You Get What You Sow: High Fidelity Image Synthesis with a Single Pretrained Network

Kefeng Zhu[1,2*] , **Peilin Tong**[1,2] , **Hongwei Kan**[1,2,3] , **Rengang Li**[1,2]
[1]Inspur Electronic Information Industry Co.,Ltd.
[2]State Key Laboratory of High-End Server & Storage Technology, China
[3]Artificial Intelligence Research Institute, China University of Mining and Technology
{zhukefeng, tongpeilin, kanhongwei, lirg}@inspur.com

Abstract

State-of-the-art image synthesis methods are mostly based on generative adversarial networks and require large dataset and extensive training. Although the model-inversion-oriented branch of methods eliminate the training requirement, the quality of the resulting image tends to be limited due to the lack of sufficient natural and class-specific information. In this paper, we introduce a novel strategy for high fidelity image synthesis with a single pretrained classification network. The strategy includes a class-conditional natural regularization design and a corresponding metadata collecting procedure for different scenarios. We show that our method can synthesize high quality natural images that closely follow the features of one or more given seed images. Moreover, our method achieves surprisingly decent results in the task of sketch-based image synthesis without training. Finally, our method further improves the performance in terms of accuracy and efficiency in the data-free knowledge distillation task.

1 Introduction

Synthesizing realistic images has been a long standing challenge in computer vision. Most recently, the great potential of the generative adversarial network (GAN) [Goodfellow *et al.*, 2014] has made it possible to generate high-quality image in various applications. It allows for generating hyper-realistic natural images [Brock *et al.*, 2018], making fake content more realistic and imperceptible to human, and letting non-artists create realistic images based on hand-drawn sketches [Chen and Hays, 2018; Isola *et al.*, 2017]. In addition, image synthesis techniques have also been applied in data-free model compression for edge device deployment [Lopes *et al.*, 2017; Chen *et al.*, 2019] using knowledge distillation method [Hinton *et al.*, 2015], as well as in data augmentation especially in medical imaging [Frid-Adar *et al.*, 2018].

However, two major constraints have restricted most state-of-the-art methods for image synthesis. On the one hand, a training dataset with millions [Deng *et al.*, 2009] or even more images are usually required, while many datasets facing hurdles to release due to privacy or security concerns. On the other hand, the training process requires large-scale computing power alongside all-round experience and knowledge from experts.

At the same time, as it is well known that a pretrained classification model, e.g. ResNet50 [He *et al.*, 2016], contains rich information of the training set, there have been a branch of methods proposed with the idea of synthesizing images by inverting a classifier. Nonetheless, most of such methods are model-compression-oriented and thus the generated images are more like “fooling samples” that are unrecognizable to human. With the aim of image synthesizing, Santurkar *et al.* [2019] developed a toolbox using single robust classifier and achieved better results in several tasks, but the quality of the synthesized images is still far from real. Yin *et al.* [2020] further improved the inversion method with the help of some metadata stored in the batch normalization (BN) layers of the pretrained model and made the synthesized image quality reach a higher level. But there is still room for improvement on the fidelity of the generated images, as it tends to cause background-inconsistent problems due to the lack of class-specific information.

In this work, we propose a novel strategy for high quality image synthesis by using a single pretrained classification network with class-conditional metadata from particular target images that we call the *seeds*. To be specific, based on the idea of model inversion, we introduce an extra class-conditional natural regularization design in general form to improve the fidelity of the generated images. Then we propose a corresponding metadata-collecting procedure applicable to different practical scenarios. The proposed strategy brings several benefits: 1. By introducing more class-specific information, it further improves the fidelity of the synthesis images (shown in Figure 1) to be comparable to that of the GAN-based methods which require training process. 2. It enables the application in some particular scenarios, e.g. people want to synthesize similar images from specific *seed* samples, due to copyright concerns. 3. The regularization term in general form adds more freedom to design and optimization. Then we show the qualitative and quantitative analysis of the generated images using the proposed method. To further demonstrate the usefulness and effectiveness of our method,

[*] Contact Author

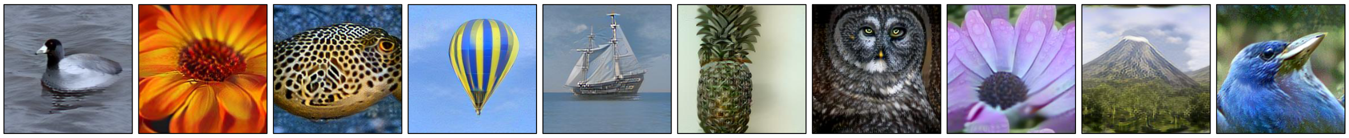

Figure 1: High fidelity synthesis samples from various classes using the proposed method. Best viewed in color.

we show the results in some other applications. Firstly, we show that we can synthesize natural images from hand-drawn sketches and the results closely follow the key features of the sketches. Then, we show the performance improvement in terms of student accuracy and efficiency in the data-free knowledge distillation task.

Our main contributions are summarized as follows:

- We demonstrate the importance of the class-conditional information in the image synthesis process. We introduce a novel strategy for high fidelity image synthesis, including a class-conditional natural regularization design (Sec. 3.2) and its corresponding metadata collecting procedure for different scenarios (Sec. 3.3).
- We improve the fidelity of the synthesis image to the level comparable to that of the state-of-the-art GAN-based methods which require training processes, especially in the *get-from-seed* scenario (Sec. 4.1).
- We demonstrate the surprising performance of the proposed method in the sketch-based image synthesis (SBIS) task. (Sec. 4.2).
- We improve upon prior work on data-free knowledge distillation and achieve better convergence while using fewer synthesized images (Sec. 4.3).

2 Related Works

2.1 Image Synthesis

Most state-of-the-art works in image synthesis are rooted in generative adversarial network (GAN) framework [Goodfellow *et al.*, 2014; Brock *et al.*, 2018]. Though the quality of the generated images is impressive, all these GAN methods require training with the access to the target dataset. Another line of work attempts to tackle the image synthesis task by inverting a pretrained network. Mahendran and Vedaldi [2015] introduced a method to inverse CNN representations with the idea of optimizing an objective function with gradient descent. Then a popular application DeepDream has been proposed for generating artistic effects on input images or drawing ”dreamed” images from random noise. Although helpful for further understanding the neural nets, the resulting images are far from realistic due to the lack of statistics of the original training set. A recent method DeepInversion [Yin *et al.*, 2020] further crafts a natural regularization term by exploiting the average statistics which stored in the BN layers of the pretrained model. The quality of the synthesized images has been greatly improved, but still with limitations due to the lack of important class-specific information.

Sketch-Based Image Synthesis (SBIS). Synthesizing an object or a scene based on hand-drawn sketch has always been a hot branch. Early sketch-based image synthesis methods are mainly based on image retrieval to composite images from a given sketch. With the rising of deep neural networks, GAN-based methods has become the mainstream for this task. Isola *et al.* [2017] proposed a general-purpose solution to this kind of image-to-image translation problems. SketchyGAN [Chen and Hays, 2018] introduced a novel network block that is suitable for both generator and discriminator. SketchyCOCO [Gao *et al.*, 2020] proposed an attribute vector bridged GAN to generate images from object-level sketches without using freehand sketches as the training data. All the GAN-based approaches shown above perform well in the task but have the same shortcoming that they require specific dataset and a training process.

2.2 Knowledge Distillation (KD)

There has been a long line of work and development on transferring knowledge from one model to another. Under the name of knowledge distillation (KD), Hinton *et al.* [2015] has defined the idea of training a compact *student* network to imitate the actions of a *teacher* network. Recently, various methods have been proposed to improve KD [Xu *et al.*, 2018; Park *et al.*, 2019; Romero *et al.*, 2015]. However, all these methods cannot be effectively implemented without the original training dataset. Therefore, a few studies have been done focusing on data-free knowledge distillation. Lopes *et al.* [2017] leveraged auxiliary metadata with the original dataset to reconstruct images. Chen *et al.* [2019] implemented GAN-based method by regarding the pretrained network as a fixed discriminator. Such methods show good performance in the KD task, whereas the generated images are not recognizable to human due to the lack of natural information.

3 Method

3.1 Background

The fundamental problem is to reconstruct input images from a given representation. The inverting process [Mahendran and Vedaldi, 2015] is aimed to find the optimal $x \in \mathbb{R}^{H\times W\times C}$ that minimizes the objective function:

$$x^* = \underset{x\in\mathbb{R}^{H\times W\times C}}{\arg\min}\, \mathcal{L}(\Phi(x), \Phi_0) + \mathcal{R}_{\text{prior}}(x), \qquad (1)$$

where $\mathcal{L}(\cdot)$ is the classification loss (cross entropy) that matches the image representation $\Phi(x)$ to the target Φ_0. $\mathcal{R}_{\text{prior}}(x)$ is a natural image prior regularization term:

$$\mathcal{R}_{\text{prior}}(x) = \alpha_{\text{TV}}\mathcal{R}_{\text{TV}}(x) + \alpha_{l_2}\mathcal{R}_{l_2}(x), \qquad (2)$$

where the *total variance* $\mathcal{R}_{\mathrm{TV}}(x)$ encourages the piece-wise consistency of images, the l_2 *norm* $\mathcal{R}_{l_2}(x)$ contains the reconstructed images to a natural range, and α_{TV}, α_{l_2} are the scaling factors respectively. The DeepInversion [Yin *et al.*, 2020] method further improves the fidelity of the synthesized images by extending the regularization with an additional term that leverages the statistics stored in the BN layers of the given model:

$$\mathcal{R}_{\mathrm{DI}}(x) = \sum_l ||\mathrm{Mean}(x_l) - \mu_l||_2 + \sum_l ||\mathrm{Var}(x_l) - \sigma_l^2||_2, \tag{3}$$

where $\mathrm{Mean}(x_l)$ and $\mathrm{Var}(x_l)$ are batch-wise mean and variance of the l-th convolutional layer, and μ_l and σ_l^2 are the corrsponding *running means* and *running variances* stored in the BN layers of the pretrained network. The $||\cdot||_2$ operator denotes the l_2 norm calculation.

The method cleverly uses the metadata which stored in the pretrained model, however, such delicate design is difficult to implement in some actual scenarios, e.g., an inference-optimized model merges the BN layer with its preceding convolution layer and thus the statistics required can be hardly extracted. Moreover, the method tends to cause a background-inconsistent problem in the synthesized images since the statistics used are global and not encoded with adequate class-specific information.

3.2 Class-Conditional Regularization

In order to introduce more class-specific information and improve the fidelity of the synthesis, we propose a class-conditional regularization term in a general form:

$$\mathcal{R}_{\mathrm{CC}}(x|y) = \sum_l \lambda_l F_l(x_l, s(x_l)|y), \tag{4}$$

where y refers to the target class, a k-dimensional one-hot vector, of the dataset on which the classifier $\Phi(x)$ is trained. $s(x_l)$ is the corresponding class-conditional statistics as a function of each feature map x_l involved in the design. F_l is the similarity function for each layer in a general form and λ_l is its corresponding weight.

With a scaling factor α_{CC}, the proposed regularization term forces the synthesized image to follow the class-specific feature distribution instead of a global average. Thus, the whole regularization term is a class-dependent function:

$$\mathcal{R}(x|y) = \mathcal{R}_{\mathrm{prior}}(x) + \alpha_{\mathrm{CC}}\mathcal{R}_{\mathrm{CC}}(x|y), \tag{5}$$

$$x_y^* = \underset{x \in \mathbb{R}^{H\times W\times C}}{\arg\min}\ \mathcal{L}(\Phi(x), \Phi_0 = y) + \mathcal{R}(x|y). \tag{6}$$

Furthermore, we can freely design the F_l for better optimization. In this work we use a specific design similar to the DeepInversion method for better comparison:

$$F_l = ||\mathrm{Mean}(x_l|y) - \hat{\mu}_{l,y}||_2 + ||\mathrm{Var}(x_l|y) - \hat{\sigma}_{l,y}^2||_2, \tag{7}$$

where the $s(x_l)$ is expanded to class-conditional batch-wise mean and varaiance estimates $\hat{\mu}_{l,y}$ and $\hat{\sigma}_{l,y}^2$ and we use equal weights for F_l in all layers.

As we will show, with the aid of more class-specific information, the proposed method effectively improves the fidelity of the synthesized images.

3.3 Metadata Collecting Procedure

Now that we are able to freely design the class-conditional regularization term, we need the corresponding class-specific metadata for the algorithm to work. The whole metadata-fetching process can be done either *during* or *after* the training process of the given model.

In the former case, the metadata can be collected during training with the original dataset following the regularization design and can be released with the pretrained model as valuable extra information. In this way, it makes the image synthesizing process strictly data-free, and we call it Scenario A.

In the latter case, it fits closer to a specific scenario that we call *get-from-seed*, possibly due to copyright concern, that people want to synthesize images particularly similar to the given one or a set of *seed* images. In this mode the metadata should be collected from such seed(s) which can be obtained arbitrarily, e.g., through the internet, as long as their inference results match the target class y. If there is a set of seed images we call it Scenario B; if there is only one seed image we call it Scenario C. Accordingly, we further propose a class-conditional metadata collecting procedure which is straightforward and applicable in all scenarios described above. The first step is to determine the data source. In Scenario A the source is the original training dataset, while in Scenario B and C the source is the retrieve "seed" images. The second step is to extract metadata $s(x_l|y)$ throughout the inference process. The proposed regularization design is referenced to determine the metadata structure and to pinpoint the locations for the extraction. If there are multiple inputs, we simply take the average. Finally the collected metadata will be passed to the image synthesis update cycle as parameters to regularize the loss function. The whole procedure for Scenario B and C is depicted in Figure 2.

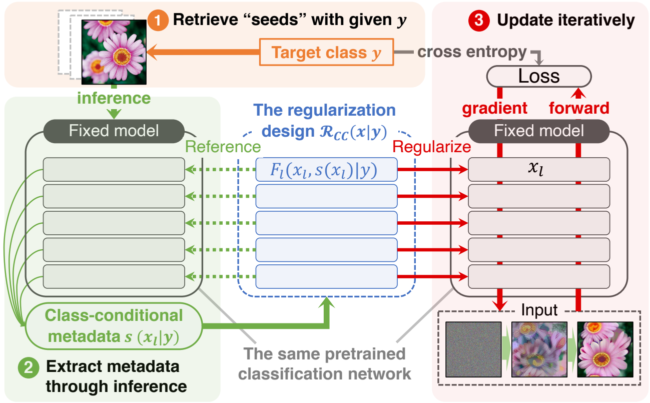
1 Retrieve "seeds" with given y
3 Update iteratively
Target class y
cross entropy
Loss
inference
gradient
forward
The regularization design $\mathcal{R}_{CC}(x|y)$
Fixed model
Reference
Regularize
Fixed model
$F_l(x_l, s(x_l)|y)$
x_l
Input
Class-conditional metadata $s(x_l|y)$
The same pretrained classification network
2 Extract metadata through inference

Figure 2: The metadata collecting procedure designed specifically for the proposed class-conditional regularization, the figure shows the steps specifically for the *get-from-seed* scenarios.

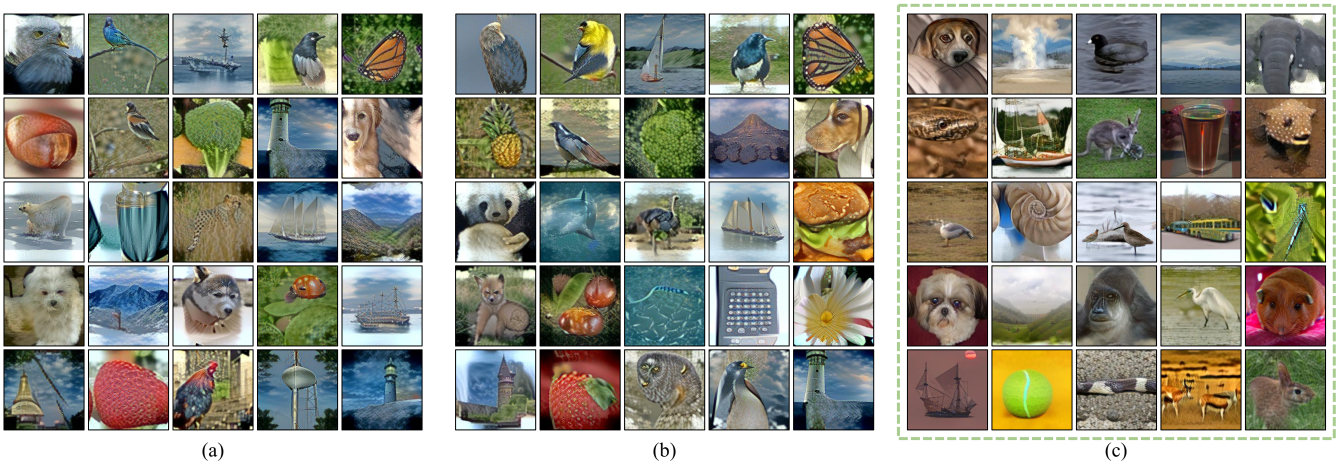

Figure 3: The synthesized images using the proposed method in different scenarios described in Sec. 3.3. Scenario A (a): original pretrained ResNet50 with metadata from the original training dataset; Scenario B (b): BN-fused ResNet50 with metadata from 50 seed images for each class; Scenario C (c): BN-fused ResNet50 with metadata from only one seed image.

4 Experiments

4.1 Evaluation on ImageNet

We perform experiments with pretrained networks based on the large-scale ImageNet dataset [Deng *et al.*, 2009] from 1000 classes. In order to thoroughly demonstrate the performance and effectiveness of the proposed strategy, we study in all three different scenarios described in Sec. 3.3.

Implementation Details

For the network, we use the publicly available pretrained ResNet50 from PyTorch as the base network. The Top-1 accuracy of the network is 76.13%. In Scenario A we just use this network directly. In Scenario B and C, we use an inference-optimized version of the network by applying the static quantization technique which fuses the BN layers into their preceding layers respectively.

For the metadata, in Scenario A we use the original ImageNet training set as the source. In Scenario B we use 50 seed images (retrieved online, inference-checked) for each class to extract metadata, and in Scenario C we use only one such seed image. All the metadata collecting processes follow the procedures described in Sec. 3.3.

For other parameters in all experiments, we use an Adam optimizer with a learning rate of 0.25 and set $\alpha_{\mathrm{TV}} = 1.0 \times e^{-4}$, $\alpha_{l_2} = 1.0 \times e^{-5}$, $\alpha_{\mathrm{CC}} = 0.01$. We run the image synthesis with a batch size of 200 using NVIDIA V100 GPU with automatic-mixed precision (AMP) [Wang *et al.*, 2019] for acceleration. For each batch, we apply 20k iterations to acquire results with good quality. In addition, we synthesize another set of images by using DeepInversion method following the settings described in [Yin *et al.*, 2020] for comparison.

Analysis of the Synthesized Images

Figure 3 shows the resulting images in different scenarios. First, we find the generated images are with high fidelity in Scenario A (Figure 3 (a)). This shows the effectiveness and improvement in image quality of our method in such a simulated data-free scenario, as the metadata is considered to be released along with the pretrained network (similar to the statistics stored in BN layers that used in the DeepInversion method). Additionally, we find that the generated images in Scenario B (Figure 3 (b)) achieve the same quality as that in Scenario A. This demonstrates that for any neural network we

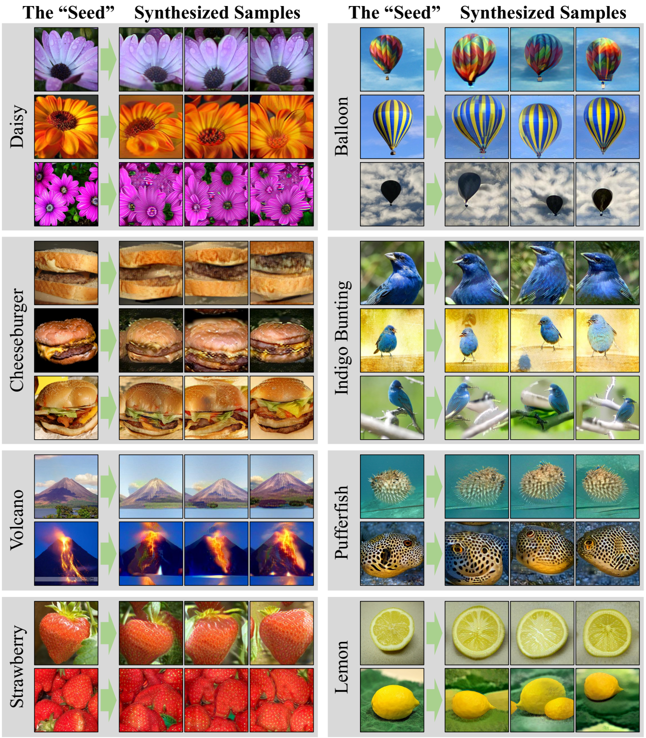

Figure 4: More results in the one-seed mode. Each gray block contains the synthesized samples (the right three columns) from different seed images (the leftmost column) of the same class. Note that the synthesized images closely follow the features of their corresponding seed, but with good diversity.

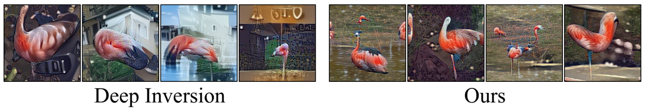

Figure 5: Comparison between the results from DeepInversion (left) and our method (right). The images are generated with a batch size of 200 with the same target class. The background problem can be mitigated (but not eliminated) by diversifying the target classes in a batch.

are able to synthesize high quality natural images by applying the proposed method. Finally, we find the results in the "one seed" mode, Scenario C (Figure 3 (c)), are with significantly better image quality.

The One-Seed Mode. In order to further analyze the results in the one-seed mode (Scenario C), we show more results *we get from what we sow* in Figure 4 and a detailed comparison between different seeds and their corresponding synthesized samples. It can be clearly observed that the synthesized images closely follow the characteristics of their seed, but with good diversity. When we sow different seeds (even from the same class), we get different results accordingly. For example, the synthesized daisies follow the features of their seeds in color, texture and posture, and even inherit the details such as the dew on the petals.

The Background Problem. When applying the DeepInversion method, the synthesized images tend to have unreasonable elements in the background, shown in Figure 5 (left). The most likely explanation is that the natural regularization term in DeepInversion is global and class-independent, which means that the non-decisive background cannot be properly reflected due to the lack of class-specific information. The problem can be mitigated (but not eliminated) by diversifying the target classes in a batch, specifically, by using non-repeat random target classes in a batch. But still we can observe the problem (in their original work) that the synthesized dog, eagle and bug have similar types of inconsistent green background. Our method has essentially solved this problem with the class-conditional design. As we can see in Figure 5 (right), or even in Figure 3 and 4, all the synthesized images using our method are with reasonable backgrounds (e.g. the water habitat that flamingos should live in) and with overall visual consistency for each sample.

Inception Score (IS). The Inception Score (IS) [Salimans *et al.*, 2016] is a popular metric for evaluating the quality of the synthesized images. This metric is designed to correlate well with human scoring of the realism of synthesized images (the higher the better). The IS uses a pretrained Inception-V3 network and calculates a statistic of the network's outputs when applied to images. Table 1 shows the IS of the images synthesized with the proposed method. We find that our method improves over DeepInversion in terms of IS by a large margin. Of course we still have to pay attention to the fact that the generated images with our method (or DeepInversion) are essentially adversarial examples and thus are more likely to produce higher confident predictions for IS.

Method	**Resolution**	**GAN**	**Inception Score**
Real Image	299		229.49
BigGAN [2018]	256	✓	202.6
Ours	**224**		**107.13**
DeepInversion [2020]	224		60.6
WGAN-GP [2017]	128	✓	11.6
DeepDream	224		6.2

Table 1: Inception Score (IS) for synthesized images using various methods on ImageNet. Note that we use Scenario A for our method to generate samples.

Models	**DeepDream Acc. (%)**	**DeepInversion Acc. (%)**	**Ours Acc. (%)**
ResNet50	100.0	100.0	**100.0**
ResNet18	28.0	94.4	**98.9**
MobileNet-V2	13.9	90.9	**95.7**
VGG11	6.7	80.1	**92.3**
Inception-V3	27.6	92.7	**93.3**
ResNet152	–	–	**99.4**
Wide_Resnet50_2	–	–	**99.1**
SqueezeNet1_0	–	–	**93.4**
VGG19_BN	–	–	**97.5**

Table 2: Classification accuracy of the ResNet50 synthesized images on a series of other models. All models are trained on ImageNet. The results on DeepDream and DeepInversion methods are from [Yin *et al.*, 2020]. For our method, Scenario A is adopted for image synthesis.

Generalization Ability. The optimization and improvement of our method has also brought better performance on generalization. It can be found in Table 2 that our method surpasses the DeepInversion method in the generalization test on a series of models.

4.2 Sketch-Based Image Synthesis (SBIS)

As we have achieved satisfactory results in the above-mentioned *get-from-seed* scenarios, another idea comes to our minds: as the synthesized images closely follow the seed images, why not try exploring whether the synthesized images can follow some specific input (can also be considered as "seed" from another angle) such as a hand-drawn sketch? In this section we study the performance of our method on the task of sketch-based image synthesis (SBIS).

Implementation Details

For the sketch data, we use the Sketchy Database [Sangkloy *et al.*, 2017] as a source. The database spans 125 categories with a total of more than 75,000 sketches of 12,500 objects. It's also characterized by providing original photos paired with the sketches. We align the categories in the Sketchy database with the ImageNet dataset and select 120 classes with a maximum of 200 sketches per class as the input for the SBIS task. All the sketches are with the size of 256 × 256. For the pretrained model, we use the BN-fused version of the PyTorch official pretrained ResNet50 network. We use a 200 max batch size for each class. Other settings in this experiment are the same as that in Scenario B, Sec. 4.1.

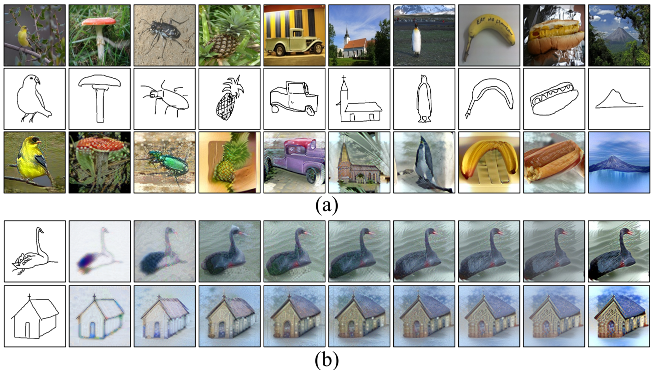

Figure 6: Sketch-based image synthesis samples: (a) shows the results from different sketches. In each column the top image is the original image that the sketch is based on, the middle is the sketch, and the bottom is the synthesized image using our method. Both original and sketch images are from the Sketchy database. (b) shows the intermediate step results in the synthesis process. The leftmost is the input sketch and the rightmost is the final result.

Results and Analysis

A set of images synthesized from sketches with the proposed method are shown in Figure 6. Remarkably, we find that our method is able to achieve satisfactory performance in this task as well, and some detailed analysis are shown as follows:

Naturalness. The synthesized images looks realistic and natural. Moreover, the details in color, texture and background can be naturally synthesized that follow the original features of the corresponding class. For example, the goldfinch is with yellow body, black head and wings, and it stands on a tree branch; the agaric is red and it lives in the grass; and the black swan is with the correct color and it swims in the water, etc.

Faithfulness. The synthesized images closely follow the input sketch in terms of key features such as shape and pose. It can be clearly found in some examples: the posture of the bird (especially the head and the beak), the structure of the church and the direction in which the hotdog is placed, etc. Furthermore, in Figure 6 (b), steps in the synthesis process explains why the results are highly consistent with the input sketch.

To the best of our knowledge, most start-of-the-art methods for this task are GAN-based which require training processes. The results in this section demonstrates that the proposed method also performs well in this task, only with a pretrained classification model. The proposed method may provide a direction worth exploring in this sketch-based image synthesis (SBIS) task.

4.3 Data-Free Knowledge Distillation

Knowledge distillation (KD) task refers to the process of transferring information of a pretrained "teacher" model to a randomly initialized "student" model through training with the help of the original training dataset based on which the teacher was trained. To verify the effectiveness of our method in this task, in this section we use our method to generate data as the training set and then analyze the performance in the KD task.

Data Amount	Original	DAFL [2019] Top-1 Acc. (%)	DeepInversion [2020] Top-1 Acc. (%)	+Ours Top-1 Acc. (%)
10K	85.44	–	55.94	**67.02**
50K	95.34	–	86.55	**89.77**
200K	–	–	91.56	**92.93**
Best	95.34	92.22*	91.56	**92.93**

Table 3: Knowledge distillation student performance using different amount of synthesized/original images on CIFAR-10. The teacher/student networks are ResNet34/ResNet18 respectively. The "Original" refers to the results using the original training set (50K images). *:The DAFL student accuracy is from [Chen *et al.*, 2019] and its corresponding teacher accuracy is 95.58%.

Implementation Details

We first train a ResNet34 network from scratch on CIFAR-10 dataset with 95.67% accuracy as the teacher. For all the image synthesis jobs in this task, we use an Adam optimizer with a learning rate of 0.05. We generate images in batches of 200 with 2000 iterations per batch, with optimized parameters $\alpha_{\text{TV}} = 2.5 \cdot 10^{-5}$, $\alpha_{\text{CC}} = 1.0$, ($\alpha_f = 1.0$ in DeepInversion). As for the metadata used in our method, we simply use the original training set as the source.

For the KD process, first we randomly initialize a ResNet18 network as the student. Then, to evaluate and compare the efficiency in the KD task, we apply our method and DeepInversion separately to generate 10k, 50k, and 200k datasets as the substitutes of the original training set. For all cases, we apply a standard KD training for 200 epochs, with temperature *T*=5, an SGD optimizer with a learning rate of 0.1 and a batch size of 128.

Analysis

From Table 3, we find that, without any limitation on the data amount, the best KD student accuracy with our method reaches 92.93% (with 200K synthesized images), surpassing both DeepInversion [2020] and DAFL [2019]. Moreover, by comparing the performance on the 10K, 50K and 200K datasets, we find that our method achieves better KD accuracy with fewer generated images. Finally, with our method, we can design any types of class-conditional natural regularization term without any restriction (like the BN requirement in DeepInversion) and thus a wider range of networks can be supported.

5 Conclusions

In this paper, we have proposed a novel strategy for natural image synthesis based on a single pretrained classifier with class-conditional information, which is either provided along with the pretrained model release or directly extracted from some given "seed" images. Experiments have shown that the proposed strategy, especially in the *get-from-seed* scenario, is able to improve the fidelity of the synthesized images to a level comparable to the state-of-the-art GAN-based methods. Moreover, the performance in tasks of SBIS and KD have further demonstrated the applicability and effectiveness of the proposed strategy.

References

[Brock *et al.*, 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[Chen and Hays, 2018] Wengling Chen and James Hays. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, Salt Lake City, UT, June 2018. IEEE.

[Chen *et al.*, 2019] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-Free Learning of Student Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages pp. 3514–3522, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009. IEEE.

[Frid-Adar *et al.*, 2018] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using GAN for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

[Gao *et al.*, 2020] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchy-COCO: Image Generation From Freehand Scene Sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5174–5183, 2020.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30:5767–5777, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, Honolulu, HI, July 2017. IEEE.

[Lopes *et al.*, 2017] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-Free Knowledge Distillation for Deep Neural Networks. *arXiv preprint arXiv:1710.07535*, November 2017.

[Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, Boston, MA, USA, June 2015. IEEE.

[Park *et al.*, 2019] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational Knowledge Distillation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3962–3971, Long Beach, CA, USA, June 2019. IEEE.

[Romero *et al.*, 2015] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. *arXiv preprint arXiv:1412.6550*, March 2015.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[Sangkloy *et al.*, 2017] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling Deep Image Synthesis with Sketch and Color. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6836–6845, Honolulu, HI, July 2017. IEEE.

[Santurkar *et al.*, 2019] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image Synthesis with a Single (Robust) Classifier. In *Advances in Neural Information Processing Systems*, volume 32, pages 1262–1273. Curran Associates, Inc., 2019.

[Wang *et al.*, 2019] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-Aware Automated Quantization With Mixed Precision. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8604–8612, Long Beach, CA, USA, June 2019. IEEE.

[Xu *et al.*, 2018] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks. *arXiv preprint arXiv:1709.00513*, April 2018.

[Yin *et al.*, 2020] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to Distill: Data-Free Knowledge Transfer via DeepInversion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8712–8721, Seattle, WA, USA, June 2020. IEEE.