Project Report: Icon Style Adaptation

Kailu Wu, Runlong Zhou
Yaoclass 80, IIIS

January 10, 2020

Abstract

Icon adaptation for Android is a style transfer problem where the goal is to transfer an icon to a given style, while in this style only a limited other icons are known. So far there's no public method for adapting icons automatically. Our project tries to use a Star-GAN based method to solve this problem. During the experiments, we derive some findings about Star-GAN and the importance of discriminators. The code of our model can be seen on GitHub.

1. Introduction

Icon design among Android apps has never been unified. To solve this issue, many third-party theme providers offer Android users with many themepacks (packages of predesigned, same style icons). But due to the large volume of Android apps, it is impossible for theme designers to manually design icons for each app, and hardly any themepack can automatically generate icons for some not-so-popular apps. So most themes do not look as nice as their introduction pictures do.

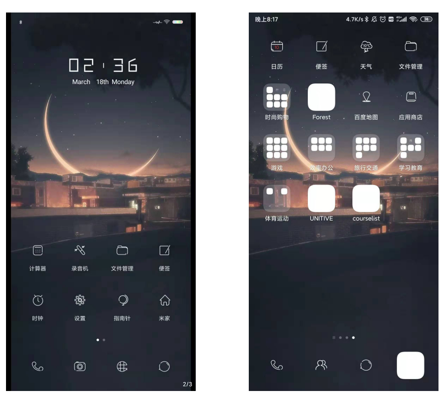

Figure 1: Comparison between the introduction picture and reality

Motivated by this phenomenon, we made an attempt to use GAN to solve this issue.

2. Related Works

2.1. Multi-Content GAN

In (1), the authors implemented an MC-GAN to adapt font styles. Given a few (≤ 7) upper-case letters of the same style, MC-GAN generates all the other letters of this style.

MC-GAN consists of two parts. The "few shot" inputs are fed to the first part, GlyphNet, so that the skeleton of the given font can be learned. The second part, OrnaNet, takes the skeletons as inputs and fine-tunes the color as well as the texture.

2.2. Super-Resolution DenseNet

As will be explained in Experiments part, inputting and outputting large icons take much more time and do not necessarily work well. Thus, we trained on small icons (64×64 pixels) and used super resolution networks to upscale the outputs (to 256×256 pixels). Authors of (7) introduced an SR network based on DenseNet (3), and they innovatively added densely skip connections between any two dense blocks to improve the performance.

2.3. Star-GAN

Star-GAN from (2) is a network that tries to solve style transfer problems on multi domains. The whole network can be regarded as a Conditional GAN (6) with cycle consistency (8).

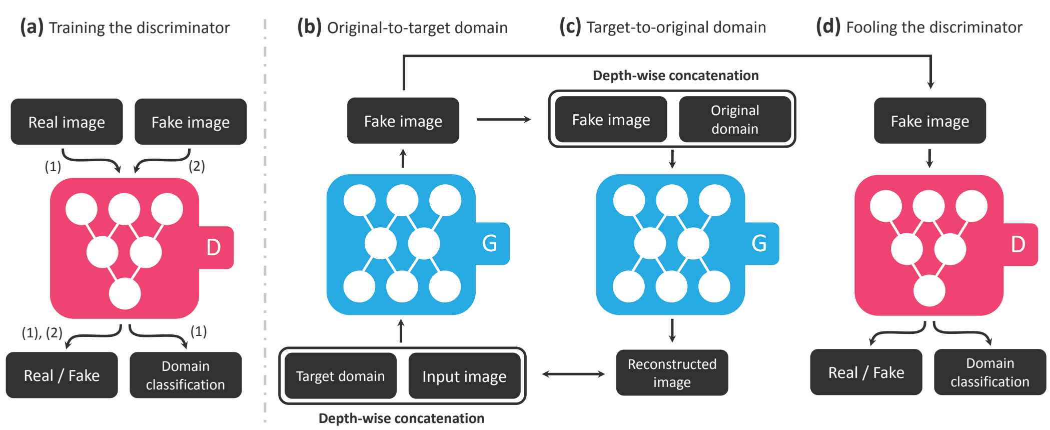

Figure 2: The structure of Star-GAN, from (2)

3. Experiments

3.1. Icon Dataset

We downloaded 111 themes from `zhuti.xiaomi.com` and extracted their icons as dataset. The dataset contain 36420 icons and 111 class, each icon has one label for the app's name. Only less than 20 icons show up in more than 50% of the themes, most icons only show up few times. Which means using a supervised method on this dataset may easily get over-fitted. So we treat this dataset as a unlabeled dataset and use unsupervised learning method.

3.2. Experiment on MC-GAN

We trained the first part (GlyphNet) of MC-GAN on our icon dataset for 15000 epochs (around 24 hours), and the result is listed as follows.

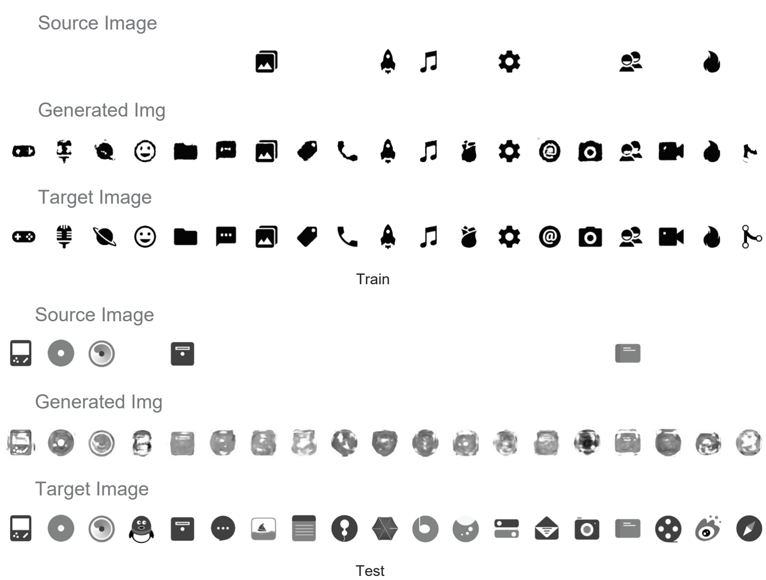

Figure 3: Result of GlyphNet on our icon dataset

Since GlyphNet does not simply output the given icon, we can see that it preserves given information well. But for the missing icons, GlyphNet can only guess QQ's icon to a barely satisfactory extent. Although the GlyphNet works well on training, either it takes much longer time to learn the feature of other icons, or it is hard for GlyphNet to generalize any better.

The possible explanations for this issue are: First, discriminators in MC-GAN only discriminate between real and fake, and do not specify which applications the icons come from. Second, no other losses (such as L_{cyc} in cycle consistence from (9)) were added to constrain the output. Third, the skeleton of icons is much more complex than the skeleton of uppercase letters.

3.3. Experiments on Star-GAN

Star-GAN's original experiments(2) dealt with 8 classes of images, while our dataset includes 111 classes of different styles. So we focused on how to adapt the original model to a large number of classes. After removing some inappropriate data argumentation used in human faces, we tried the following changes. See figure 4 for part of generated images. We'll compare the generated images by what the intuitive feeling is, whether there is artifacts and whether important symbols are kept.

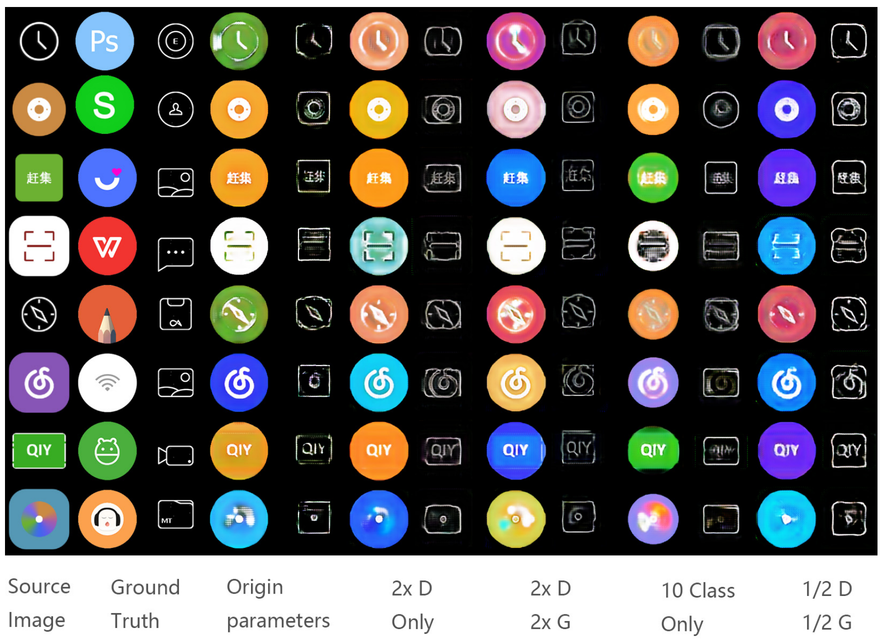

Figure 4: Part of results from different experiments. Ground Truth is some arbitrary images pick from that class because there's no corresponding icon in that class.

Experiment 0. We kept the original parameters of Star-GAN to generate icons of shape 128×128.

Experiment 1. Considering of large number of classes, we enlarged the width of both Generator and Discriminator by 2 times. The result turned out to be not as good as the original one. From the training loss we found that Discriminator could not discriminate real or fake well, compared to the original model.

Experiment 2. Another worry of large number of classes led to a hypothesis that whether decreasing the number of classes yields better performance. Thus we picked only 10 classes from our icon dataset as a new dataset. The result shows that with fewer data the model will over-fit to the dataset because both Discriminator and Generator tend to memorize the data.

Experiment 3. Since there's only 36 thousand images while the model has over 100 million parameters, we tried to reduce the size of both Generator and Discriminator at the same time. The result shows that this is not as good as the original one, especially on icons that needs more details like "Gan Ji". Which means 36 thousand icons are enough for 100 million parameters to learn well.

Experiment 4. Based on above, we tried to enlarge the discriminator by 2 times and change the rate of alternative training for Star-GAN. From the result we can find that the generated icons have clearer details, smoother colors and less artifacts than the results of above experiments.

3.4. Experiment on SR DenseNet

Android system requires the icons to have high resolution, thus we need to train directly on icons with resolution of 256×256 (or even higher) pixels. But enlarging the input and output sizes leads to a much longer training time, insatiable GPU memory requirements and substantial increase in model sizes. Thus, training directly on large inputs and outputs does not necessarily work better than on small inputs and outputs.

(a) Ideal case

(b) Actual case

Figure 5: Comparison between SR DenseNet (left) and resize (right)

SR DenseNet alleviates this pain. Frist, SR DenseNet works better than simply resize (bicubic) the outputs to higher resolutions. Our pretrained SR DenseNet produces better anti-aliasing and sharper boundaries than simple resize. Second, SR DenseNet requires less time than training directly on 256×256 pixels inputs, as the training of SR DenseNet is independent of the icon dataset and requires much less time than a GAN.

However, the outputs of the actual cases do not show that apparent advantages of SR DenseNet over resize, since the outputs of GAN is not that precise.

4. Conclusion

We offered an application of GAN on transfering icon styles: a modified version of Star-GAN and a pretrained SR DenseNet were combined to produce high resolution icons of several given styles. During our experiments we have found that, StarGAN can learn features from large amount of different domains ($111 \gg 8$ as proposed in (2)), and a stronger (larger or taking more information) discriminator is needed for larger amount of different domains. When designing networks, we need to take the physical constraints into account, such as time, memory and size of datasets.

References

[1] Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer, 2017.

[2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, 2017.

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[5] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution, 2017.

[6] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[7] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4809–4817, Oct 2017.

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.