

# Iconify: Converting Photographs into Icons

Takuro Karamatsu  
 Kyushu University  
 Fukuoka, Japan

Gibran Benitez-Garcia  
 Keiji Yanai  
 The University of  
 Electro-Communications  
 Tokyo, Japan  
 yanai@cs.uec.ac.jp

Seiichi Uchida  
 Kyushu University  
 Fukuoka, Japan  
 uchida@ait.kyushu-u.ac.jp



Figure 1: Iconified results. Black-and-white icons (top) and color icons (bottom) are generated from photos.

## ABSTRACT

In this paper, we tackle a challenging domain conversion task between photo and icon images. Although icons often originate from real object images (i.e., photographs), severe abstractions and simplifications are applied to generate icon images by professional graphic designers. Moreover, there is no one-to-one correspondence between the two domains, for this reason we cannot use it as the ground-truth for learning a direct conversion function. Since generative adversarial networks (GAN) can undertake the problem of domain conversion without any correspondence, we test CycleGAN and UNIT to generate icons from objects segmented from photo images. Our experiments with several image datasets prove that CycleGAN learns sufficient abstraction and simplification ability to generate icon-like images.

## CCS CONCEPTS

• **Applied computing** → *Media arts*.

## KEYWORDS

icons; image generation; generative adversarial networks

### ACM Reference Format:

Takuro Karamatsu, Gibran Benitez-Garcia, Keiji Yanai, and Seiichi Uchida. 2020. Iconify: Converting Photographs into Icons. In *2020 Joint Workshop*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*MMArt-ACM '20, June 8, 2020, Dublin, Ireland*

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-7137-7/20/06...\$15.00  
<https://doi.org/10.1145/3379173.3393708>

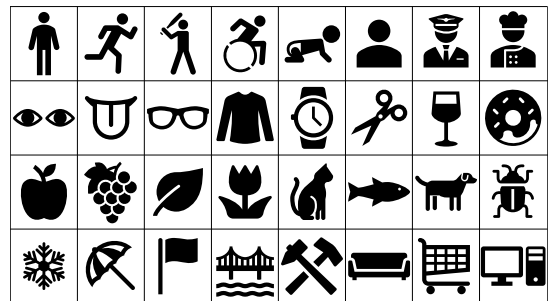


Figure 2: Black-and-white icon images provided in Microsoft PowerPoint.

*on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia (MMArt-ACM'20), June 8, 2020, Dublin, Ireland. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3379173.3393708>*

## 1 INTRODUCTION

Throughout this paper, we assume that icon images, or pictogram, are designed by abstracting and simplifying some object images. Figure 2 shows the black-and-white icon images provided in Microsoft PowerPoint. We can observe that icon images are not just binarized object images but designed with severe abstraction and simplification of the original object appearance. For example, person's heads are often drawn as a plain circle. Graphic designers have professional knowledge and skills of abstraction and simplification while keeping discriminability as the original object.

This paper reports our trials to generate icon images automatically from natural photographs by using machine learning techniques. Our main purpose is to reveal whether the machine learning techniques can capture and mimic the abstraction and simplification skill of human experts on designing icons. We encounter the following three difficulties that make our task challenging.

The first difficulty is that this is a domain conversion task between two sample sets (i.e., domains). If we have a dataset with image pairs of an icon and its original photo image, our image generation task becomes a direct conversion, which can be solved by conventional methods, such as U-net or its versions. However, it is not feasible to have such a dataset in practice. Hence, we only can prepare a set of photo images and a set of icon images, without any one-to-one correspondence between the two domains.

The second difficulty lies in the large style difference between the photo image domain and the icon image domain. For example, the appearance of a person's head is totally different than that represented in icon images, as shown in Figure 2. Thus, the selected machine learning technique must be able to learn a mapping to fill the large gap between both domains.

The third difficulty lies in the large appearance variations in both domains. Although icon images are simple and plain, they still have large variations in their shapes to represent various objects. Object photo images have even more variations in their shape, color, texture, etc. The mapping between the two domains needs to cope with these variations.

We, therefore, employ CycleGAN[1] and UNIT[2] as the machine learning techniques for our task. Both of them can learn the mapping between the two different domains thanks to a cycle-consistency loss, and this mapping can be used as a domain converter. Note that the original papers of CycleGAN and UNIT tackle rather easier domain conversion tasks, such as horse and zebra and winter and summer scenery. On the other hand, for our task, they have to learn the mapping between a photo image set and an icon image set. So that, the learned mapping can convert arbitrary objects from the photo image to its iconified version.

The results of our trials with several image datasets reveal that CycleGAN is able to iconify photo images even with the mentioned difficulties, as shown in Figure 1. This proves that CycleGAN can learn the abstraction and simplification ability. We also reveal that the quality of the generated icons can be improved by limiting both domains to a specific object, such as persons.

## 2 RELATED WORK

### 2.1 Logos and icons

To the best of our knowledge, there is no computer science research for icons generation, which are defined as abstracted and simplified object images. Instead, we can find many research trials about *logo*. In [3], logo is defined as “a symbol, a graphic and visual sign which plays an important role into the communication structure of a company” and classified into three types: Iconic or symbolic logo, text-based logo, and mixed logo. In this sense, logo is a broader target than icon for visual analytics research.

Comparing to traditional logo design researches that often focus how the logo design affects human behavior and impression

through subjective experiments (e.g., [4–7]), recent researches become more objective and data-driven. Those works are supported by different logo image datasets, such as FlickrLogos[8], LOGO-net[9], WebLogo-2M[10], Logo-2K+[11], and LLD[12]. Especially, LLD is comprised of 6 million logo images and sufficient as a dataset for data-hungry machine learning techniques.

### 2.2 Image generation by machine learning

After the proposal of variational autoencoder (VAE), Neural Style Transfer (NST) [13] and generative adversarial networks (GAN), many image generation methods based on machine learning have been proposed. Especially, GAN-based image generation is a big research trend, while being supported by many quality improvement technologies, such as [14–16].

GANs are also extended to deal with image conversion tasks. Pix2pix [17] is a well-known technique for converting an input image from a domain  $X$  to an image in a domain  $Y$ . Pix2pix is trained with a “paired” sample set  $\{(x, y) | x \in X, y \in Y\}$ . For example,  $x$  is a scene image during daytime and  $y$  is a nighttime image at the same location. By training pix2pix with such pairs, a day-night converter can be performed. CycleGAN[1] and UNIT[2] can also realize a domain conversion task but they are more advanced than pix2pix. Just given two sample sets (i.e., two domains) and without any correspondence between them, they can learn a mapping function between both domains.

Those image generation and conversion methods are also used for generating visual designs. For example, the idea of NST is applied to attach decoration to font images [18] and logo skeleton [19]. GAN is applied to font generation [20, 21]. In [22], a conditional GAN is proposed to paint an edge image with a similar color style to a color image. In [12], GANs are used to generate general logo images from random vectors. In [23], reinforcement learning is employed for sketch abstraction.

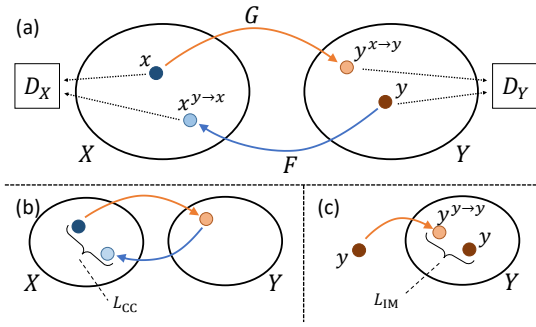
In this paper, we treat an icon generation task as a domain conversion between the photo image domain and the icon image domain. Since there is no prior correspondence between them, we employ CycleGAN [1] and UNIT [2]. We will see that those GANs can bridge the huge gap between the two domains and establish a mapping that “iconify” a photo image to an icon-like image.

## 3 GANS TO ICONIFY

We employ CycleGAN[1] and UNIT[2] to transform natural photos to icon-like images. Both of them are a domain conversion method and can determine a mapping between two domains (i.e., image sets) without giving one-to-one correspondence between the elements of the two sets. In our task, it is not feasible to give one-to-one correspondence between a photo and an icon image in advance to training. Therefore CycleGAN and UNIT are reasonable choices.

### 3.1 CycleGAN

CycleGAN[1] determines a mapping between two image sets,  $X$  and  $Y$ , without giving any image-to-image correspondence. Figure 3 illustrates the overall structure of CycleGAN, which is comprised of two generators (i.e., style transformers)  $G$  and  $F$  and two discriminators  $D_X$  and  $D_Y$ . In other words, two GANs ( $G \leftrightarrow D_Y$  and  $F \leftrightarrow D_X$ ) are coupled to bridge two domains  $X$  and  $Y$ .



**Figure 3: (a) Overview of CycleGAN [1]. Two GANs are coupled to bridge two domains  $X$  and  $Y$ . (b) Cycle-consistency loss,  $L_{CC}$ . (c) Identity mapping loss,  $L_{IM}$ .**

Those modules are co-trained by three loss functions: the adversarial loss  $L_{GAN}$ , the cycle-consistency loss  $L_{CC}$ , and the identity mapping loss  $L_{IM}$ . The adversarial loss is used for training two GANs. The cycle-consistency loss is necessary to realize a bi-directional and one-to-one mapping between  $X$  and  $Y$  by letting  $G^{-1} \sim F$  and vice versa. The identity mapping loss is an optional loss and used for the color constancy on the style transformation by  $F$  and  $G$ .

In the following experiment, we use the network structure and the original implementation<sup>1</sup> provided by the authors [1]. Note that for the experiments to generate black-and-white icons from color photos (Sections 5.1 and 5.2), the color constancy is not necessary. Therefore we weaken the identity mapping loss for those experiments.

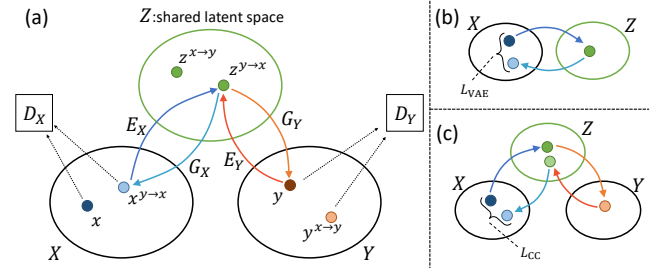
### 3.2 UNIT

UNIT [2] can be considered as an extended version of CycleGAN, which accomplish style transformation between two image sets,  $X$  and  $Y$ . Its main difference from CycleGAN is the condition that an original image and its transformed image should be represented by the same variable in the latent space  $Z$ . As illustrated in Figure 4, UNIT is comprised of two encoders  $E_X$  and  $E_Y$ , two generators  $G_X$  and  $G_Y$  and two discriminators  $D_X$  and  $D_Y$ . Note that the generator  $G$  of CycleGAN is divided into  $E_X$  and  $G_Y$  in UNIT. Those modules are co-trained by VAE loss  $L_{VAE}$ , adversarial loss  $L_{GAN}$ , and cycle-consistency loss  $L_{CC}$ . The VAE loss is introduced so that the latent variable contains sufficient information of original images. In the following experiment, we use the network structure and the original implementation<sup>2</sup> provided by the authors [2].

## 4 IMAGE DATASETS TO ICONIFY

### 4.1 Object photograph data

Since icons have no background in general, we need to prepare object images without background. Unfortunately, there is no large-scale image dataset that satisfies this condition. We, therefore, resort to MS-COCO [24], which is an image dataset with pixel-level ground-truth for semantic segmentation. Figure 5 shows an image



**Figure 4: (a) Overview of UNIT [2]. (b) VAE-loss,  $L_{VAE}$ . (c) Cycle-Consistency loss,  $L_{CC}$ .**

from MS-COCO and its pixel-level ground-truth for three objects, “person”, “dog”, and “skateboard”. Including those three classes, MS-COCO provides ground-truth for 80 object classes.

Figure 6 shows examples of object images extracted by using the pixel-level ground-truth. After removing very small objects, we get 11,041 individual objects from 5,000 images of the MS-COCO. Those images were resized to be  $256 \times 256$  pixels including a white margin. Note that obtained object images often do not include the whole object. Thus, a part of an object is missed in most samples due to the occlusion in the original image. In addition, the object boundary is often neither smooth nor accurate. Therefore, these object images are not perfect as the training samples for icon generation, although they are the best among the available datasets.

### 4.2 Icon image data

As an icon image dataset, we used black-and-white icon images provided by Microsoft PowerPoint. Figure 2 shows examples. Those icons are categorized into 26 classes and the total number of images is 883. Those images are resized to be  $256 \times 256$  pixels including a white margin. As data augmentation during the training of GAN, they are translated, rotated, and scaled to increase their number up to 8,830.

### 4.3 Logo image data as an alternative to icon images

As an alternative to PowerPoint icons, we also examine logo images from LLD [12]. Logos and icons are different in their purpose and shape. For example, texts are often used in logos but not in icons. In addition, we can find more colorful images for logos than icons. However, they are still similar in their abstract design and therefore we also examine logo images. Figure 7 shows logo examples from LLD-logo. The 122,920 logo images in LLD-logo were collected from twitter profile images. In our experiment, we select 20,000 images randomly and resize them to be  $256 \times 256$  pixels (including a white margin) from their original  $400 \times 400$  pixels.

## 5 EXPERIMENTAL RESULTS

### 5.1 Iconify human photos

As the first task, we train both GANs using only icons and photo images depicting persons. Figure 8 shows those training samples. By limiting the shape diversity in the training samples, we can observe the basic ability of GANs to iconify. In advance to training,

<sup>1</sup><https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>

<sup>2</sup><https://github.com/mingyuliu/UNIT>



Figure 5: (a) An image from MS-COCO. Three object class labels, “person”, “dog”, and “skateboard”, are attached to this image. (b) Pixel-level ground-truth for those three classes.

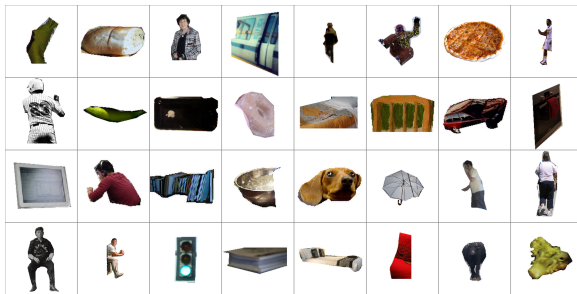


Figure 6: Object photo images extracted from MS-COCO.



Figure 7: Logo images from LLD [12].

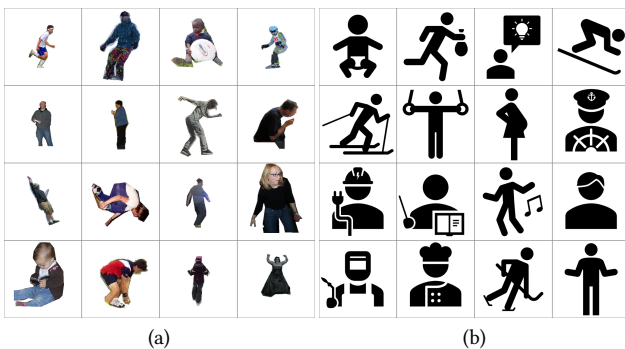


Figure 8: (a) Person photos and (b) person icon images.

we excluded person images which only capture a small part of a human body, such as hand and ear. Icon images showing multiple persons are also excluded. Finally, 1,440 icon images augmented from 72 icon images and 1,684 person photos are used as training samples for CycleGAN or UNIT in this experiment.

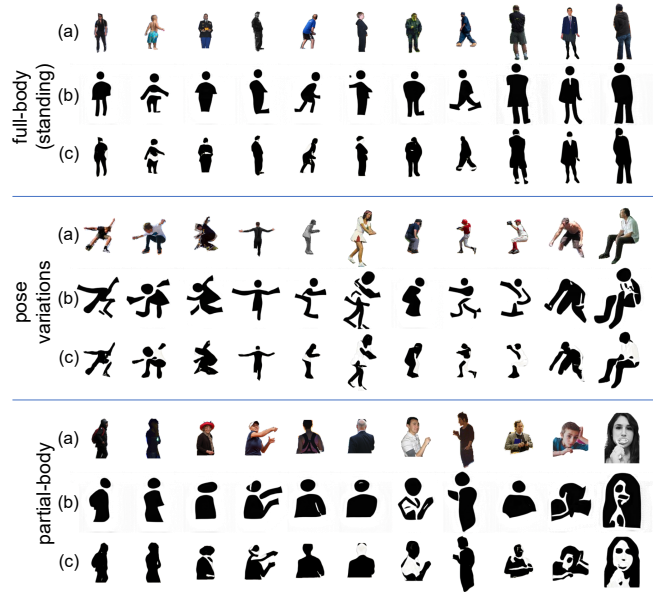


Figure 9: Iconified person photos by GANs trained with icons and photo images depicting persons. (a) Original person photo. (b) Iconified result by CycleGAN. (c) Iconified result by UNIT.

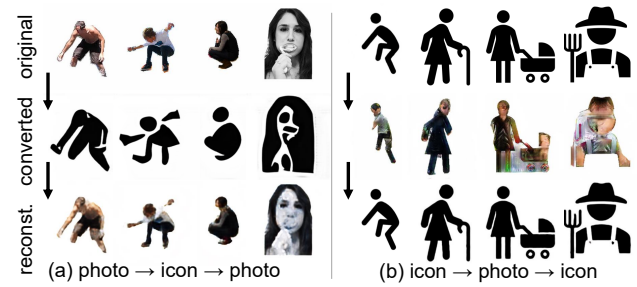


Figure 10: Reconstruction results by CycleGAN in two scenarios, (a) and (b).

Figure 9 shows iconified person photos by CycleGAN and UNIT. These result images are the iconified results of the training samples. Since the number of images is very limited for this “person-only” experiment, it was not realistic to separate the images for training and testing. It should be noted that showing the results of the training samples is still reasonable. This is because, in our task, there is no ground-truth of the iconified result for each photo image; in other words, we do not use any ground-truth information during training. The results in the later sections contain the iconified results of the untrained samples.

From Figure 9 we can see that both GANs successfully convert person photos into icon-like images; they are not just a binarization result but showing strong shape abstraction. Especially, CycleGAN generates more abstract icon images with a circular head and a simplified body shape. It is noteworthy that the head is often separated from the body part and it makes the generated images



Figure 11: Iconified general image photos by CycleGAN trained with PowerPoint icon images. In the orange box, results of untrained samples are shown. In the blue box, typical failure results are shown.



Figure 12: Iconified general image photos by CycleGAN trained with LLD-logo images. In the orange box, results for untrained samples are shown. In the blue box, typical failure results are shown.

more icon-like. For facial images (in the bottom row), their iconified results are not natural. This is because we did not use icon images that show facial details during training.

Comparing to CycleGAN, the results by UNIT are less abstract (i.e., keeping the original shape of person photo) and therefore more similar to the binarization results. Since UNIT has a strong condition that the original photo and its iconified image share the same latent variable, it was difficult to realize strong shape abstraction.

Since CycleGAN has the cycle-consistency loss, it is possible to reconstruct the original photo image from its iconified versions. Figure 10 (a) shows several reconstruction results. It is interesting to note that the original color image is still reconstructed from the black-and-white iconified result. It is also interesting to note that we can convert icon images to photo-like images by using the same CycleGAN model. The examples in Figure 10 (b) show the difficulty of this icon-to-photo scenario. However, the reconstructed icon images are almost the same as the original ones.

## 5.2 Iconify general object photos with PowerPoint icons

As the second task, we use all photos from MS-COCO (Figure 6) and all icon images from PowerPoint (Figure 2) to train CycleGAN and generate the iconified results of general object photos. Since

the first task reveals that CycleGAN has more abstraction ability than UNIT, we only use CycleGAN in this experiment.

This task is far more difficult than the previous; this is because CycleGAN needs to deal with not only the shape variations by the abstraction in icon images but also the shape variations by different object types (e.g., cars and balls). Moreover, the shape variations of object photo images are very severe due to the partial occlusions and non-accurate extractions, as noted in 4.1.

To deal with the huge variations, we used a simple coarse-to-fine strategy for training CycleGAN. Specifically, we first train CycleGAN with the training samples resized to be  $32 \times 32$ . Then, we fine-tune the CycleGAN with  $64 \times 64$ , then  $128 \times 128$ , and finally  $256 \times 256$ . Similar coarse-to-fine strategies are used for other GANs, such as PGGAN[15], SinGAN[16], and DiscoGAN[25].

Figure 11 shows the iconified results. The top row shows the results of the training samples (as noted 5.1, showing the result of training samples is still reasonable since our framework is based on CycleGAN and there is no ground-truth). The results in the orange box of the bottom row show the results of untrained samples (collected from copyright-free image sites). The iconified images show reasonable abstraction from the original photo images and it makes the iconified images different from binarization and edge extraction images.

Although the iconified images are promising to give a hint of icon design, the abstraction is not so strong as Figure 9 of the first task. In addition, the iconified results are different from our “standard” icons. For example, the iconified doughnut and clock images in Figure 11 are different from the standard doughnut and clock icons in Figure 2, respectively. Since there is neither a common rule nor a strong trend in designing the standard icons of various objects, our iconified results show those differences.

The results in the blue box of Figure 11 are typical failure cases. From left to right, the first (orange) and second (keyboard) cases show too much abstraction. Since the original photo images are rather plain, the iconified results also become rough contour images. The third (car) case shows just a fragment of a car and the result cannot represent any car-like shape. The fourth (person) shows blob-like spurious noise, which are caused by insufficient training steps; in fact, in the early steps of CycleGAN training, we often find such failures.

The last failure (hot dog) is an interesting but serious case. Although abstraction has been made appropriately, we cannot identify this iconified result as a hot dog. This case suggests that we need to be careful of the selection of the photo image for making its icon — hot dog has to have its best appearance, shape, posture, and view angle for a legible icon. Non-legible iconified results occur in other objects by this reason.

### 5.3 Iconify general object photos with logos

Figure 12 shows the iconified results by CycleGAN trained with logo images from LLD[12]. The top row shows the results of training samples (i.e., the object images from MS-COCO) and the orange box in the bottom row shows the results of the untrained samples. The photo images are converted like illustrations and therefore we can confirm CycleGAN can generate color icons. In some iconified results, the outline (i.e., edges) of the object is emphasized.

Comparing to the second task, it is also observed that the legibility of the icon images is greatly improved by color. For example, the hot dog icon in the top row shows better legibility than its black-and-white version in Figure 11. Other iconified results also depict their original object more easily than black-and-white versions, even though the colors in the iconified images are not the same as the original object colors.

In the blue box of Figure 12, five typical failure cases are shown: from left to right, no significant change, too much abstraction, text-like icon, text-like spurious noise, and blob-like spurious noise. The first case often occurs when the input photo shows a large object with no background part or a single-color object. The second occurs at fragmentary objects. The third occurs at flat objects; this is maybe due to many logo images from LLD contain a text part.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we experimentally proved that the transformation of natural photos into icon images is possible by using generative adversarial networks (GAN). Especially, CycleGAN [1] has a sufficient “abstraction” ability to generate icon-like images. For example, CycleGAN can generate person icons where each head is represented as a plain circle separated from the body part. From the qualitative evaluations, we can expect that the generated (i.e.,

iconified) images will give hints to design new icons for some object, although the iconified images sometimes show unnecessary artifacts or severe deformations.

As future work, it is better to conduct a subjective or objective evaluation of quality of the iconified images. Finding a larger icon dataset is also necessary to improve the quality. A more interesting task is the analysis of the trained GANs for understanding how the abstraction has been made; this will deepen our understanding about the strategy of professional graphic designers.

## ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP17H06100.

## REFERENCES

- [1] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *Proc. ICCV*, 2017.
- [2] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” *Proc. NIPS*, 2017.
- [3] G. Adir, V. Adir, and N. E. Pascu, “Logo design and the corporate identity,” *Procedia-Social and Behavioral Sciences*, vol. 51, pp. 650–654, 2012.
- [4] L. E. Hem, and N. M. Iversen, “How to develop a destination brand logo: A qualitative and quantitative approach,” *Scandinavian J. Hospitality and Tourism*, vol. 4, no. 2, pp. 83–106, 2004.
- [5] R. Van der Lans, J. A. Cote, C. A. Cole, S. M. Leong, A. Smidts, P. W. Henderson, C. Bluemelhuber, P. A. Bottomley, J. R. Doyle, A. Fedorikhin, J. Moorthy, B. Ramaseshan, and B. H. Schmitt, “Cross-national logo evaluation analysis: An individual-level approach,” *Marketing science*, vol. 28, no. 5, pp. 968–985, 2009.
- [6] L. Cian, A. Krishna, and R. S. Elder, “This logo moves me: Dynamic imagery from static images,” *Journal of Marketing Research*, vol. 51, no. 2, pp. 184–197, 2014.
- [7] B. van Grinsven, and E. Das, “I love you just the way you are: When large degrees of logo change hurt information processing and brand evaluation,” *Advances in Advertising Research*, vol. 6, pp. 379–393, 2016.
- [8] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol, “Scalable logo recognition in real-world images,” *Proc. ICMR*, 2011.
- [9] S. C. H. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu, “Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks,” *arXiv*, 2015.
- [10] H. Su, S. Gong, and X. Zhu, “Weblogo-2m: Scalable logo detection by deep learning from the web,” *Proc. ICCV Workshops*, 2017.
- [11] J. Wang, W. Min, S. Hou, S. Ma, Y. Zheng, H. Wang, and S. Jiang, “Logo-2K+: A large-scale logo dataset for scalable logo classification,” *arXiv*, 2019.
- [12] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool, “Logo synthesis and manipulation with clustered generative adversarial networks,” *Proc. CVPR*, 2018.
- [13] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” *Proc. CVPR*, 2016.
- [14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of Wasserstein GANs,” *Proc. NIPS*, 2017.
- [15] T. Karras, T. Aila, S. Laine, J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” *Proc. ICLR*, 2018.
- [16] T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” *Proc. ICCV*, 2019.
- [17] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proc. CVPR*, 2017.
- [18] S. Yang, J. Liu, Z. Lian, and Z. Guo, “Awesome typography: Statistics-based text effects transfer,” *Proc. CVPR*, 2017.
- [19] G. Atarsaikhan, B. K. Iwana and S. Uchida, “Contained neural style transfer for decorated logo generation,” *Proc. DAS*, 2018.
- [20] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, “Multi-content GAN for few-shot font style transfer,” *Proc. CVPR*, 2018.
- [21] H. Hayashi, K. Abe, and S. Uchida, “GlyphGAN: Style-consistent font generation based on generative adversarial networks,” *Knowledge-Based Systems*, vol. 186, 2019.
- [22] T. H. Sun, C. H. Lai, S. K. Wong, and Y. S. Wang, “Adversarial colorization of icons based on structure and color conditions,” *Proc. ACM-MM*, 2019.
- [23] U. R. Muhammad, Y. Yang, Y. Z. Song, T. Xiang, and T. M. Hospedales, “Learning deep sketch abstraction,” *Proc. CVPR*, 2018.
- [24] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” *Proc. ECCV*, 2014.
- [25] T. Kim, M. Cha, H. Kim, J. K. Lee, J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” *Proc. ICML*, 2017.