Generating Images from Small Datasets Using Adaptive Point-wise Grouped Convolutions

Mana Takeda[1] and Keiji Yanai[1]

Department of Informatics, The University of Electro-Communications, Tokyo, Japan
takeda-m@mm.inf.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. The recent development of Generative Adversarial Networks (GANs) has made it possible to generate images with high quality. However, the problem is that it usually requires a large amount of training data and a long training time. On the other hand, few-shot GANs have been studied to fine-tune an image generation model trained on large-scale data with a small number of training data in a short time. In this work, we propose a new model for few-shot GANs using adaptive point-wise grouped convolution layers. The experiments have shown that our method can generate images with higher quality than the conventional methods.

Keywords: few-shot image generation · group convolution · VAE · GAN.

1 Introduction

Deep learning models typically use a large amount of data for training. However, the construction of large datasets requires a lot of effort. For training of models with small datasets, using pre-trained models is effective. In image classification models, transfer learning has been widely studied, where a model is trained using a large labeled dataset such as ImageNet, and then transferred to other domains using a small dataset. The method of learning of the target data set from the weights of the pre-trained model is called fine-tuning. Even fine-tuning of a model with other domain datasets than the pre-trained dataset tends to improve the accuracy. This is because the pre-trained model has acquired generally useful weights that cannot be obtained with a small target data set.

Regarding deep generative models, methods to transfer prior knowledge has been proposed. Noguchi and Harada [12] proposed a new method to generate images from a small data set by transferring a pre-trained generative model. To adapt the prior knowledge, they focused on the scale and shift parameters of the generative batch statistics. By updating the scale and shift parameters that adjusts the weights of the filters of the convolutional layers, the pre-trained model can be adapted to the target domain of a small data set.

Similar to Noguchi and Harada, we propose a method to generate images from a small dataset by transferring a pre-trained generative model. Unlike them, we add adaptive point-wise grouped convolutional layers to the generator, and adapt the pre-trained model to the target domain by learning a cross-channel combination of features in the hidden layer of the generator. The experiments on small datasets have show that the proposed method can produce higher quality images than the conventional methods, and it can flexibly interpolate between images.

2 Related works

2.1 Generative Adversarial Network

In the field of image generation, GAN (Generative Adversarial Network) [2] has achieved great success. The structure of a GAN consists of two parts: a generator and a discriminator. The role of the generator is to generate as realistic an image as possible, and the role of the discriminator is to identify whether a given image is a realistic image or not. In the training process, these two networks are trained to compete with each other. The generator network transforms an input noise vector into a real image to fool the discriminator network as much as possible, while the discriminator network is trained to be as undeceived as possible.

2.2 Few-shot GANs

In general, GANs require a large number of training samples to produce high-quality images. Even few-shot GANs require large-scale image datasets such as ImageNet for pre-training, while smaller datasets are used for fine-tuning. Several methods have been proposed for GANs with a few-shot. Conditional GANs are trained on a large set of images, and then a new class of images is trained using a small number of samples. There are two types of GANs that can be trained in a few-shot training. (1) The pre-trained and few-shot domains are the same, and only 3-5 images are used to train a few-shot model to add a new category for the same domain. (2) The pre-trained and few-shot domains are different, and about 50 images are used to train a few-shot model for a new domain. In this study, we focus on the case of (2).

The representative works of (1) include Hong et al. [4, 3] and FS-GAN [14]. Hong et al. [4, 3] proposed a method to fuse information from multiple conditional images of the same class. FS-GAN [14] factorized the weights of the convolutional and fully connected layers of the pre-trained model with SVD to identify a meaningful parameter space for adaptation.

The representative works of (2) include Yijun et al. [9], Noguchi and Harada [12]. Yijun et al. [9] extended the idea of Elastic Weight Consolidation [7] to adapt the source model to the target domain. Extending the idea of Consolidation to adapt the source model to the target domain by penalizing large changes in important weights (estimated via Fisher information) in the source model.

Noguchi and Harada [12] proposed a method to adapt a pre-trained generative model to datasets from different domains. To effectively use the pre-learned knowledge, the weights of the convolutional layers of the generator are all fixed during fine-tuning. Instead, only the scale and shift parameters of the batch normalization (BN) layer are adapted to small datasets in different domains from the training data used in the pre-training. Using the class-conditional batch normalization used in SNGAN projection [10], it was possible to generate a variety of images from a small number of training samples by dynamically changing the scale and shift parameters. Fine-tuning of the BN parameters is considered as channel-wise feature modulation [13].

Like Noguchi and Harada, our work focuses on improving the sample efficiency of GANs by adapting a few-shot. However, unlike them, we do not only tune the parameters of each channel but also tune the channels by linear combination among them. In other words, while they employ “channel-wise” feature modulation, we employ “cross-channel” feature modulation. To enable cross-channel feature modulation, We propose to introduce adaptive point-wise grouped convolutional layers into the generator during fine-tuning. We then propose to freeze all convolutional layers and train only those layers. This allows us to adapt the pre-trained generator more flexibly by replacing the per-channel modulation with inter-channel modulation.

3 Method

Using a small dataset, we propose a method to adapt a pre-trained generative model to a different domain than the pre-trained one. Extending the work of Noguchi and Harada [12], we introduce adaptive point-wise grouping convolution to achieve more flexible domain adaptation. This means that instead of per-channel feature modulation, we use inter-channel feature modulation.

3.1 Adaptive point-wise convolution

In this work, we use point-wise convolution, which is a component of depth-wise separable convolution, as a channel selection method. A depth-wise separable convolution is an important component of many efficient neural network architectures [5]. It consists of two kinds of convolutional layers. The first layer is a depth-wise convolution layer, which performs lightweight filtering by applying one convolutional filter per input channel. The second layer is a 1×1 convolution layer, called a point-wise convolution, which builds new features by computing a linear combination of the input channels.

We briefly analyze the proposed method in terms of channel selection. Applying a point-wise convolution and computing the linear combination of the input channels is equivalent to the computation of a fully-connected layer in the channel direction, which can be represented as the following convolutional operation:

$$\mathbf{x}_{\mathbf{Adapt}} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{1}$$

Here, $\mathbf{x}$ represents a feature vector over all the channels on a certain pixel (or position) of the input feature map, and $\mathbf{x}_{\mathbf{Adapt}}$ represents the output vector of the corresponding pixel. Since the computation of a point-wise convolution is independent between each pair of the pixels, we can regard a point-wise convolution as a point-wise fully-connected layer. $\mathbf{W}$ and $\mathbf{b}$ represent a weight matrix and a bias vector of a point-wise convolution, respectively. When the number of input channels is c_{in} and the number of output channels is c_{out}, they are $\mathbf{W} \in \mathbb{R}^{c_{out} \times c_{in}}$ and $\mathbf{b} \in \mathbb{R}^{c_{out}}$. In case of "adaptive" point-wise convolution, we vary $\mathbf{W}$ and $\mathbf{b}$ adaptively by generating them dynamically with an external fully connected (FC) layer. Varying $\mathbf{W}$ means adjusting the weights of a linear combination of the channel elements, while changing $\mathbf{b}$ means adjusting the threshold of the ReLU activation just after the adaptive point-wise layer. In the proposed method, the parameters of the adaptive point-wise convolution, $\mathbf{W}$ and $\mathbf{b}$, are generated in one fully connected layer from the potential vector z. This allows us to adapt the model more flexibly to images in new domains.

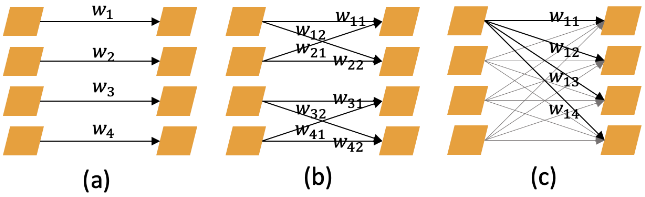

Fig. 1. (a) Channel-wise modulation (BN). (b) Limited cross-channel modulation. (c) Full cross-channel modulation.

3.2 Reduction of training parameters

Adaptive point-wise convolution allows for more complex representations using a linear combination of channels compared to the class conditional batch normalization (BN) used in work of Noguchi and Harada [12]. However, the number of parameters in the weights $\mathbf{W}$ is $c_{in} \times c_{out}$, which is much larger than the number of weights c_{out} in conditional BN. This may lead to overfitting when training on small datasets. Therefore, as a way to reduce the number of parameters, we apply the idea of a grouped convolution into an adaptive point-wise convolution. In a grouped convolution, an input feature map is grouped in the channel direction, and a convolutional operation is applied among each of the groups. If the number of channels to be grouped is equal to the number of channels in the convolutional layer, we can represent depth-wise convolution. If we make a point-wise convolution to a depth-wise point-wise convolution, it means channel-wise weighting which is equivalent to the γ weight of a batch normalization layer. Figure 1 depicts the differences among conditional BN, point-wise grouped convolution, and point-wise convolution, which corresponds to channel-wise modulation [13], limited cross-channel modulation, and fully cross-channel modulation, respectively. In this paper, limited cross-channel modulation is adopted to balance the number of parameters and the flexibility of feature modulation.

In this work, we propose to use a point-wise grouped convolution instead of a normal point-wise convolution which requires a $c_{in} \times c_{out}$ weight matrix. We restrict the number of training parameters for point-wise grouped convolution to 2, 4, and 8 times that of class conditional batch regularization.

3.3 Training

In case of GAN, the discriminator distinguishes between real training images and fake generated images, and the generator generates realistic images by performing adversarial training. However, this method assumes that the distribution of training samples can be densely filled. If the number of training samples is small, overfitting for small datasets will occur and the learning will become unstable. Therefore, it is desirable to train in a supervised training framework such as VAE. To train the generator by supervised learning, we also optimize the latent vector z corresponding to the training image, following Noguchi and Harada [12]. The proposed network is shown in Figure 2. During training, we optimize the loss function L, which is modeled as a distance function from the target image. The loss function L is similar to [12] in that it uses the L1 loss, which is the pixel-level distance, and the Perceptual loss, which is the semantic-level distance.

We assume that the generative network has been pre-trained on a large dataset such as ImageNet. In the fine-tuning step using a small dataset, we first insert an adaptive point-wise grouped convolutional layer with a corresponding fully connected (FC) layer immediately after the batch normalization layer and then fine-tune the parameters of the FC layer while keeping all the parameters of the original generative network frozen. Similar to [12], the latent variables are also updated during the fine-tuning.

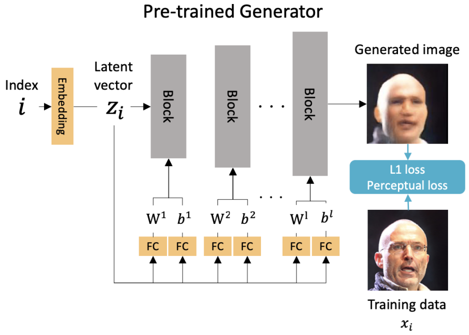

Fig. 2. The proposed model. The yellow blocks in the figure represent the trainable layers. During training, the latent variable z and the parameters of adaptive point-wise convolution are updated to minimize L1 loss and perceptual loss.

3.4 Inference

At the time of inference, a random image can be generated by inputting a randomly sampled vector z to the generator based on the standard normal distribution. However, since the generator only learns the relationship between the potential vector and the sparse training sample, its performance degrades for z that is far from the training sample. To solve this problem, we sample z from the truncated normal distribution as in Noguchi and Harada [12]. This technique is known as the truncation trick [1]. In the same way as [12], 0.4 was used as the truncation threshold.

4 Experiments

Several experiments were conducted to evaluate the stability of the proposed method in generating images from small datasets. We also compared our method with existing studies. For the generator, BigGAN [1] was used, referring to Noguchi and Harada [12]. We used an image size of 128x128. In all the experiments, we used the BigGAN-128 model pre-trained on ImageNet which consisted of five ResBlocks.

4.1 Datasets and metrics

The datasets used in the experiments were the FFHQ dataset [6], the passion-flower image from the Oxford 102 flower dataset [11], the African firefinch image from the 260 Bird Species dataset [1], and the African firefinch image in the Cars dataset [8], and an image of a BMW in the Cars dataset [8]. The domains used in this experiment, “Human face”, “Passionflower”, “African firefinch”, and “BMW” are not included in the ImageNet class, and thus can be regarded as the different domains from ImageNet. As an evaluation metric, to evaluate the quality of the generated images, KMMD (Kernel Maxi-mum Mean Discrepancy) was employed. KMMD is characterized by its ability to produce stable results even when the number of images in the dataset is small. The KMMD was computed using a Gaussian kernel between images pre-trained in an inception network consisting of training and generated images. The lower the KMMD, the better the quality. Note that the hyperparameters of KMMD used in the experiment are different from those used in Noguchi and Harada [12] and thus show different values.

4.2 Comparison with the baseline

The method (2) of the few-shot GAN explained in Section 2.2 was used as the baseline. Whereas [12] updates the scale and shift parameters of the batch statistics for each channel, in our work we adapt the domain by updating the parameters of the adaptive point-wise grouped convolution. Our method has the advantage of being able to mix multiple channel activations in a point-wise convolution, which allows for more flexible adaptation. Although this method has the advantage of allowing more flexible adaptation, the number of parameters can be huge. Therefore, we use grouped convolution to limit the number of training parameters. The number of parameters in [12] is used as a reference. By varying the number of groupings, we compare the quality of the generated images by the proposed method when the number of parameters is increased to 2, 4, and 8 times. Here, the baseline method [12] corresponds to the case where the number of channels to be grouped is set to the same value as the number of channels in the convolutional layer. We generated images in the “Human face” domain using 25, 50, and 100 images sampled from the FFHQ dataset [6]. The experimental results are shown in Table 1 ,which shows that the quality of the proposed method improves as the number of parameters increases. This indicates that the adaptive point cloud convolutional layer increases the variation of feature channels by combining activations across multiple channels. We also ran an experiment without additional grouping. In this case, the number of parameters was about 765, which was too large to train. Therefore, we set the batch size to 1 to train the model, taking into account the memory limit of the GPU. Unfortunately, the training failed due to insufficient memory on the GPU.

4.3 Experiments with additional datasets

In this experiment, we used all four datasets to compare our method with the baseline [12]. We used a sampling of 25, 50, and 100 images from each of the four datasets. Our method used an adaptive point-wise grouped convolution with parameters eight times larger than the baseline [12]. Figure 3 shows some generated images for the four datasets with three kinds of sample numbers, and Table 2 shows its qualitative results on them.

[1] https://www.kaggle.com/gpiosenka/100-bird-species

From Figure 3, it is clear that the proposed method can generate more detailed images than the baseline. In Table 2, the proposed method showed higher quality than the baseline method for all the datasets and all the numbers of training samples. This is because we reuse the previously learned feature channels and combine them among the channels to learn the best representation. Compared to the baseline method which modifies the activation within each channel by scaling or shifting, the proposed method achieves more flexible representation.

Figure 4 shows the result of interpolation between two randomly generated latent vectors. Despite the small amount of training data, the interpolation is clearer, smoother, and more stable than the baseline [12].

Table 1. Relation between the number of parameters and quality of the generated images.

Model	Parameter ratio	Number of data	KMMD
[12]	1	25	2.966
		50	2.507
		100	2.509
Ours	2	25	2.944
		50	2.496
		100	2.493
	4	25	2.942
		50	2.491
		100	2.490
	8	25	**2.928**
		50	**2.485**
		100	**2.487**

Table 2. Quantitative comparison.

Dataset	Model	Number of data	KMMD
Human face	[12]	25	2.966
		50	2.507
		100	2.509
	Ours	25	**2.928**
		50	**2.485**
		100	**2.487**
Passion flower	[12]	25	2.976
		50	2.977
		100	2.965
	Ours	25	**2.955**
		50	**2.960**
		100	**2.954**
African firefinch	[12]	25	2.965
		50	2.531
		100	2.532
	Ours	25	**2.937**
		50	**2.493**
		100	**2.506**
BMW	[12]	25	2.969
		50	2.522
		100	2.518
	Ours	25	**2.934**
		50	**2.487**
		100	**2.498**

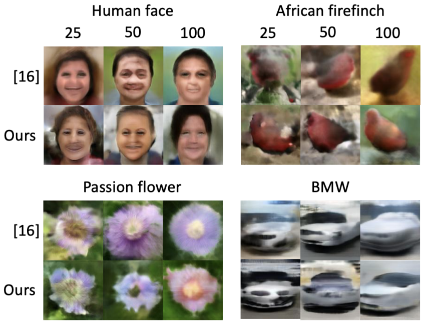

Fig. 3. Qualitative evaluation for the four dataset.

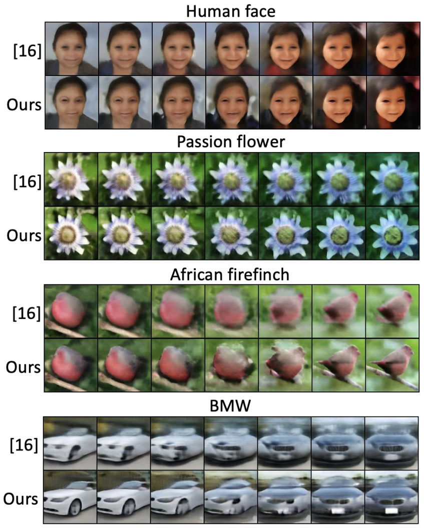

Fig. 4. Interpolation between two images.

5 Conclusions

In this work, we proposed a simple and effective method for generating images from small datasets. By fine-tuning the FC layers which dynamically generates the parameters of the adaptive point-wise grouping convolution, the proposed method is able to generate new images from much fewer images than required for training a regular generator, using prior knowledge of the pre-trained generator. The results show that the proposed method is able to synthesize higher-quality images with fewer training datasets than the existing baseline methods. This suggests that cross-channel modulation is more flexible and adaptable than per-channel modulation. In the future, we plan to investigate the possibility of generating higher-quality images with fewer data sets.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: Proc. of International Conference on Learning Representation (2019)
2. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv:1406.2661 (2014)
3. Hong, Y., Niu, L., Zhang, J., Zhang, L.: MatchingGAN: Matching-based few-shot image generation. In: International Conference on Multimedia and Expo (2020)
4. Hong, Y., Niu, L., Zhang, J., Zhao, W., Fu, C., Zhang, L.: F2GAN: Fusing-and-filling gan for few-shot image generation. In: Proc. of ACM International Conference Multimedia (2020)
5. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017)
6. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proc. of IEEE Computer Vision and Pattern Recognition (2019)
7. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences p. 114(13):3521–3526 (2017)
8. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
9. Li, Y., Zhang, R., Lu, J., Shechtman, E.: Few-shot image generation with elastic weight consolidation. In: In: Proc. of Neural Information Processing Systems (2020)
10. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: Proc. of International Conference on Learning Representation (2018)
11. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proc. of Indian Conference on Computer Vision, Graphics and Image Processing (2008)
12. Noguchi, A., Harada, T.: Image generation from small datasets via batch statistics adaptation. In: Proc. of IEEE International Conference on Computer Vision (2019)
13. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: FiLM: Visual reasoning with a general conditioning layer. In: Proc. of AAAI Conference on Artificial Intelligence (2018)
14. Robb, E., Chu, W.S., Kumar, A., Huang, J.B.: Few-shot adaptation of generative adversarial networks. arXiv:2010.11943 (2020)