: Principles and Tools for Generating Word Blends

Sam H. Ross*, Ecenaz Jen Ozmen**, Maria V. Kogan**, Lydia B. Chilton**
* Barnard College
** Columbia University
shr2118@barnard.edu,eo2419@columbia.edu,mk4036@columbia.edu,chilton@cs.columbia.edu

## ABSTRACT

Combining text and images is a powerful strategy in graphic design because images convey meaning faster, but text conveys more precise meaning. Word blends are a technique to combine both elements in a succinct yet expressive image. In a word blend, a letter is replaced by a symbol relevant to the message. This is difficult because the replacement must look blended enough to be readable, yet different enough to recognize the symbol. Currently, there are no known design principles to find the most aesthetically pleasing word blends. To establish these principles, we run two experiments and find that to be readable, the object should have a similar shape as the letter. However, to be aesthetically pleasing, the font should match some of the secondary features of the image: color, style, and thickness. We present WordBlender, an AI-powered design tool to quickly and easily create word blends based on these visual design principles. WordBlender automatically generates shape-based matches and allows users to explore combinations of color, style, and font that improve the design of blends.

## CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Computing methodologies** → *Computer vision tasks.*

## KEYWORDS

Design tools; artificial intelligence; computational design;

**ACM Reference Format:**
Sam H. Ross*, Ecenaz Jen Ozmen**, Maria V. Kogan**, Lydia B. Chilton** . 2018. WordBlender: Principles and Tools for Generating Word Blends. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

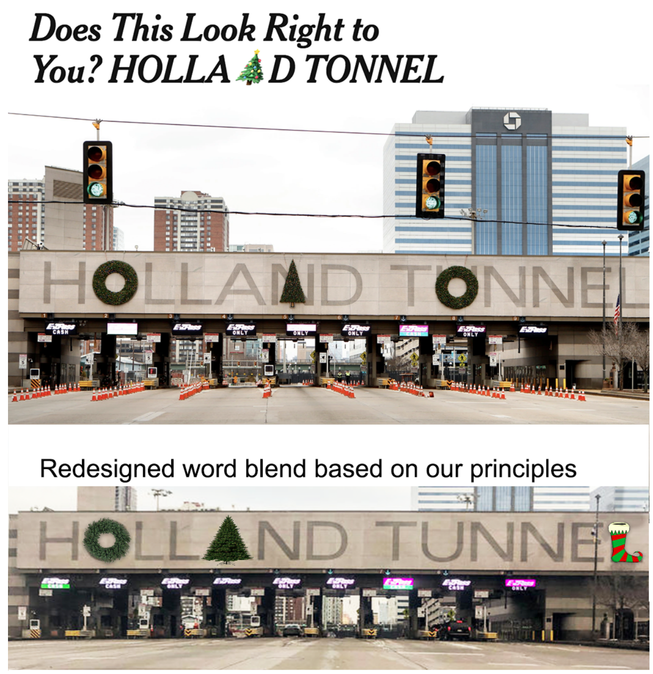


**Figure 1: New York Times article making fun of a bad word blend, and our redesigned word blend based on the principle of shape fit. The 'O' is replaced by a Wreath, the 'A' is replaced by a tree, and the 'L' is replaced by an elf boot.**

## 1 INTRODUCTION

Combining text and images is a powerful strategy in graphic design because images convey meaning faster, but text conveys more precise meaning [35]. There are many computational tools to help users in design tasks mixing text and images: laying out text and image elements [25, 30, 34], matching the mood of the image with the mood of the font [11], and generating graphics to help illustrate elements of a sports story [27]. WordBlends are a different way of combining words and images, not by placing them within a layout, but by blending them together to link concepts in a succinct, yet expressive manner.

Word blends are a common graphic design technique used in logos, posters, advertisements, and announcements to associate a word with a concept, like associating the Holland Tunnel with Christmas. To make a word blend, a designer replaces a letter in a word with a symbol relevant to a message. For example, a Christmas wreath in place of an "O" in "Holland" to make the Holland Tunnel seem more festive during December.

Word blends are hard for the same reason design is often hard: there is a large space of possible combinations, and unclear or contradictory rules to guide you. For Word Blends, the letter and objects should be blended enough that the word is still readable, but not so blended that the object is not visible. Figure 1 is an extreme example where The New York Times made fun of a word blend that has been hung prominently on the Holland Tunnel for years [15]. Their critique is that although replacing an O with a wreath is fine,


*Woodstock '18, June 03–05, 2018, Woodstock, NY*

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

replacing the N with a tree is terribly unaesthetic, and replacing the U with a wreath makes "TUNNEL" look like "TONNEL."

In this paper, we first run experiments to discover principles for making aesthetic word blends, then present a tool to aid users in following these principles. We show that the key property of a word blend is to match letters with objects of the same basic shape. Figure 1 shows this principle applied to fix the Holland Tunnel example. However, the highest rated aesthetic blends also match on secondary visual properties of the object: style, color and thickness. There are no explicit rules for choosing which secondary properties to blend. They are sometimes selected by feature in the image (like color) but sometimes selected based on qualities not in the image such as the mood of the font or connotation of the word. From these experiments, we present an interactive tool that helps users in their design process of finding a good word blend by generating multiple appropriate designs based on shape fit and allows users to make slight adjustments and select their favorite.

## 2 RELATED WORK

### 2.1 Automated Design Tools

Automated design tools have a long history of aiding people both within graphic design and outside graphic design. In graphic design, these tools can automatically make readable route maps [5], step-by-step assembly instructions [4], and diagrams for explaining how things work. A general framework for making automated design tools is to make a design problem a search problem [3], where design principles define a search space, and a set of constraints to guide the search.

Automated tools are also helpful in design tasks outside of graphic design: education [24], medicine [16], interior design [26], games [32], urban planning [7], and accessibility [13]. In many of these a similar approach is taken: search a space of combinations in order to maximize an objective function. This can also be done with more recent deep learning algorithms that do not use design principles, by simply learning from many examples. This has enabled fun tools such as Faceswap [28] and style transfer [18] as well as potential dangerous tools such as deep fakes [33]. Although these tools are fast, not all design problems can be done automatically. Many design problems are not fully specified and currently require human judgement and guidance.

### 2.2 AI Tools to Support the Design Process

Most design work is done through the the iterative design process [29]. It acknowledges that not all the rules are known, and new problems and solution spaces may be discovered during the design process. Such a process is impossible to fully automate, however we can build AI tools to support each step of the iterative design process such as ideation [36, 37], search for existing solutions [12, 14, 20], prototyping [21, 22], refinement [6], synthesis [9, 19, 22] and evaluation [8]. Some tools even attempt to computationally aid every step of the design process [10, 23].

All these tools help users in two fundamental ways: (1) they help navigate the search faster with data mining, specialized search tools, and heuristic evaluation and (2) they reduce the time to create the artifacts by supporting prototyping, synthesis, and refinement. To help people create word blends, we seek to find as many design principles as possible, but if the rules are not clear, we can use interactive AI tools to guide the design process and allow people to make the final adjustments and selection of results.

## 3 DISCOVERING DESIGN PRINCIPLES

In a word blend, the object should be recognized both as itself, and as the letter of the word it replaces. According to cognitive neuroscience [1], the human visual system (HVS) uses four features to recognize an object: its rough shape, color, internal details, and fine-grained silhouette. This suggests that considering the principles of word blends, there are four dimensions to think about: shape, font-style, font-color, and font-thickness. Additionally, the HVS uses a hierarchy of these features: Basic shape is the first and most important features; color, details and silhouette are second. This indicates that the most important feature to match on is shape, and that font-style, color, and thickness should be considered after shape. From this theory we pose and test two hypotheses for design principles for word blends:

*H1. Shape match. Viewers will prefer word blends with the exact match over both close and bad shape matches.* Objects that are a poor match for the letters will negatively impact the aesthetics of the blend, like the tree covering the 'N' on the Holland Tunnel. Objects that are a close (but not exact match) to the letters will negatively impact the readability of the word, like the wreath in the 'U' of the Holland Tunnel. Although the viewer can probably still figure out meaning from context, the appearance and the context disagree and this will confuse and bother viewers.

*H2. Secondary visual features. Viewers will prefer blends that match on 2 of the 3 secondary features, but it does not matter which ones they are.* Three features will be too many to make the object stand out, and zero or one will be too few to look blended. Thus, matching on two secondary features will help the object balance between being individually recognizable and blended with the word.

### 3.1 Experiment 1: Shape Match

We first test whether exact shape match is preferred more than close shape match and bad shape match.

*3.1.1 Method.* To test shape match, we first identified exact, close, and bad letter matches between the letters in the alphabet. For example, the letter 'P' is closer to the letter 'F' than the letter 'Z'. We compared capital letters to each other using pixel-by-pixel matching on images, recording the proportion that the pixels matched between the two letters. Each proportion was the sum of all mismatched pixels out of the total pixels in the overlapped letter images. Letters only had an exact match with themselves. The highest proportion match after that was considered a close match. Anything with a low proportion was a bad match. From this we determined which letters were exact, close and bad matches to other letters.

Then, we found 23 objects that were a very good match for 23 of the 26 capital letters of the alphabet in a sans-serif, medium weight, black font. We had three designers brainstorm and agree on the symbols for each letter. They couldn't agree on good symbols for three letters, 'S', 'R', and 'Z', so we left them out of the test set. For all 23 objects, we created three word blends that placed the object in the spot of an exact, close, and bad match. See Figure 2.

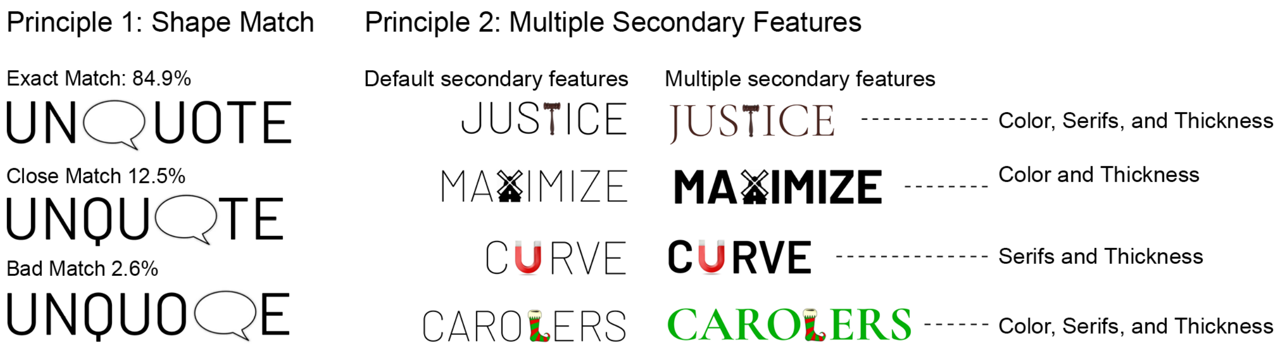


**Figure 2: Principles for blending: shape match and multiple secondary features.**

For each set, we asked them to select their favorite blend. We ran this experiment and the next on the same 15 undergraduates (13 female). The experiment lasted 15 minutes and users were paid $10 for their time.

*3.1.2 Results.* In 84.9% of the cases (293/345) users preferred the exact match. In 12.4% of the cases (43/345) people preferred the close match, and in 2.6% of the cases users preferred the bad match (9/345). Overwhelmingly, users preferred the exact match. To test whether people prefer exact match, we ran a chi-squared test that showed that the number of people preferring the exact match is significantly different than the other cases ($\chi^2$ (1) = 190.0, p<0.0001). This test indicates support for H1 - that users prefer exact match shapes over close matches or bad matches.

Of the close matches that people were divided on, they were choosing between 'O' and 'Q', or 'M' and 'N' or 'C' and 'G'. It is possible that readability is preserved in these cases, making their choice aesthetic, rather than functional. We believe that the small number of people who chose the bad matches are negligible, and possibly due to oversight.

**Principle 1: Objects and letters should have exact shape match.**

## 3.2 Experiment 2: Color, Thickness, and Font Style Match

After matching an object based on shape, the blend will have a minimum level of readability and aesthetic quality. Next, we test whether we can maximize the aesthetic quality of the blend by using two of the three secondary visual features: font style, color, and thickness.

*3.2.1 Method.* To test the effect of secondary feature match on aesthetic quality, we took the 23 word blends with exact shape from the previous experiment and for each one create eight combinations of the three secondary features (serif/sans-serif, default color/matching color, thick/thin). We showed all 8 versions of each word blend to users in random order and asked them to select which blend they thought was the most readable and aesthetic.

*3.2.2 Results.* To test whether users prefer blends that match on exactly two secondary features, we ran a chi-squared test to see if the number of times people pick a blend with two matching features is better than chance. Contrary to H2, and we found that it was not. Users choose chose blends that matched on two features in only 41% of cases (142/345). By by chance they would have chosen these in 37.5% of the cases. According to a chi-squared test these two are not statistically different ($\chi^2$ (1) = 0.863, p<0.35).

In general, people preferred blends with more matches. They selected blends with 0 or 1 matches only 26.3% of the time (91/345). They chose blends with 2 or 3 matches (the maximum) the other 73.6% of the time (254/345). This difference is statistically significant ($\chi^2$ (1) = 40.7, p<0.0001). To understand what was driving these choices, we looked at their explanations for their choices.

Some participants did agree with our hypothesized reasoning: that objects should blend with the text on two secondary principles, but stand out on one secondary principle. The most chosen blend for "CURVE" had matches on two secondary principles: serifs and thickness, but not color (Figure 2). P7 explain her choice by saying *"if it's all red, it's a lot. the black is better"*.

However, for many other words, users overwhelmingly chose blends that had matches on all three secondary principles, such as "CAROLERS." P10 said *"the colors are matching and the serifs match with the curvature of the shoe."* In retrospect, it is obvious why this blend isn't too well matched. The red stripe details on the object naturally make it stand out against the green font. Another feature users mentioned but we did not consider was the connotation of the fonts. P5 chose the bold font for "MAXIMIZE" because *"boldness reminds me of what it means to maximize."* and p4 picked a serif font for "JUSTICE' because *"when I think of law, I think of this font"* (p7).

This is an argument that we should not try to fully automate the design process - there are more secondary and possibly tertiary features of word blending that we might never be able to test. However, we can still support the user in making their choices easier.

**Principle 2: Blending multiple secondary principles is generally better, however, exact number and choice of secondary principles is left to the discretion of the user.**

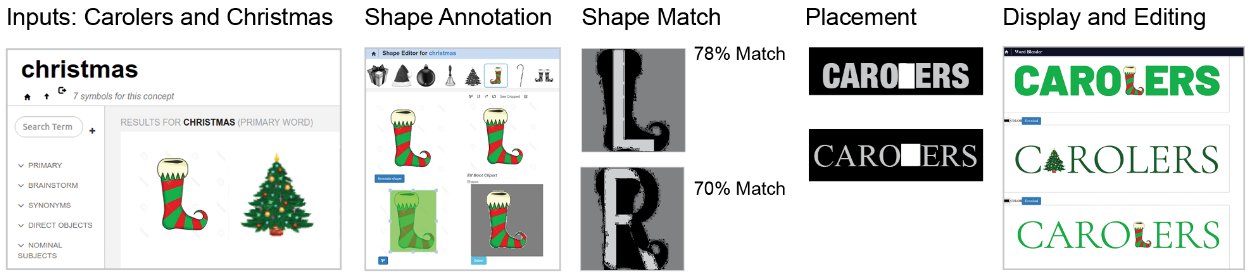


**Figure 3: The stages of the WordBlender system. The user inputs a concept (*Christmas*) and a word (*carolers*). The system automatically extracts the object and classifies the shape, and places the symbols in the corresponding letter in the word. The user sees multiple word blends with different fonts and colors, and can pick their favorite to download.**

## 4 THE WORDBLENDER SYSTEM

From these principles, we learned that shape match is most important, but the secondary visual aspects have more flexibility in their application. Thus, we designed WordBlender to first automatically generate word blends based on shape match, then allow flexibility for users to explore which secondary visual principles get blended. The WordBlender is a Flask-based web application that uses OpenCV, Tensorflow [2], NLTK, and Numpy on the back-end to run computer vision and deep learning algorithms to classify and generate word blends, and HTML5 and Fabric.js on the front end to display word blends and allow users to edit them.

**Inputs** First the user inputs a word and concept to associate it with. For example, they may enter the word *carolers*, and want it associated it with *Christmas*. First, they have to find symbols of Christmas. Our system speeds up this process by using NLTK, WordNet, and ConceptNet to find synonyms and related words for *Christmas*, then the Google Image Search API to display the top 10 results for each of those terms (Figure 3).

**Shape match** Once the set of symbols are found, the system finds matches between letters and symbols. There are two challenges here: first, automatically cropping the main object out of the images, and second, classifying which letter it looks most like. To extract the object, we use a pre-trained model for deeply supervised salient object detection [17] to get a mask of the object. We then use the mask to find contours and determine a bounding rectangle for the object. The image is then cropped to the bounding rectangle identified by finding contours with OpenCV. For most images, this runs quickly (less than 10 seconds) but doesn't always produce an accurate mask so we use interactive GrabCut as a back-up [31] method of extraction.

To classify which letter the extracted object looks like, we run an algorithm that compares the image contours to contours of all the letters of the alphabet in two sans-serif fonts - a thin font (Avenir) and a thick font (Helvetica Bold). We resize the cropped object to letter size without changing the aspect ratio. We overlap our image with the letter image and calculate how many times in each pixel position the images don't match. We divide this value by the total number of pixels to get the percentage mismatch. The user sees an ordered list of the 5 highest-ranked matching letters and can reorder the list if they see any errors. This result in an overall accuracy of 78.9% for matching a image to it's correct letter in the top 5 results.

**Object Placement** After the shape and letter matches are found, the system displays them to the user. First, it needs to find the location of the target letter. To do this, it creates an image of the word and finds the bounding box for the target letter. We place the object in the word by using the bounding box identified for the corresponding letter and the bounding rectangle identified for the object. The scale factor is calculated by dividing the height of the box for the letter by the bounding rectangle of the object.

**Display and edit word blends** To display and edit a word blend on the front-end, we create a Fabric.js canvas that contains two text objects, each corresponding to the portion of the word before and after the letter, with a gap in the middle where the cropped image is placed. All three objects within the canvas can be moved, resized, and rotated. A color can also be selected from the image using and HTML5 color picker, which automatically turns both text objects to that color. This allows users to do fine-grained editing of each blend. In addition to editing each blend, we produce multiple word blends with different combinations of the objects in the database and different fonts in different weights and styles. The user can also change the color of the text by using an HTML5 color picker. Once the user finds a word blend they think is good they can download the blend from the canvas as a png and import it into any application they want to use.

## 5 CONCLUSION AND FUTURE WORK

Even when fully automatic tools fail, AI tools can support stages of the design process to expand the search, or speed up the synthesis of designs. For creating word blends, we have shown how some design principles are reliable enough to be executed fully automatically, but other design principles still require human perceptions and judgement to apply well. In informal trials with novices and experts, we have found that the WordBlender tool greatly speeds up the exploration and execution of designs. Users are often surprised by how much the secondary design principles improve the aesthetics of their design. In future work, we will study the benefits of the tool in terms of time and creativity gains of using a tool.

## REFERENCES


[1] 2017. Cognitive neuroscience of visual object recognition. https://en.wikipedia.org/wiki/Cognitive_neuroscience_of_visual_object_recognition.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[3] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. 2011. Design Principles for Visual Communication. *Commun. ACM* 54, 4 (April 2011), 60–69. https://doi.org/10.1145/1924421.1924439

[4] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. 2003. Designing Effective Step-by-step Assembly Instructions. *ACM Trans. Graph.* 22, 3 (July 2003), 828–837. https://doi.org/10.1145/882262.882352

[5] 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 (SIGGRAPH '01)*. ACM, New York, NY, USA, 241–249. https://doi.org/10.1145/383259.383286

[6] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2011. Effective End-user Interaction with Machine Learning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, 1529–1532. http://dl.acm.org/citation.cfm?id=2900423.2900664

[7] Dino Borri and Domenico Camarda. 2009. The Cooperative Conceptualization of Urban Spaces in AI-assisted Environmental Planning. In *Proceedings of the 6th International Conference on Cooperative Design, Visualization, and Engineering (CDVE'09)*. Springer-Verlag, Berlin, Heidelberg, 197–207. http://dl.acm.org/citation.cfm?id=1812983.1813012

[8] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. 2017. Learning Visual Importance for Graphic Designs and Data Visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 57–69. https://doi.org/10.1145/3126594.3126653

[9] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physical Demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 809–820. https://doi.org/10.1145/2984511.2984559

[10] Lydia B. Chilton, Savvas Petridis, and Maneesh Agrawala. 2019. VisiBlends: A Flexible Workflow for Visual Blends. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 172, 14 pages. https://doi.org/10.1145/3290605.3300402

[11] Saemi Choi, Kiyoharu Aizawa, and Nicu Sebe. 2018. FontMatcher: Font Image Paring for Harmonious Digital Graphic Design. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 37–41. https://doi.org/10.1145/3172944.3173001

[12] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. 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)*. ACM, New York, NY, USA, 845–854. https://doi.org/10.1145/3126594.3126651

[13] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically Generating Personalized User Interfaces with Supple. *Artif. Intell.* 174, 12-13 (Aug. 2010), 910–950. https://doi.org/10.1016/j.artint.2010.05.005

[14] Karni Gilon, Joel Chan, Felicia Y. Ng, Hila Liifshitz-Assaf, Aniket Kittur, and Dafna Shahaf. 2018. Analogy Mining for Specific Design Needs. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 121, 11 pages. https://doi.org/10.1145/3173574.3173695

[15] Michael Gold. 2018. Does This Look Right to You? HOLLAD TONNEL. *The New York Times* (Dec 2018). https://www.nytimes.com/2018/12/13/nyregion/holland-tunnel-decorations-christmas-tree.html

[16] Narayan Hegde, Jason D Hipp, Yun Liu, Michael Emmert-Buck, Emily Reif, Daniel Smilkov, Michael Terry, Carrie J Cai, Mahul B Amin, Craig H Mermel, Phil Q Nelson, Lily H Peng, Greg S Corrado, and Martin C Stumpe. 2019. Similar image search for histopathology: SMILY. *npj Digital Medicine* 2, 1 (2019), 56. https://doi.org/10.1038/s41746-019-0131-z

[17] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. 2017. Deeply Supervised Salient Object Detection with Short Connections. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5300–5309.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*.

[19] Joy Kim, Mira Dontcheva, Wilmot Li, Michael S. Bernstein, and Daniela Steinsapir. 2015. Motif: Supporting Novice Creativity Through Expert Patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1211–1220. https://doi.org/10.1145/2702123.2702507

[20] Ranjitha Kumar, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R. Klemmer, and Jerry O. Talton. 2013. Webzeitgeist: Design Mining the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3083–3092. https://doi.org/10.1145/2470654.2466420

[21] James A. Landay. 1996. SILK: Sketching Interfaces Like Krazy. In *Conference Companion on Human Factors in Computing Systems (CHI '96)*. ACM, New York, NY, USA, 398–399. https://doi.org/10.1145/257089.257396

[22] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-driven Scenes. *ACM Trans. Graph.* 36, 4, Article 130 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073653

[23] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. 2000. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 510–517. https://doi.org/10.1145/332040.332486

[24] J. Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. 2016. Interface Design Optimization As a Multi-Armed Bandit Problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4142–4153. https://doi.org/10.1145/2858036.2858425

[25] Paridhi Maheshwari, Nitish Bansal, Surya Dwivedi, Rohan Kumar, Pranav Manerikar, and Balaji Vasan Srinivasan. 2019. Exemplar Based Experience Transfer. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. ACM, New York, NY, USA, 673–680. https://doi.org/10.1145/3301275.3302300

[26] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive Furniture Layout Using Interior Design Guidelines. In *ACM SIGGRAPH 2011 Papers (SIGGRAPH '11)*. ACM, New York, NY, USA, Article 87, 10 pages. https://doi.org/10.1145/1964921.1964982

[27] Ronald Metoyer, Qiyu Zhi, Bart Janczuk, and Walter Scheirer. 2018. Coupling Story to Visualization: Using Textual Analysis As a Bridge Between Data and Interpretation. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 503–507. https://doi.org/10.1145/3172944.3173007

[28] Yuval Nirkin, Iacopo Masi, Anh Tu an Trãn, Tal Hassner, and Gérard Medioni. 2017. On Face Segmentation, Face Swapping, and Face Perception. *arXiv preprint arXiv:1704.06729* (April 2017).

[29] Donald A. Norman. 2002. *The Design of Everyday Things.* Basic Books, Inc., New York, NY, USA.

[30] 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 (CHI '15)*. ACM, New York, NY, USA, 1221–1224. https://doi.org/10.1145/2702123.2702149

[31] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH '04)*. ACM, New York, NY, USA, 309–314. https://doi.org/10.1145/1186562.1015720

[32] Gillian Smith, Jim Whitehead, and Michael Mateas. 2010. Tanagra: A Mixed-initiative Level Design Tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10)*. ACM, New York, NY, USA, 209–216. https://doi.org/10.1145/1822348.1822376

[33] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. 2015. Real-time Expression Transfer for Facial Reenactment. *ACM Transactions on Graphics (TOG)* 34, 6 (2015).

[34] 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 (IUI '18)*. ACM, New York, NY, USA, 547–558. https://doi.org/10.1145/3172944.3172949

[35] A. Wheeler. 2012. *Designing Brand Identity: An Essential Guide for the Whole Branding Team.* Wiley. https://books.google.com/books?id=Od3VyITkCSUC

[36] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Distributed Analogical Idea Generation: Inventing with Crowds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1245–1254. https://doi.org/10.1145/2556288.2557371

[37] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Searching for Analogical Ideas with Crowds. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1225–1234. https://doi.org/10.1145/2556288.2557378