A Food Dish Image Generation Framework Based on Progressive Growing GANs

Su Wang[1], Honghao Gao[2(✉)], Yonghua Zhu[3], Weilin Zhang[1], and Yihai Chen[1]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
{wongsou, zeroized, yhchen}@shu.edu.cn
[2] Computing Center, Shanghai University, Shanghai, China
gaohonghao@shu.edu.cn
[3] Shanghai Film Academy, Shanghai University, Shanghai, China
zyh@shu.edu.cn

Abstract. The generative adversarial networks (GANs) have demonstrated the ability to synthesize realistic images. However, there are few researches applying GANs into the field of food image synthesis. In this paper, we propose an extension to GANs for generating more realistic food dish images with rich detail, which adds a food condition that contains taste and other information. That makes the model generate images with rich details. To improve the quality of the generated image, the taste information condition is added to each stage of the generator and discriminator. First, the model learns embedding conditions of food information, including ingredients, cooking methods, tastes and cuisines. Secondly, the training model grows progressively, and the model learns details increasingly during the training process, which allows the model to generate images with rich details. To demonstrate the effectiveness of our proposed model, we collect a dataset called Food-121, which includes the names of the food, ingredients, cooking methods, tastes, and cuisines. The results of experiment show that our model can produce complex details of food dish image and obtain high inception score on the Food-121 dataset compared with other models.

Keywords: GANs · Food dish image synthesis · Food dataset

1 Introduction

Generative Adversarial Networks (GANs) [1] were first proposed by I. Goodfellow and some pioneer researchers in 2014. Since then, they have been successfully used in the image generation area. GANs perform well on datasets with single object in the image, such as human faces in the CelebA dataset [2], birds images in the CUB dataset [3] and flowers images in the Oxford-102 dataset [4]. But the generated images are not realistic enough when multiple irregularly shaped objects exist in the images. The images do not have realistic and rich details, and this is especially obvious in food dish image generation, as there are often many ingredients in food images, and the ingredients have various visual effects. Hence, GANs can probably not generate realistic food dish image only through image data.

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 323–333, 2019.
https://doi.org/10.1007/978-3-030-30146-0_22

The GANs-based approach is too uncontrollable; therefore, conditional constraints are added to GANs, which is called conditional GANs (CGANs) proposed by Mirza [5]. Karras [6] proposed progressive growing GANs (PGGANs) which generate images by progressively increasing the resolution. PGGANs learn the structure of images at first and then focus on details so that the images look realistic and have higher resolution. Hamada [7] proposed a PGGAN to generate full-body high-resolution anime images by adding a structure condition.

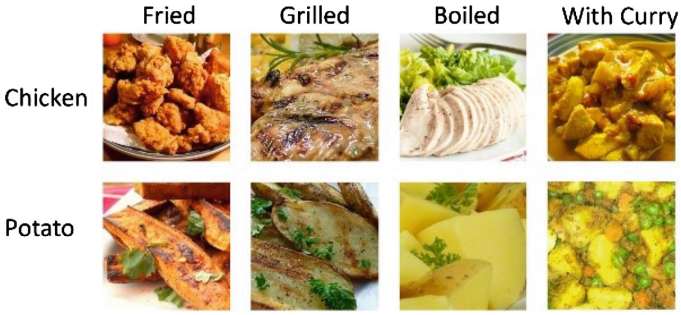

Fig. 1. Different images of ingredients with different cooking methods

The visual appearance of the food is usually determined by the ingredients. However, the relationship between the ingredients and the corresponding food image is complicated, so simply generating food images from the ingredients may cause large deviations. Figure 1 shows the food images of chicken and potatoes cooked using different cooking methods. As shown in Fig. 1, when chicken is cooked by different methods, the colors and textures are quite different. Such phenomena can also be seen on potatoes. On the other side, when chicken and potatoes are cooked separately using the same method, their images share many visual features. Other ingredients in the recipes also have this similarity. Therefore, the cooking method has a great impact on the food dish image and plays an important role in determining the appearance of cooked food.

In addition to the cooking method, taste and cuisine of the food also impact the appearance of the image. Therefore, it is necessary to take food information that contains ingredients, cooking method, taste, and cuisine into account when generating images. To deal with textual food information, Salvador [8] separately processes the ingredients and instructions, where ingredients are represented as a vector with word2vec method and instructions are encoded with skip-thought [14]. The recipe representation is thus obtained through concatenating the ingredient and the instruction representation.

In this paper, we propose a generative adversarial network specialized for food image generation, and collect a dataset, Food-121, that contains full food information and its corresponding image. We describe its comparison with other datasets in Sect. 4. Our model uses the architecture of CGANs, where food information representation works as the condition so that the model can utilize ingredients, cooking method, taste, and cuisine to generate better food images.

The remainder of this paper is organized as follows. Section 2 briefly introduces the preliminary and the related works. Section 3 explains our method in detail. In Sect. 4, our proposed dataset Food-121 is presented, and the experimental results are discussed. Finally, the conclusion and future works are given in Sect. 5.

2 GANs

2.1 Preliminary

The framework of GANs consists of two competitors, generator and discriminator. The task of the generator is to generate a sample that the discriminator cannot discriminate between real and fake. At the same time, the task of the discriminator is to discriminate between the real image and the sample generated by the generator [1]. In training, generator G that inputs the random noise vector and outputs the fake data. Discriminator D inputs the real data or fake data and outputs the possibility that it is real data or fake data. To generate a fake image, the training method achieves the best results by pitting generator G and discriminator D against each other. The process makes the sample data distribution close to the true data distribution. The value function of GANs is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

The value function of GANs is shown in (1) [1]; G represents the generator, D represents the discriminator, p_{data} represents for the real data distribution, and p_z represents the noise distribution. The goal of D is to maximize $V(D, G)$, and the goal of G is to minimize $V(D, G)$. G and D are trained at the same time. Finally, G can estimate the distribution of the real samples.

CGANs are an extension of the GANs that are used to solve the problem of uncontrollable image in GANs. In CGANs, conditional information is added to both the generated model G and the discriminant model D to guide the training of the model when using GANs [5]. The difference between GANs and CGANs is that CGANs add a conditional input vector y to the random noise z.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (2)$$

As is shown in (2) [5], the value function of conditional GANs is a two-person minimax game with a condition. The conditional constraint y is simultaneously added to both the generator model G and the discriminator model D to guide the data generation process. The condition can be any additional information. The noise z and the condition y are input to the generator. The data x and the condition y are input as inputs simultaneously to the discriminator. After the conditional input vector y concatenate is in the noise vector z, the generated vector is input to the generator G. Then training is in the same way as GANs.

2.2 Related Works

Salvador [8] established a large-scale, structured recipe dataset, Recipe1 M, which contains more than 1,000,000 recipes and 800,000 food images. As the largest public recipe dataset, Recipe1 M provides the ability to train high-performance models. Using these data, they trained a neural network to find the joint embedding of recipes and images to complete the image-recipe retrieval task. In addition, it was demonstrated that regularization via the addition of a high-level classification objective both improves retrieval performance to rival that of humans and enables semantic vector arithmetic.

To solve the problem of generating images by words, Reed [9] proposed a GAN with learning interpolation and a matching aware generator GAN-INT-CLS model. GAN-INT-CLS generates images through descriptive text. El [10] proposed a GAN to generate food images through recipes. Zhang [11] proposed the stacked GAN model to improve the resolution of generated images. Zhang [12] proposed a GAN model that combined the attention mechanism with GANs.

3 Methodology

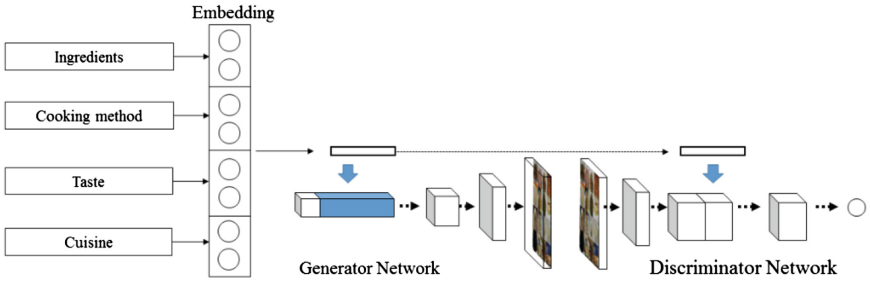

Fig. 2. The overall architecture of our model.

Figure 2 shows the architecture of our model. There are three parts in our model: food text encoder, generator network, and discriminator network. In the training process, food text encoder first transforms the four parts of an input text to vectors, concatenate these vectors to a feature map and feed the feature map into the generator network. Next, the generator network generates an intermediate image. Finally, the discriminator network compares the image and the true image concatenated with food condition and outputs the probability of the image being faked.

3.1 Food Text Encoder

The food information contains four parts: the ingredients, the cooking method, the taste, and the cuisine. For ingredients, the food information encoder learns an ingredient level word2vec representation. The cooking method is encoded by a skip-instruction model.

The skip-instructions [8] model is proposed to obtain the representation of the instructions. Skip-instructions is a kind of sequence-to-sequence [13] models and is based upon skip-thoughts technique. Skip-thoughts encodes a sentence and uses that encoding as context when decoding the previous and next sentences. Skip-instruction adds a start and end to the instructions and uses an LSTM instead of a gated recurrent unit (GRU).

Skip-thought [14] is an unsupervised learning sentence encoder, and sentences with similar semantics are mapped to similar vector representations. The encoder uses the following function:

$$r^t = \sigma(W_r x^t + U_r h^{t-1}) \tag{3}$$

$$z^t = \sigma(W_z x^t + U_z h^{t-1}) \tag{4}$$

$$\bar{h}^t = \tanh(W x^t + U(r^t \odot h^{t-1})) \tag{5}$$

$$h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \bar{h}^t \tag{6}$$

where $\bar{h}^t$ is the proposed state update at time t, z^t is the update gate, r_t is the component-wise product of reset gate $\odot$.

The decoder uses the following formula:

$$r^t = \sigma(W_r^d x^{t-1} + U_r^d h^{t-1} + C_r h_i) \tag{7}$$

$$z^t = \sigma(W_z^d x^{t-1} + U_z^d h^{t-1} + C_z h_i) \tag{8}$$

$$\bar{h}^t = \tanh(W^d x^{t-1} + U^d(r^t \odot h^{t-1}) + C h_i) \tag{9}$$

$$h_{i+1}^t = (1 - z^t) \odot h^{t-1} + z^t \odot \bar{h}^t \tag{10}$$

The decoder is a natural language model with the condition on the output h_i of the encoder. Matrix C_z, C_r and C are used to bias the update gate, reset gate, and hidden state computation by the sentence vector.

For taste and cuisine, we use the word2vec level representation. The cooking method is embedded by a skip-instruction encoder. Then, we obtain the food condition vector by concatenating four representations.

3.2 Conditional Progressive Growing Generator and Discriminator Network

Our model is trained in a progressively increasing manner [6]. To generate higher resolution images, the model starts with training resolution of 4 $\times$ 4 pixels in generator and discriminator, then incrementally add layers to generator and discriminator and evaluate the resolution.

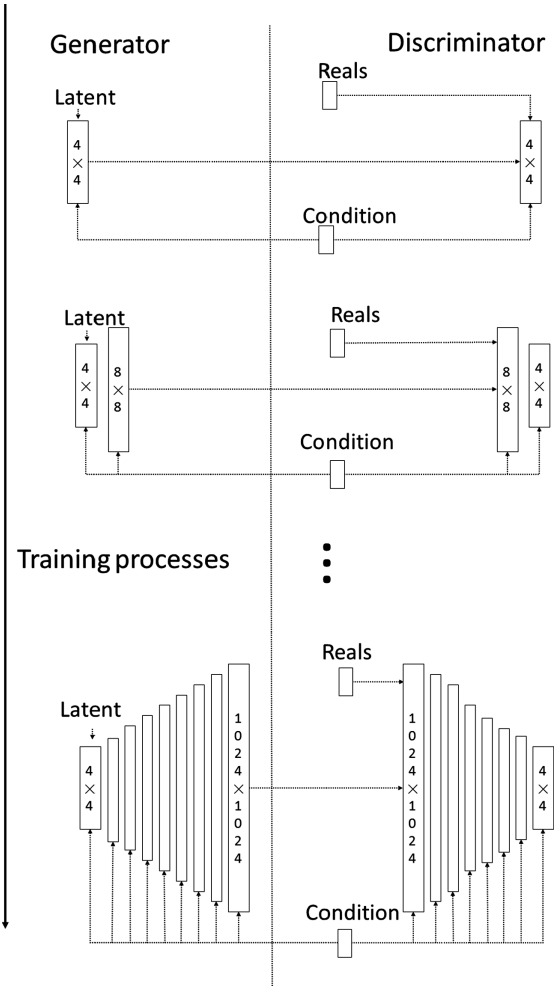

Fig. 3. Generator (G) and Discriminator (D) architecture of our model

Figure 3 illustrates the training process of the generator and the discriminator of our model. A latent vector concatenated with food information condition is input to the initial generator, where at the beginning a low-resolution image is generated. Then the generated image, the truth image, and with the food information condition are fed to the discriminator network together, and the discriminator outputs the possibility that the generated image is not faked. In the subsequent epochs, the generator progressively generates images of higher resolution. Therefore, the quality of the generated food images is gradually improved, and the details of the images become clearer and richer during the process.

4 Experiment

4.1 Dataset

Table 1. Comparison between datasets

	Recipes	Images	Difference
Food-121	121,478	121,478	with taste and cuisine information
Recipe1 M	1,000,000	800,000	–
Food-101	–	101,000	–
VIREO Food-172	65284	110241	only Chinese cuisine

In this section, we introduce our dataset, Food-121, which contains full food information. There are a few datasets about recipes and food images, such as Recipe1 M dataset [8], Food-101 dataset [15], and VIREO Food-172 dataset [16]. Recipe1 M consists of over 1,000,000 recipes and 800,000 food images, but the recipes are only about ingredients and cooking instructions. Food-101 contains 101,000 images in 101 ingredients. But VIREO Food-172 does not have taste and cuisine information. And it is only for Chinese cuisine.

To explore the impact of taste and cuisine on the food images, data are collected from websites. The texts of ingredients, cooking method, taste, and cuisine are extracted from raw HTML and download the corresponding food images. Samples with unclear expression of food information or with blurred images are removed. Eventually, Food-121 dataset contains 121,478 pieces of food information and image. The statistical data of Food-121 and three prior datasets are listed in Table 1, and an example of Food-121 data item is shown in Fig. 4.

Fig. 4. An example of Food-121 date item

4.2 Evaluation Metrics

It is necessary to evaluate the generation model in two aspects: whether the generated image is clear and whether the generated image is diverse. If the generated image is not clear enough, this obviously shows that the generated model is not performing well. When the generated image is clear enough, it still need to determine whether the model

can generate enough of variety. Therefore, the inspection score (IS) [17] is used to evaluate our model in both aspects.

Considering the above two aspects, the formula of Inception Score is:

$$\mathbf{IS}(G) = \exp(E_{\boldsymbol{x} \sim p_g} D_{KL}(p(y|x)||p(y)) \tag{11}$$

The generated sample image x is input into Inception V3. Vector y of 1,000 dimensions is output. The basis for the IS to determine the authenticity of the data is derived from the training set of Inception V3.

However, IS also has problems. Only the generated samples are considered, and the real data are not considered because the IS cannot evaluate the distance between the real data and the sample. Therefore, the Fréchet inception distance (FID) [18] is also calculated. The FID calculates the distance between the real image and the fake image at the feature level. The formula for the FID is as follows:

$$\mathbf{FID} = ||m_r - m_g||^2 + Tr(C_r + C_g - 2(C_r C_g)^{1/2}) \tag{12}$$

Where m_r and m_g are the mean of the features of the real images and generated images, C_r and C_g are the covariance matrix of the features of the real images and the generated picture. The FID is a measure of the distance between two multivariate normal distributions.

Multiscale structural similarity (MS-SSIM) scores [19] represent a set of randomly sampled pairs of images within a given class.

$$\mathrm{SSIM}(x, y) = [l_m(x, y)]^{\alpha M} \cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \tag{13}$$

The exponents αM, β_j and γ_j are used to adjust the relative importance of different components. The mean of the MS-SSIM scores is used, in which a high mean MS-SSIM indicates mode collapse or low sample diversity.

4.3 Experiment Settings

The model uses the same loss function as [6]. Constricted by the graphics processing unit (GPU), the model trains networks with 10 k images and food conditions for each stage. The model uses a mini-batch size 12 and Adam [20] with β1 = 0, β2 = 0.99 to train the networks.

4.4 Results

In this section, experimental results are discussed. Figure 5 shows some generated images of our method and baseline methods. From these images, it can be seen that when there are multiple objects in the image, our model outperforms the baseline methods. That means our model can generate more realistic images with rich details with the help of food information. Our model can also learn various ingredients visuals in different tastes.

(a) (b) (c)

Fig. 5. Food image generation: comparison between GAN-INT-CLS (left), DCGAN (middle), and our model (right)

Figure 5 shows the results of our model and deep convolutional GAN (DCGAN) [21]. Our model learns the composition of the image that there is a dish with various foods because progressive growing GANs architecture is used. The training helps the model learn the rough composition of the images first. Then, the model focused on the details. Therefore, food images generated by our model look more realistic (Table 2).

Table 2. Inception and Fréchet inception distance between DCGAN and our model

	Inception score	Fréchet inception distance	MS-SSIM
DCGAN	3.32 ± 0.15	109.3248 ± 10.0526	0.1256
Our model	**4.56 ± 0.18**	**85.2194 ± 9.2543**	**0.0612**

The IS, FID, and MS-SSIM of DCGAN and our model are evaluated. The higher IS for the model is, the more realistic the generated images are. The smaller the FID is, the smaller distance between the generated images and the real images. In addition, MS-SSIM reflects the seriousness of the mode collapse problem of the model. A high MS-SSIM means a low sample diversity. Our model obtains a higher IS and a lower FID and MS-SSIM on the Food-121 dataset, which means that our model can generate more realistic and various food dish images (Table 3).

Table 3. Inception and Fréchet inception distance between PGGAN and our model

	Inception score	Fréchet inception distance	MS-SSIM
Progressive growing GAN	4.25 ± 0.12	95.7254 ± 8.4682	0.0965
Our model	**4.56 ± 0.18**	**85.2194 ± 9.2543**	**0.0612**

The IS, FID, and MS-SSIM of PGGAN and our model are evaluated. Our model obtained a higher IS and a lower FID and MS-SSIM on the Food-121 dataset, which means that our model can probably generate more realistic and various food images

than PGGAN. Besides, compared with DCGAN, the IS, FID, and MS-SSIM did not increase as much as the experiment between DCGAN and our model. Therefore, we hypothesize that the progressive growing training architecture can greatly increase the quality of generation. However, with the condition of the ingredients, cooking method, taste, and cuisine, the model can generate food images with more details.

However, our model is not sufficient for generating a background for the food image, probably because the background of the image has not been dealt with in the Food-121 dataset.

5 Conclusion

In this paper, we demonstrate the effectiveness of adding food information conditions to GANs to generate realistic food images. The quality of the generated food image can be improved by adding food information conditions to the generator and discriminator. It increases the resolution slowly, and adds the embedding of ingredients, taste, cuisine and cooking method to the generator and discriminator in each step, which makes it possible to generate controllable food images. The experimental result shows that our model performs better on the food dataset Food-121. The model's IS and FID are evaluated. Table 1 shows that our model has a higher IS and lower FID on the Food-121 dataset, which means our model performs food image generation well.

Our experiment was restricted by GPU memory constraints, so the model will be trained with more food images in the future. Our model sometimes generates some strange backgrounds. Thus, the image data in Food-121 are planned to clean and retain only the food portion in the picture. More high-resolution food images will be collected from websites to improve the quality of image generation. In addition, every food dish image will be tagged by its ingredients, taste, cooking method and cuisine, extract features from the image in each tag or multi-tag to make our model learn more about the food dish image features.

Acknowledgments. This work is supported by the National Key Research and Development Plan of China under Grant No. 2017YFD0400101, the Natural Science Foundation of Shanghai under Grant No. 16ZR1411200, and CERNET Innovation Project under Grant No. NGII201 70513.

References

1. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
2. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
3. Welinder, P., et al.: Caltech-UCSD birds 200 (2010)
4. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pp. 722–729

5. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 [cs.LG] (2014)
6. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: 6th International Conference on Learning Representations (2018)
7. Hamada, K., Tachibana, K., Li, T., Honda, H., Uchida, Y.: Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. In: European Conference on Computer Vision, pp. 67–74 (2018)
8. Salvador, A., et al.: Learning cross-modal embedding for cooking recipes and food images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3020–3028 (2017)
9. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: 33rd International Conference on Machine Learning, pp. 1060–1069 (2016)
10. El, O.B., Licht, O., Yosephian, N.: GILT: generating images from long text. arXiv preprint arXiv:1901.02404 (2019)
11. Zhang, H., et al.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915 (2017)
12. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318 (2018)
13. Sutskever, I., Vinyals, O., Le, Q. V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
14. Kiros, R., et al.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3294–3302 (2015)
15. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: European Conference on Computer Vision, pp. 446–461 (2014)
16. Chen, J., Ngo, C.W.: Deep-based ingredient recognition for cooking recipe retrieval. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 32–41 (2016)
17. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
18. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, pp. 6626–6637 (2017)
19. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, pp. 1398–1402 (2003)
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)