Staging E-Commerce Products for Online Advertising using Retrieval Assisted Image Generation

Yueh-Ning Ku[1,†], Mikhail Kuznetsov[2,†], Shaunak Mishra[3,*,†] and Paloma de Juan[4]

[1]Snap Inc., USA
[2]Amazon, USA
[3]Amazon, USA
[4]Yahoo Research, USA

Abstract

Online ads showing e-commerce products typically rely on the product images in a catalog sent to the advertising platform by an e-commerce platform. In the broader ads industry such ads are called dynamic product ads (DPA). It is common for DPA catalogs to be in the scale of millions (corresponding to the scale of products which can be bought from the e-commerce platform). However, not all product images in the catalog may be appealing when directly re-purposed as an ad image, and this may lead to lower click-through rates (CTRs). In particular, products just placed against a solid background may not be as enticing and realistic as a product staged in a natural environment. To address such shortcomings of DPA images at scale, we propose a generative adversarial network (GAN) based approach to generate staged backgrounds for un-staged product images. Generating the entire staged background is a challenging task susceptible to hallucinations. To get around this, we introduce a simpler approach called copy-paste staging using retrieval assisted GANs. In copy paste staging, we first retrieve (from the catalog) staged products similar to the un-staged input product, and then copy-paste the background of the retrieved product in the input image. A GAN based in-painting model is used to fill the holes left after this copy-paste operation. We show the efficacy of our copy-paste staging method via offline metrics, and human evaluation. In addition, we show how our staging approach can enable animations of moving products leading to a video ad from a product image.

Keywords

image generation, generative adversarial networks, online advertising, e-commerce

1. Introduction

The choice of image for an online ad can have a significant impact on the online user exposed to the ad. If the ad image is enticing enough, it can not only create brand awareness among online users but also drive them to click the ad and make subsequent purchases (conversions) [1, 2]. However, if the ad image is not properly designed to capture the user's attention, it would lead to poor user interactions and adversely affect the advertising platform (by lowering revenue) and the advertiser (by lowering conversion rate). In this context, a common observation [3] is that ad images with products in a natural or real world setting (lifestyle images) tend to have better online performance. For example, an ad selling a chair is expected to perform better if the image shows a chair in a living room versus a chair against a solid (synthetic) background (as shown in Figure 1). However, such staging of products may be expensive and time consuming, specially when a vendor is selling multiple products at the same time. In DPA offerings from ad platforms (e.g., Yahoo), the catalog images from an e-commerce vendor (e.g., Walmart, Amazon) are typically used directly as ad images. As described later in our data analysis (based on data from an ad platform), a major fraction of such images are not staged, and hence there is a scope to enhance such images (e.g., by generating a suitable background for the product).

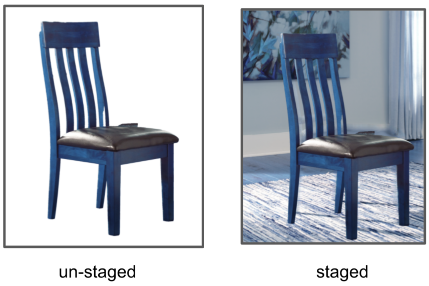


Figure 1: Un-staged and staged version of a chair sold by an e-commerce vendor (staged version is likely to have better CTR).

Image generation has been an actively studied

AdKDD'23, Long Beach, CA

*Corresponding author.

†Work done while at Yahoo Research.

CEUR Workshop Proceedings (CEUR-WS.org)

topic for the past few years. Diffusion models [4] and GANs [5, 6] have been powering the state-of-the-art results in this area. Image in-painting [7] is a slightly easier version of the problem where only parts of the image need to be generated as opposed to the whole image. To the best of our knowledge, we are the first to study GAN based image generation approaches for enhancing product images to serve as ad images. Our main contributions can be summarized as follows.

1. We study three tasks (as outlined below): (i) vanilla staging, (ii) copy-paste staging, and (iii) image-to-parallax animation.
2. In task 1, we aim to generate the entire background for a product. We use pix2pix [6] to train a GAN model with pairs of images (input: segmented out product image, output: staged product image with ground truth background).
3. In task 2, our goal is to retrieve a similar product image (with staging) and copy-paste the background while filling in gaps (holes) created in the process of swapping products. We leverage GAN-based in-painting to fill in the gaps mentioned above. We also introduce a weighted boundary loss for in-painting to focus on the image generation quality at product boundaries. Through Frechet inception distance (FID) score, and human evaluations we show that copy-paste staging is significantly better than the vanilla staging baseline.
4. In task 3, we use GAN-based in-painting to create a sequence of images simulating the main product's movement against the staged background as in a parallax animation. The foreground and background both move, but at different speeds, creating the illusion of depth. This is to show how our approach can lead to video ads from product images.

Our retrieval based approach (second task above) shares the intuition common in text generation: retrieval augmented generation (RAG) has better context understanding and generation quality. The remainder of this paper is organized as follows: related work in Section 2, problem formulation in Section 3, relevant data in Section 4, and our proposed approaches in Section 5. We go over our experimental results in Section 6, and end with a discussion in Section 7.

2. Related work

Online advertising: In online advertising, the ad creative (text and image) plays an important role in influencing online users towards brand awareness, clicks and purchases [2, 8, 9]. Studying ad images and text using state-of-the-art deep learning models in computer vision and natural language processing (NLP) is an emerging area of research. In [10], ad image content was studied using computer vision models, and their dataset had manual annotations for: ad category, reasons to buy products advertised in the ad, and expected user response given the ad. Using this dataset, [11, 12] used ranking models to recommend themes for ad creative design using a brand's Wikipedia page. In [3], object tag recommendations for improving an ad image was studied using data from A/B tests. Although related to ads, the above methods are not applicable in our setup since none of them are image generation methods.

GANs for image generation Generative Adversarial Networks (GANs) are a popular approach for image generation. While vanilla GANs [5] generate images from random noise, Conditional GANs [13] also allow to use extra information in a generative process. Recently developed pix2pix method [6] and its successors [14] use Conditional GANs for image-to-image generation optimizing GAN objective together with distance to a target image. While we focus on GANs in this paper, our copy-paste staging approach can be generalized to more recent diffusion models [4] (discussed in Section 7).

Saliency detection Saliency detection in product images is needed to understand which parts of an image correspond to the main product being advertised (as opposed to the background). Salient object detection (SOD) aims to detect the most visually attractive objects with precise boundaries in images (i.e., it returns a boundary map which can be used to segment out objects from the image). With the introduction of convolutional neural networks (CNNs) in computer vision, SOD accuracy has witnessed remarkable improvements. Recently, U^2-Net [15] achieved state-of-the-art results for saliency detection by using a nested version of U-Net [16]-like architecture to capture richer local and global information. In our proposed approaches, we use saliency detection via U^2-Net as a building block.

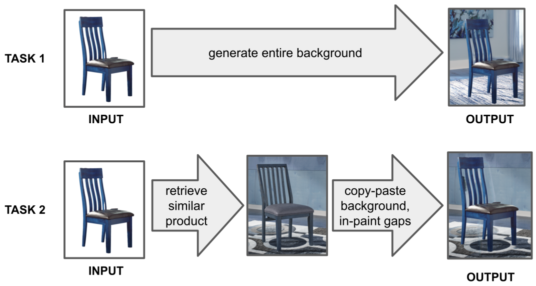

TASK 1
generate entire background
INPUT
OUTPUT
TASK 2
retrieve similar product
copy-paste background, in-paint gaps
INPUT
OUTPUT

Figure 2: Tasks 1 and 2 explained with examples. In task 2, we need to retrieve a related product image with staging and then copy paste its background onto the input image, whereas in task 1 the goal is to generate the entire background.

3. Problem formulation

We study three tasks in this paper as outlined below.

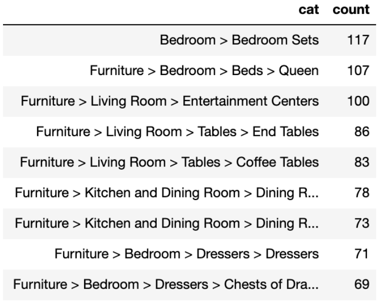
Task 1: vanilla staging. In task 1, the input is a product image (without staging), and the desired output is a product image with a model generated relevant background (stage). The model should generate the entire background as shown in Figure 2.

Task 2: retrieval assisted copy-paste staging. Task 2 is a simpler version of task 1. Here, we are given a pool of existing product images $\mathscr{P}$, and we need to retrieve a similar product image with staging such that we can copy-paste the staged background from the retrieved image onto the input image as shown in Figure 2. Image generation (in-painting) is needed to fill in the gaps after copy-pasting (since the input product and the product in the retrieved image are not identical, gaps will be created when we swap products).

Task 3: image-to-parallax animation. In this task, the goal is to take an input image (as shown in Figure 2 for task 2), and create an animation (i.e., sequence of images), where the object in the input image (as in Figure 2) appears to be moving against a stationary but staged background. Such animations are expected to lead to higher user engagement [17].

4. Data

For our experiments, we sampled data from Yahoo Gemini DPA (spanning November-December 2020). With an impression threshold of 10, 000, we collected a sample of ∼ 18, 899 ads (each corresponding to a product), from which 11, 927 had solid backgrounds, and the rest 6, 972 had staged backgrounds. Each product in the sample belongs to a certain hierarchical category, e.g. ”Kids > Kids Nightstands”, ”Furniture > Bedroom > Headboards > Queen”, ”Shoe > Heel > Oxford Heel”. For our experiments, we considered only furniture images with staged backgrounds, which left us with 2071 images. Top-frequent subcategories with instance counts for the “Furniture” category (as in our sample) are shown in Figure 3.



cat	count
Bedroom > Bedroom Sets	117
Furniture > Bedroom > Beds > Queen	107
Furniture > Living Room > Entertainment Centers	100
Furniture > Living Room > Tables > End Tables	86
Furniture > Living Room > Tables > Coffee Tables	83
Furniture > Kitchen and Dining Room > Dining R...	78
Furniture > Kitchen and Dining Room > Dining R...	73
Furniture > Bedroom > Dressers > Dressers	71
Furniture > Bedroom > Dressers > Chests of Dra...	69

Figure 3: Subcategories for the “Furniture” category and counts.

5. Product staging using GANs

In this section, we first explain the salient object detection model which is common to our proposed approaches for all the tasks outlined in Section 3. Next, we cover our proposed approaches for tasks 1, 2, and 3 (tasks explained in Section 3) in Sections 5.2, 5.3, and 5.4 respectively. Our major modeling contributions are in the approaches for tasks 2 and 3; for task 1, although we define the task, we use existing approaches (pix2pix [6]) to solve it. Task 2 is an easier version of task 1, and our proposed approach leads to more realistic staged product images compared to pix2pix for task 1.

5.1. Product segmentation via saliency

For tasks 1, 2 and 3, we use U^2-Net [15] as our saliency object detector. The saliency object detector plays a crucial role in the first step of our approaches since it separates the main product(s) that will be replaced or copy-pasted versus the background in a product image. Once saliency probability maps are obtained from U^2-Net, we set the threshold at 0.5 to generate binary masks and separate foreground pixels from the background

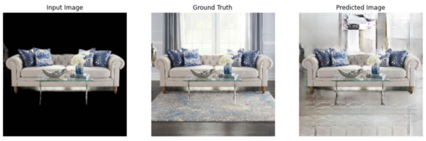


Figure 4: Example of vanilla staging using pix2pix.

pixels. For each task, we use product segmentation in a different manner as explained below.

5.2. Vanilla staging

For vanilla staging, we use the pix2pix method to generate an image background for a product which is segmented by a saliency mask. The algorithm is a conditional GAN optimizing the loss combining 1) regular GAN objective, and 2) ℓ_1 distance between original and restored images. We use product segmentation to prepare pairs of images to train pix2pix for stage (background generation). In particular, given an image with a staged product, we remove the background via product segmentation and use this as the input image to pix2pix. Figure 4 shows an example for this approach (original image in the middle, segmented product on the left, and the version with generated staging on the right).

5.3. Retrieval assisted copy-paste staging

For copy paste staging, we bypass the problem of generating the entire image background by using backgrounds from other relevant images. Our method consists of the following steps:

1. For a given segmented product (Figure 1, left), retrieve top-k similar products from a training collection (Fig. 6a). Similarity measure is a cosine distance between embeddings of corresponding product images provided by Inception-V3 [18].
2. For the top-k similar images, segment out the original products (Fig. 6b) and fill in the holes by inpainting using EdgeConnect [7] (GAN based model) and a new loss function that we introduce in Section 5.3.1.
3. Copy-paste the original product image to the inpainted top-k similar images, aligning shape and center mass for the corresponding product masks (Fig. 6c).

The above algorithm is illustrated with examples in Figure 5. We provide additional examples in Figure 6. After completing steps 1-3, we generate k product images with various backgrounds, only small parts of which (holes around the product before/after) are generated by GANs, which makes the images look more real if comparing with vanilla staging. For better background generation we introduce a new loss function as described below.

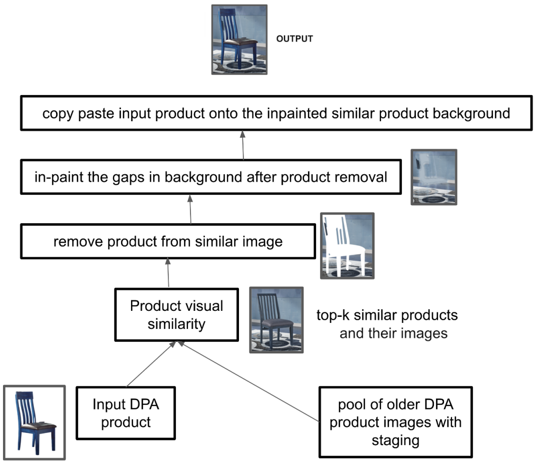
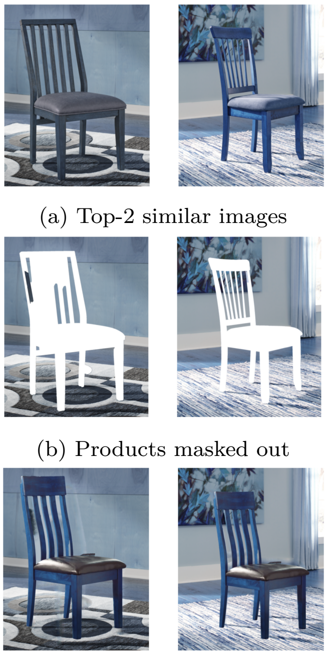

OUTPUT
copy paste input product onto the inpainted similar product background
in-paint the gaps in background after product removal
remove product from similar image
Product visual similarity
top-k similar products and their images
Input DPA product
pool of older DPA product images with staging

Figure 5: Proposed copy-paste staging algorithm.



(a) Top-2 similar images

(b) Products masked out

(c) Copy-paste result after inpainting

Figure 6: End-to-end result of copy-paste staging for top-2 similar images

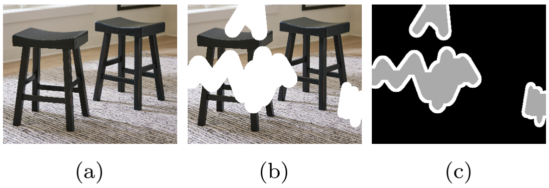

(a) (b) (c)

Figure 7: Illustration of weighted boundary loss. 7a: Ground-truth training data. 7b: Free-form masked training data. 7c: Weighted mask.

5.3.1. Weighted Boundary Loss

Recent works [7] and [19] explore coarse-to-fine inpainting approaches, since the structures of objects are complex and diverse, adding an intermediate step, like edge maps or monochromic images, can help models to learn progressively and eventually generate better final inpainted outputs. We propose a weighted boundary loss (WBL) to not only simplify the learning process (since the model needs to focus on lesser area), but also mimic the end application use case. Following prior work [7], our total generator loss consists of a conventional adversarial loss L_{ADV} and a feature-matching loss L_{FM}. In addition to these two losses, since our goal is to make the model learn better at the boundary of the masked area, we add weighted boundary loss L_{WBL} to amplify the loss penalty at the boundary area pixels. WBL is:

$$L_{WBL} = W_{map} * L_{\ell_1-norm}(E_{GT}, E_{pred}), \qquad (1)$$

where E_{GT} is ground truth edge map of input images, E_{pred} is predicted edge map generated by the generator. The W_{map} is a pixel-wise weighted map and has the same size as input masked images and ground truth. To be more specific, the W_{map} has $\lambda_{boundary}$ for pixels around the boundary between masked area and unmasked area, and $\lambda_{non-boundary}$ for pixels away from the boundary, the pixel-wise $L_{l1-norm}$ will multiply the corresponding λ as we calculate L_{WBL}. As Figure 7 illustrates, for each training sample, we create free-form dense masks by the method proposed by [20]. Then, we find the boundary area of the free-form mask and assign $\lambda_{boundary}$ (white area in Figure 7c) and $\lambda_{non-boundary}$ (gray area in Figure 7c). For experiments, we fixed $\lambda_{boundary} = 0.9$ and $\lambda_{non-boundary} = 0.1$.

5.4. Image-to-animation

Parallax effect happens when the background pixels move slower than foreground objects in an animation, thereby creating an illusion of depth in a two-dimensional image. Generally, parallax effect requires independent foreground images and background images, and proper technique to make transparent backgrounds. In our proposed approach, by leveraging the power of salient object detection and in-painting, a parallax effect animation can be generated from a 2D image. Practically, we run salient object detection to define foreground pixels, then gradually move the foreground object around creating empty gaps between the current position of the object and original position. To fill the empty gap, we then use image in-painting model to in-paint those pixels and create serial realistic images. We illustrate the sample results of the above approach in Figure 8.

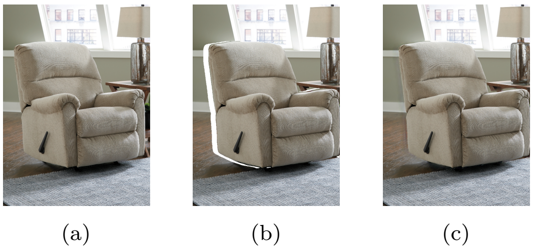

(a) (b) (c)

Figure 8: Illustration of image parallax effect. 8a: original image 8b: image after shifting all the salient pixels. 8c: gap in-painted.

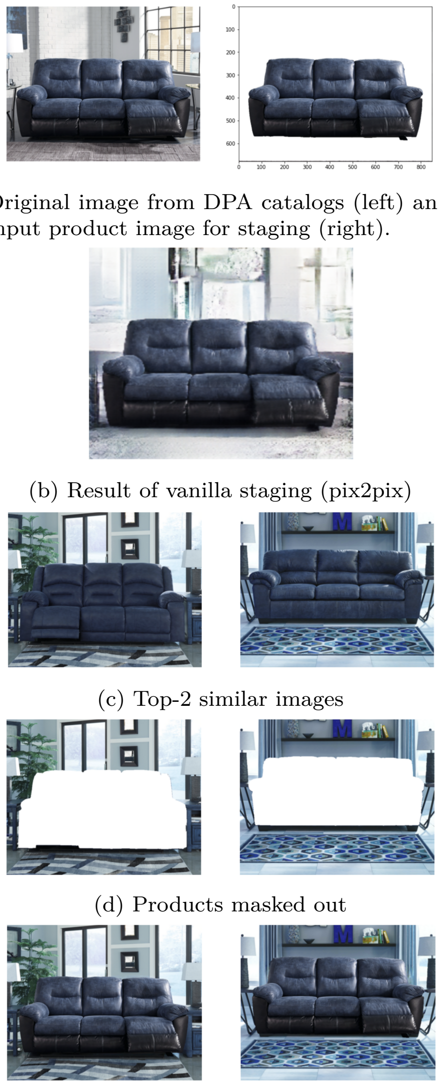
6. Results

We first go over some sample results for tasks 1-3 followed by offline metrics (retrieval performance, generation quality) and human perceptual study results.

6.1. Sample results for tasks 1, 2 and 3

Task 1 (vanilla staging): sample results for task 1 (obtained via pix2pix) are shown in Figures 9 (b) and 10 (b). Compared to the original image with the background, the generated image has a lot of artifacts, and does not look so realistic. As we discuss below, the copy-paste staging results look more realistic.

Task 2 (copy-paste staging): Figures 9 (e) and 10 (e) show sample results for task 2 using the proposed copy-paste staging approach. Overall, the copy-paste staging results look much more realistic compared to pix2pix results.



(a) Original image from DPA catalogs (left) and the input product image for staging (right).

(b) Result of vanilla staging (pix2pix)

(c) Top-2 similar images

(d) Products masked out

(e) Results of Copy-paste staging

Figure 9: Example 1 for copy-paste staging.

Task 3 (parallax animation) A sample for the generated image-to-parallax animation: https://www.dropbox.com/s/9at5gz24ukhf2gi/product_staging_image_to_parallax_demo.mp4?dl=0.

6.2. Offline metrics

Table 1
Precision/Recall results for image retrieval

	@1	@3	@5
precision@k	0.468	0.409	0.374
recall@k	0.468	0.664	0.734

Table 2
FID scores for various staging and training options

	EdgeConnect (baseline)	EdgeConnect + WBL
vanilla staging	127.77	122.22
copy-paste staging	38.44	37.44

Similar product image retrieval performance: a retrieved image is considered similar, if it belongs to the same subcategory as the input product. For example, if the input is a queen bed of subcategory ”Furniture > Bedroom > Headboards > Queen”, the retrieved product should fall into the same subcategory. Table 1 shows precision-recall results for similar products retrieval. For evaluation, we considered furniture subcategories (42 in total) with number of images ≥ 20.

Generation quality: we measure the performance of our copy-paste staging results by evaluating Frechet inception distance (FID) [21]. FID is a popular metric for evaluating the quality of images created by GANs. The Wasserstein-2 distance in FID is calculated by comparing the features distribution of in-painted images with the distribution of real images, where the features are generated by a pre-trained InceptionV3 model. The comparison results are shown in Table 2. Since the copy-paste method in-paints only small regions of image around an object, it achieves much better FID score than vanilla staging. WBL further improves FID score in both methods.

6.3. Human evaluation

For a human perceptual study, we performed pairwise comparison tests, where experts were given two images at once, and the task was to determine which image appears more realistic (natural). For each comparison, we used three independent expert judgements, and decided the winner by majority voting. We performed three such tests: 1) 100 comparisons of vanilla staging images vs. ground truth images; 2) 100 copy-paste staging (with WBL) vs. ground truth; 3) 100 copy-paste (with WBL) vs. vanilla staging. The study showed:

- 0% of pix2pix images were better than ground truth;
- 3% of copy-paste images were better than ground truth;

- 76% of copy-paste images were better than pix2pix.

The above results clearly demonstrate the superiority of copy-paste staging and are in line with offline FID scores.

7. Discussion

Our proposed approach provides low budget advertisers a way to stage products digitally without having to spend on the physical resources needed for staging. Staging a room can easily cost up to few hundred dollars for an advertiser, and with image generation methods like the ones we have proposed, this would be basically free of cost (except for the legalities around copying backgrounds from other images). Leveraging the recent progress in prompt based image generation models, our approach can be further improved along the following lines: backgrounds from similar images could be used to generate prompts which then generate the background of the original product image. In addition, staged ads and parallax animations are expected to drive user engagement, and validating such hypothesis via an A/B test is one of our next steps.

References

[1] N. Bhamidipati, R. Kant, S. Mishra, A large scale prediction engine for app install clicks and conversions, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, 2017.

[2] Y. Zhou, S. Mishra, J. Gligorijevic, T. Bhatia, N. Bhamidipati, Understanding consumer journey using attention based recurrent neural networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, 2019.

[3] S. Mishra, M. Verma, Y. Zhou, K. Thadani, W. Wang, Learning to create better ads: Generation and ranking approaches for ad creative refinement, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, 2020.

[4] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, ICML 2021, ????

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[6] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[7] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, M. Ebrahimi, Edgeconnect: Generative image inpainting with adversarial edge learning, 2019.

[8] S. Mishra, M. Kuznetsov, G. Srivastava, M. Sviridenko, Visualtextrank: Unsupervised graph-based content extraction for automating ad text to image search, KDD '21, 2021.

[9] M. Verma, S. Mishra, Recommendation systems for ad creation: A view from the trenches, RecSys '22, 2022.

[10] Z. Hussain, M. Zhang, X. Zhang, K. Ye, C. Thomas, Z. Agha, N. Ong, A. Kovashka, Automatic understanding of image and video advertisements, in: CVPR, 2017.

[11] S. Mishra, M. Verma, J. Gligorijevic, Guiding creative design in online advertising, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, 2019.

[12] Y. Zhou, S. Mishra, M. Verma, N. Bhamidipati, W. Wang, Recommending themes for ad creative design via visual-linguistic representations, in: Proceedings of The Web Conference 2020, WWW '20, 2020.

[13] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[14] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional gans, in: IEEE CVPR, 2018, pp. 8798–8807.

[15] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, M. Jagersand, U2-net: Going deeper with nested u-structure for salient object detection, Pattern Recognition 106 (2020) 107404.

[16] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

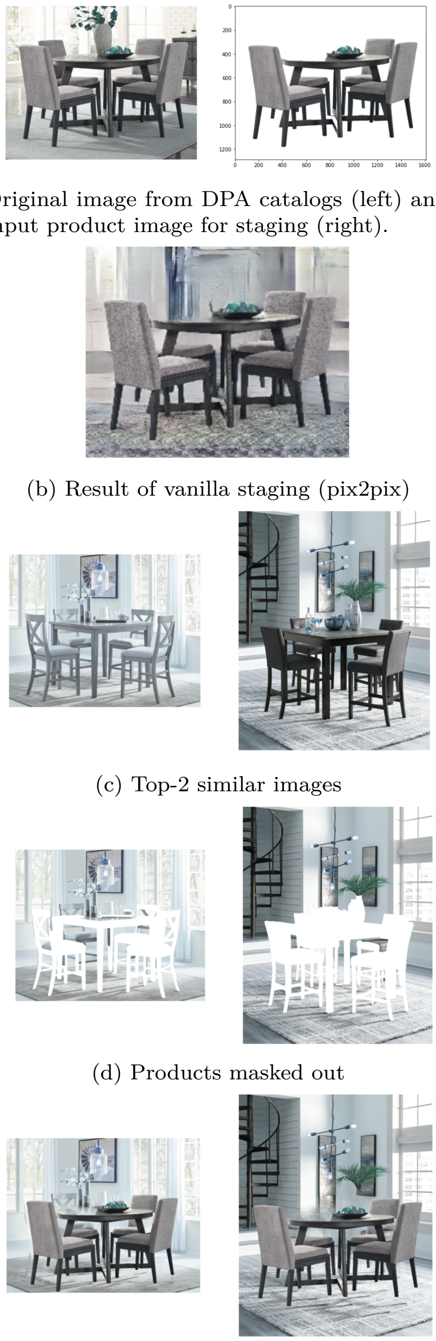
[17] New study by verizon media, magna, & ipg media lab finds interactive ad formats engage hard-to-convince audiences, https://www.verizonmedia.com/press/2021/04/12/new-study-by-verizon-media-magna-ipg-media-lab, ????

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: IEEE CVPR, 2016, pp. 2818–2826.

[19] T. Wang, H. Ouyang, Q. Chen, Image inpainting with external-internal learning and monochromic bottleneck, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5120–5129.

[20] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T. S. Huang, Free-form image inpainting with gated convolution, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4471–4480.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Advances in neural information processing systems 30 (2017).



(a) Original image from DPA catalogs (left) and the input product image for staging (right).

(b) Result of vanilla staging (pix2pix)

(c) Top-2 similar images

(d) Products masked out

(e) Results of Copy-paste staging

Figure 10: Example 2 for copy-paste staging.