

Emojis generation based on deep convolution generative adversarial network

Kejia Tang

Data Science & Financial Mathematics and Statistics, The University of Sydney,
Sydney, 2006, Australia

ktan3747@uni.sydney.edu.au

Abstract. With the development of information technology and mobile communication, people's usage of emoji is increasing. However, designing emojis by artists can be time-consuming and costly. Therefore, this study attempts to use the Deep Convolution Generative Adversarial Network (DCGAN) method in deep learning to automatically generate emojis. DCGAN is a combination of Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN). It introduces convolutional networks into the generative model for unsupervised training, which can improve the learning effect of the generator network. DCGAN consists of two different models: a generator model and a discriminator model. The generator's role is to produce fake images that are similar to the training images, and the discriminator's role is to judge whether the fake image is the same as a real image. During training, the generator continuously tries to defeat the discriminator by producing better and better fake images, while the discriminator tries its best to distinguish between real and fake images. Through continuous optimization of the generator and discriminator, this study can finally obtain clear new emojis. Overall, the images generated by DCGAN can largely fulfill our purpose. Many novel emojis could be obtained to replace the work of human artists. However, the quality of the generated emojis has some defects, and it is still necessary to screen out images with strange shapes that are not suitable for daily life.

Keywords: DCGAN, emoji generation, deep learning.

1. Introduction

Emojis are commonly used in social media and may be considered a language for graphic illustration of emotions. Shigetaka Kurita invented emojis for Japanese operators of mobile phones in the late 1990s [1]. Emojis enhance the efficacy of computer-mediated communication by allowing users to convey more subtle emotions, such as cartoon face, food beyond text-based information. With the continuous development of mobile platforms, people's demand for emoji has increased significantly. However, the creation of emoticons is a sophisticated procedure that takes a long time for the artist. Because of a lack of imagination, the artist is only able to generate one emoji for each work. To solve this limitation, this study is attempting to employ artificial intelligence to aid in the development of emoticons in order to increase the diversity of emojis. This is a really significant and intriguing study area. This paper attempts to use the Deep Convolution Generative Adversarial Network (DCGAN) method to produce innovative

emojis that humans cannot imagine. Emoji pictures already in use serve as training data for the DCGAN, which subsequently generates new characters at random.

The application of deep learning in daily life becomes increasingly extensive. Especially in the field of generation, Generative Adversarial Network (GAN) has attracted the attention of many researchers. Goodfellow et al. introduced Generator adversarial networks for the first time in 2014 [2]. With the ongoing research of specialists in deep learning, GAN has improved continually and produced many popular architectures. Compared with GAN, DCGAN has made some adjustments and used transposed convolution operation [3]. The images generated by DCGAN are clearer and more colorful than GAN. In contrast to traditional GAN, conditional GAN's input consists of two components: random noise distribution on one side and conditional label input on the other. The new model can produce handwritten digit images on the MNIST dataset in response to digit class labels. Jung et al. used Conditional GAN with a 3D discriminator to generate MRIs showing Alzheimer's disease development [4].

CycleGAN can be used to convert photos across styles. However, matched training data won't be accessible for many jobs. Without paired examples, cycleGAN translates pictures from the original dataset to the goals. Zhu et al. have used this technique in a variety of industries [5]. For instance, after learning the database containing horses and zebras, the horses and zebras on fresh images are transformed to each other. Coupled Generative Adversarial Network might learn to produce comparable smiley and frowny faces, as well as blond-haired and non-blond-haired ones, and so on. Zhang et al. suggested using StackGAN to investigate text-to-image synthesis and obtained excellent results in their paper on StackGAN [6]. StackGAN consists of a pair of networks that, when provided with a textual description, can generate lifelike visuals. Text-to-image synthesis offers a wide range of practical applications, including synthesizing pictures from textual descriptions, translating textual narrative into comics, and building internal representations of textual descriptions.

In this regard, given the benefits of Deep Convolutional GAN emoji generation, it theoretically becomes a feasible option to build an emoji producing machine. This research describes a Deep Convolutional GAN application for generating emojis using datasets from Facebook, Google, and Apple, etc. Deep Convolutional DCGAN are ideal for image processing and generation tasks since they considerably increase the stability of GAN training and the quality of generated outcomes. As a result, this study employ DCGAN to create new emojis. In this paper, the emojis from different platforms was collected as comprehensively as possible, and then this study processed these emo. A better generator and discriminator structure than the basic GAN model was developed, and the generator and discriminator were continuously trained.

2. Method

2.1. Dataset description and pre-processing

The emoji image data used in this study is downloaded from the Kaggle website [7]. The emoji collection is obtained from online sources including Apple, Facebook and Google etc. The emojis are all 72 by 72 pixels in size, and they are based on the RGB colour model. The dataset is first constructed as an image folder that contains every emoji. Before training, every image is converted into 64×64 pixels for speeding up the training process and reducing the computational resource, which is suitable for the generator and discriminator structures. Figure 1 provides the sample data of the collected dataset. In addition, the developed data loader separates the dataset into 128 batch sizes. To fulfil a requirement, i.e. a normal distribution with mean=0, standard deviation=0.02, all convolutional, convolutional-transpose, and batch normalization layers are re-initialized.



Figure 1. The sample data of the collected dataset.

2.2. DCGAN

GAN is a data generation model that is widely used in recent years. GAN is composed of two networks, namely the generation network G (i.e. Generator) and the discriminative network D (i.e. Discriminator). G is responsible for generating a picture, it receives a random noise z , generates a picture through the noise, and records the generated picture as $G(z)$. The discriminator (D) serves the purpose of evaluating the authenticity of an image denoted by x . Specifically, $D(x)$ represents the probability that x corresponds to a genuine image. An output value of 1 indicates a certainty of 100% that x is a genuine image, whereas an output of 0 suggests that x is not an authentic image. During the training phase, the generator network (G) endeavors to create genuine images with maximal accuracy in order to outsmart the discriminative network (D). Conversely, D is aimed at distinguishing the artificially generated images from the genuine ones with utmost precision. This dynamic interplay between G and D constitutes a game-like framework. DCGAN is a deep convolutional network structure added on the basis of GAN, which specifically generates image samples. D and G in DCGAN adopt a relatively special structure in order to effectively model the image.

In the present investigation, the generator (G) receives a 100-dimensional vector, referred to as the noise vector (z). The initial layer of G constitutes a fully connected layer, whereby the 100-dimensional vector is transformed into a $4 \times 4 \times 1024$ -dimensional vector. Subsequently, transposed convolution is employed from the second layer onwards to sample and progressively diminish the number of channels. Ultimately, the output of G comprises a three-channel image measuring 64 units in width and height. Please refer to Figure 2 for an illustration of the exemplar generator architecture.

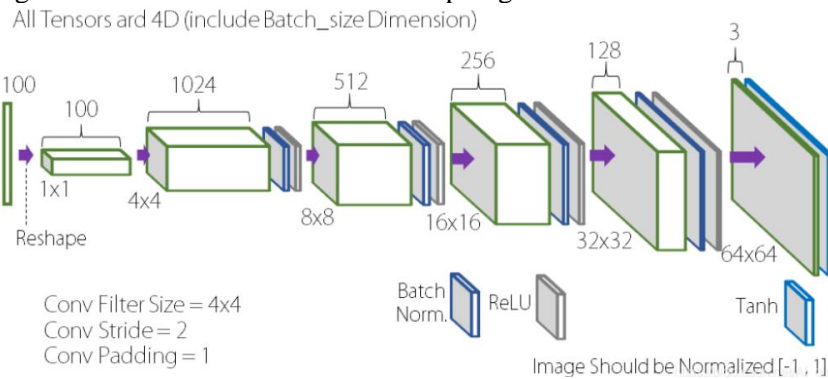


Figure 2. The sample architecture of the generator [8].

Regarding the discriminator (D) depicted in Figure 3, its input corresponds to an image, while its output represents the probability that the input image is a genuine one. Within the framework of Deep Convolutional Generative Adversarial Networks (DCGAN), the structure of D comprises a convolutional neural network, which has been proven to be effective in various computer vision tasks [9-11]. The input image undergoes convolution through multiple layers to obtain a convolutional feature, which is subsequently processed by the logistic function, yielding an output that can be interpreted as a probability estimate.

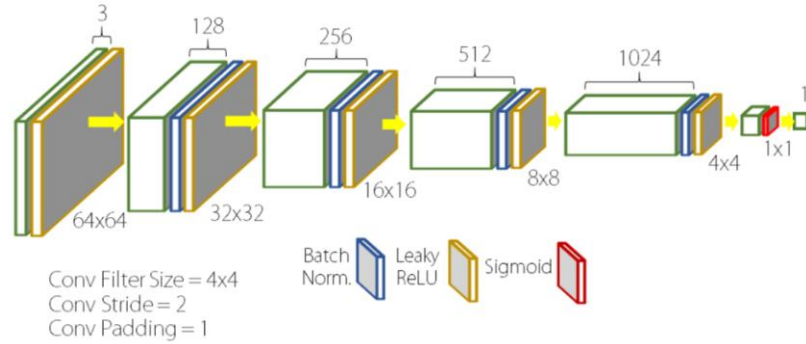


Figure 3. The sample architecture of the discriminator [8].

2.3. Implementation details

DCGAN employed the Adam optimization algorithm during training and this study set parameters (i.e. learning rate = 0.0005). Python was used to carried out the experiments for data preparation and DCGAN modelling. The testing was carried out on a PC with 8GB of RAM running the Windows 10 operating system [12]. The binary cross-entropy loss function was the one employed in our experiment. The calculation of the loss function is done in the discriminator. Since the discriminator's output is a binary classification of TRUE/Fake, the binary cross-entropy function is used overall. If the discriminator is required to accurately distinguish the generated images, $1-D(G(z))$ needs to be as close to 1 as possible. For training the generator, the discriminator part should be kept fixed so that the generated data can be classified by the discriminator, i.e., $D(G(z))$ approaches 1, which validates the generated data through the discriminator [13].

$$l(x, u) = L = \{l_1, \dots, l_N\}^T, \quad l_n = -[y_n \times \log x_n + (1 - y_n) \times \log(1 - x_n)] \quad (1)$$

3. Result and discussion

The generated images (100 epochs) obtained by the proposed DCGAN model can be found in Figure 4. In addition, the corresponding loss curve is presented in Figure 5.

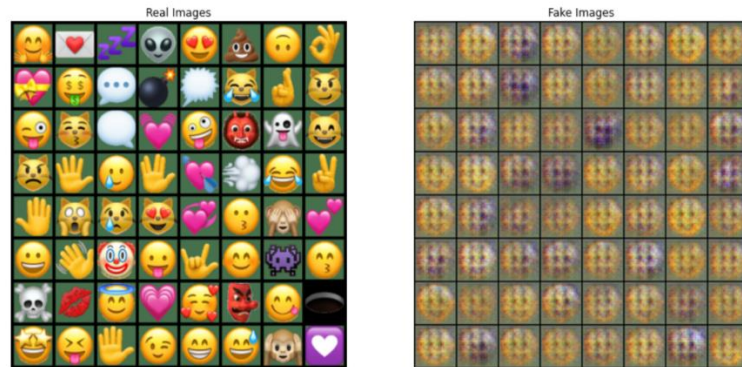


Figure 4. Real emojis collected on the platform VS Emoji obtained after 100 iterations.

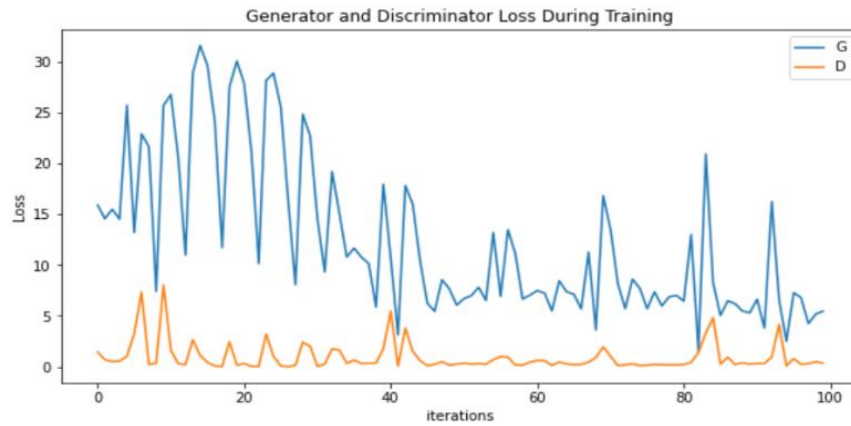


Figure 5. The losses of generator and discriminator during 100 epochs.

The generated images (500 epochs) obtained by the proposed DCGAN model can be found in Figure 6. In addition, the corresponding loss curve is presented in Figure 7.



Figure 6. Real emojis collected on the platform VS Emoji obtained after 500 iterations.

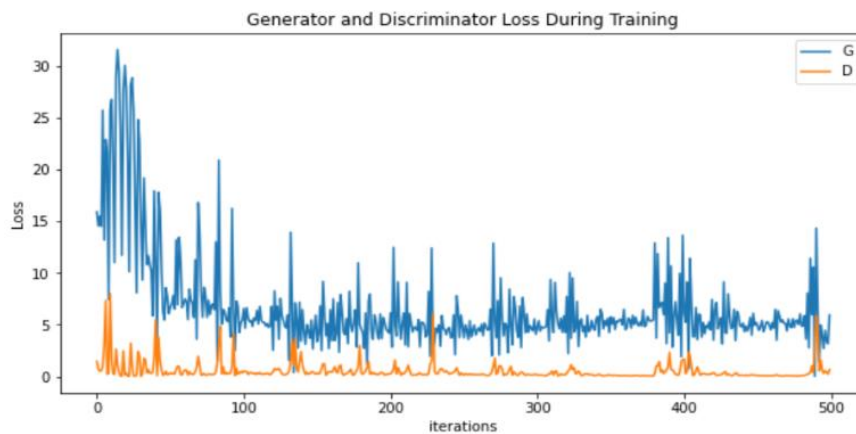


Figure 7. The losses of generator and discriminator during 500 epochs.

The generated images (1000 epochs) obtained by the proposed DCGAN model can be found in Figure 8. In addition, the corresponding loss curve is presented in Figure 9.



Figure 8. Real emojis collected on the platform VS Emoji obtained after 1000 iterations.

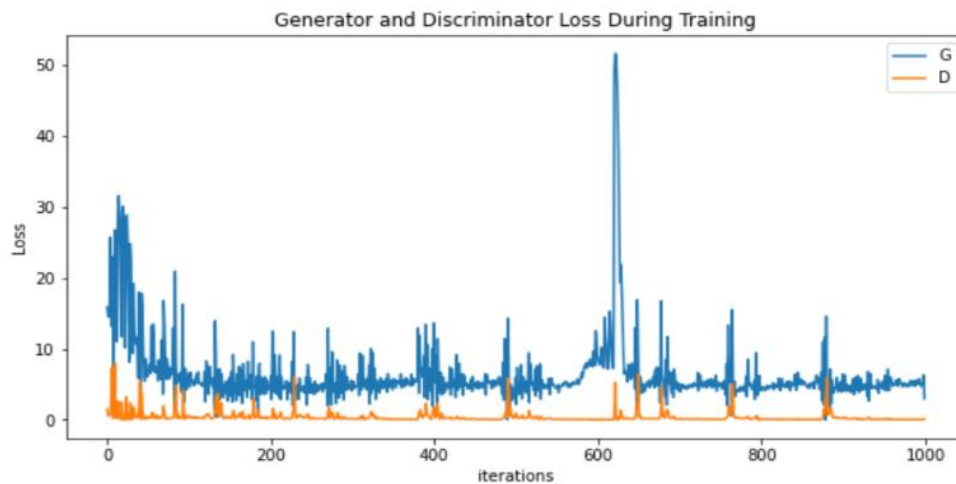


Figure 9. The losses of generator and discriminator during 1000 epochs.

The figures presented above reveals that the generated emojis remain notably unclear even after 200 iterations. Upon reaching 500 iterations, the generated images demonstrate a rudimentary emoji shape, yet continue to possess a significant degree of blurriness. However, the quality of the generated emojis drastically improves after the model reaches 1000 iterations. It is observed that the authenticity of the generated emojis increases with an increase in the number of training iterations. Additionally, the emojis generated from the images exhibit a considerable level of diversity, with certain emojis displaying unconventional shapes that diverge from typical emoji usage, while others boast innovative and well-defined shapes. As the number of training iterations increases, the generator and discriminator losses progressively stabilize and fluctuate within a specific range.

GAN continuously optimizes the generator and discriminator through the game, making the generated images more and more similar to the real ones in distribution. Because the new emojis are randomly assembled from the training set, some of the generated fake images are not very logical, while some of the generated images successfully conform to people's usage logic. There are still some limitations to this result. For example, even if the number of iterations reaches 1000, the generated fake images are still not clear enough. Secondly, some of the images are generated from faces and hands, and the results do not conform to people's daily usage habits. Further screening of these strange images is required.

4. Conclusion

The primary objective of this research paper is to employ deep learning techniques to generate a range of innovative emojis, thereby circumventing the exorbitant expenses associated with employing artists for their creation. In this study, the DCGAN approach is leveraged to learn from existing emojis on the network platform, train the generator and discriminator, and optimize parameters through the Adam optimizer to generate novel emojis. Through continuous iterations, a reasonably good generated image was produced. Upon examining the images, it is evident that some of the generated images are imaginative and visually striking. Moving forward, there exist several significant research opportunities to pursue, including collecting extant emojis in the human world for learning and ascertaining automatically whether the newly generated emojis conform to prevalent usage patterns.

References

- [1] Ljubešić N Fišer D (n.d.). 2016 A Global Analysis of Emoji Usage [https://aclanthology.org/W16-a\)2610.pdf](https://aclanthology.org/W16-a)2610.pdf)
- [2] Goodfellow I Pouget-Abadie J Mirza M et al. 2020 Generative adversarial networks Communications of the ACM 63(11): 139-144
- [3] Li Z Wan Q 2021 Generating Anime Characters and Experimental Analysis Based on DCGAN Model 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI) IEEE 27-31
- [4] Jung E Luna M Park S H 2022 Conditional GAN with 3D Discriminator for MRI Generation of Alzheimer's Disease Progression. Pattern Recognition 109061
- [5] Zhu J Y Park T Isola P Efros A A 2017 Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks Openaccess.thecvf.com. https://openaccess.thecvf.com/content_iccv_2017/html/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.html
- [6] Zhang H Xu T Li H et al. 2017 Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks Proceedings of the IEEE international conference on computer vision 5907-5915
- [7] Kaggle 2021 Full Emoji Image Dataset. (n.d.). <https://www.kaggle.com/datasets/subinium/emojiiimage-dataset>
- [8] Lee H 2022 DCGAN in Tensorflow. GitHub. https://github.com/HyeongminLEE/Tensorflow_DCGAN
- [9] Yu Q Wang J Jin Z et al. 2022 Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training Biomedical Signal Processing and Control 72: 103323.
- [10] Mahmoud S S Pallaud R F Kumar A et al. 2023 A Comparative Investigation of Automatic Speech Recognition Platforms for Aphasia Assessment Batteries Sensors 23(2): 857
- [11] Li R Gao R Suganthan P N 2023 A decomposition-based hybrid ensemble CNN framework for driver fatigue recognition Information Science 624: 833-848
- [12] Xu C 2021 Analysis of Emoji Generation Based on DCGAN Model 2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI) Shenyang Chin pp 130-134
- [13] Xu W and Xu Z 2022 Visual Performance of DCGAN Model for Analyzing Emoji Generation 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA) Dalian pp. 886-889