JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 1 | June 2023: 44-51
DOI: 10.15575/join.v8i1.790

Implementation of Generative Adversarial Network to Generate Fake Face Image

Jasman Pardede[1], Anisa Putri Setyaningrum[2]
[1,2]Department of Informatics, Faculty of Industry, Institut Teknologi Nasional Bandung, Indonesia

Article Info

Article history:

Received July 21, 2021
Revised March 29, 2022
Accepted May 10, 2023

Keywords:

Fake Image
LSGAN
Original Image
Supervised Contrastive Loss

ABSTRACT

In recent years, many crimes use technology to generate someone's face which has a bad effect on that person. Generative adversarial network is a method to generate fake images using discriminators and generators. Conventional GAN involved binary cross entropy loss for discriminator training to classify original image from dataset and fake image that generated from generator. However, use of binary cross entropy loss cannot provided gradient information to generator in creating a good fake image. When generator creates a fake image, discriminator only gives a little feedback (gradient information) to generator update its model. It causes generator take a long time to update the model. To solve this problem, there is an LSGAN that used a loss function (least squared loss). Discriminator can provide a strong gradient signal to generator update the model even though image was far from decision boundary. In making fake images, researchers used Least Squares GAN (LSGAN) with discriminator-1 loss value is 0.0061, discriminator-2 loss value is 0.0036, and generator loss value is 0.575. With the small loss value of the three important components, discriminator accuracy value in terms of classification reaches 95% for original image and 99% for fake image. In classified original image and fake image in this study using a supervised contrastive loss classification model with an accuracy value of 99.93%.

Corresponding Author:

Jasman Pardede,
Department of Informatics, Faculty Industry, Institut Teknologi Nasional Bandung
Jl. P.H.H Mustafa No. 23 Bandung, Indonesia
Email: jasmanpardede78@gmail.com

1. INTRODUCTION

Recently, numerous violations utilize innovation to produce someone's confront, which incorporates a terrible impact on that individual. The system can manipulate face, but not a few of these novelties are used for positive things, such as making fake face to perform scenarios or actions that have a negative impact. The produce of a system that can detect fake or real faces using the GAN algorithm as a reference in producing fake images, it will be able to minimize the occurrence of deception or spreading false information on the visuals of someone whose face is generated [1]. Before build system that can classify image fake or not, we need to make fake images that can be use as input to the classification model, so the model can learn and create some pattern that will be used for prediction

GAN can build a fake image with a process where the generator will try to create a fake image as input to the discriminator. If the discriminator still recognizes a generated image as fake, it will be returned to the generator network. The generator must be trained to be able to create a set of fake images that the discriminator cannot see as fake. GAN in this study will be used to generate fake images then will be classified using supervised contrastive loss.

There are several architectures of GAN to generate fake image. The previous researches on generated fake images are CartoonGAN used for creating fake images as cartoon image [2], StackGAN version 1 and StackGAN version 2 to generated realistic image synthesis [3], StarGAN for multidomain image-to-image translation [4], GAN-DeepFD for generate fake image in the wild with best accuracy score is 94,7% [5], CycleGAN [6], and Least Squared GAN (LSGAN) used for detecting fake image into the wild with accuracy score 98,1% [7]. In this research, LSGAN selected for generate fake face image because the decision boundary of LSGAN can decrease training time of generator to update its model. Even though implementation of LSGAN in previous research [5] and [7] could produce best accuracy score among other architectures of GAN. This research aims to generate fake face images using LSGAN and classify fake face images and original face images using supervised contrastive loss. The parameter that becomes the focus is the loss value of discriminator and generator. The smaller loss value will generate a better fake image in visual view.

2. METHOD

This research uses LSGAN architecture as a model to generate fake image with several stages of model process and supervised contrastive loss to classify whether input image is included in the original image or fake image. Main flowchart to describe work of overall system performance show in Figure 1.

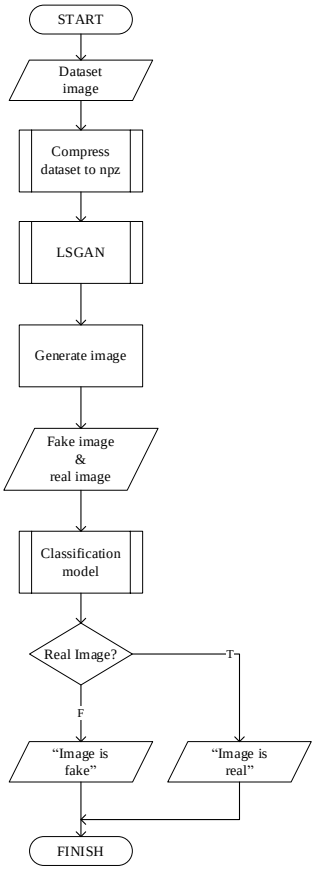


Figure 1. Flowchart System

2.1. Preparation

In this research using CelebA [5] dataset with total 202.599 face images because there are so many possible patterns to be formed using this dataset because it consists of thousands different celebrity faces, this can produce a generator model with many possible generated image results, but we use 70.000 images as input for training model. The reason for using only 70000 images as training input is because the research tools can only train 70000 images from 202,595 images from the source dataset. Because the number of datasets is too large, it is necessary to decrease the size of them. System got coordinate of face in every image on dataset using MTCNN (Multi Convolutional Neural Network) [8]. Then, every pixel of image will be converted into *numpy* array and dataset saved as *npz* file.

2.2. LSGAN (Least Squared Generative Adversarial Network)

LSGAN consists of two parts, namely generator and discriminator. The basic concept of GAN explains that generator is a neural network that will be trained to produce best possible fake image, while discriminator functions to classify generator's fake image comes from real data or not. The generator must be able to fool discriminator by producing fake image that is very similar with original image. The discriminator must be making a decision of boundaries to separate the real image and fake image, the generator is responsible for making new images that look like real. Discriminator of LSGAN is implemented using MSE (Mean Squared Error) functions such as L2 Loss [9]. The formula of LSGAN is defined as follow:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}E_{x\sim p_{data}(x)}[(D(x)-b)^2] + \frac{1}{2}E_{z\sim p_{data}(z)}[(D(G(z))-a)^2]$$

where: a is fake label, b is real label, and c is value of G that wants to make D believe with fake data.

LSGAN built with neural network architecture consisting of several convolution blocks and other processes on generator and discriminator. LSGAN architecture can be seen in Figure 2:

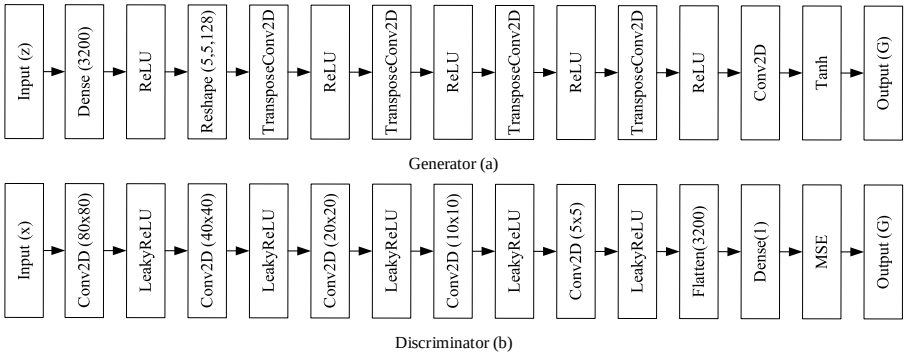


Figure 2. (a) Generator architecture, (b) Discriminator architecture

Step by step of creating LSGAN model and get the loss value of the generator and discriminator for the generated image process can be seen in Figure 3.

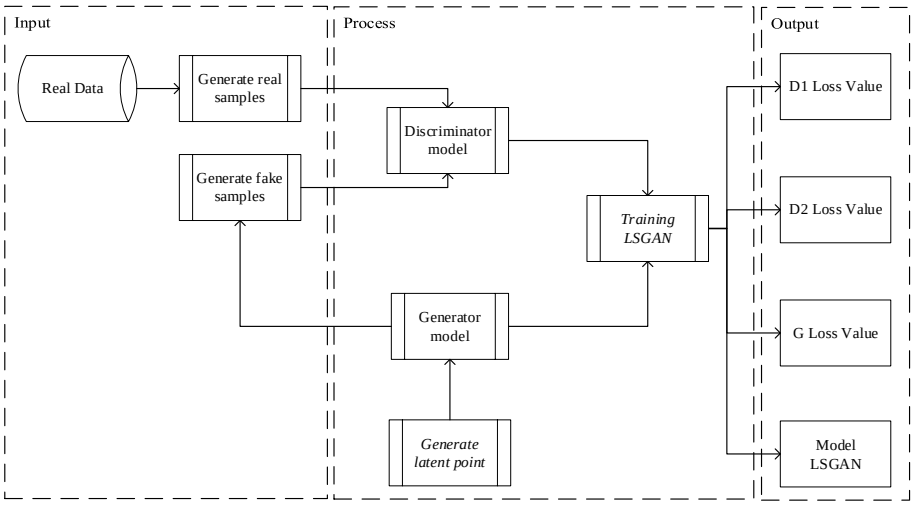


Figure 3. Block Diagram

Explanation for the block diagram in Figure 3 is as follows:

1) Load real data
   Create a dataset with scale corresponds range of [-1, 1] to match image output by generator model.
2) Generate real samples

After defining the dataset, then taking samples from the dataset as real data.

3) Generate latent points
   The next stage is creating a function that can be used as input to generator model which is generate random points in the latent space (latent point).
4) Generator model
   Generator task is generating a fake image from a collection of datasets and random noise (latent point). Dimension of latent point is 1 x 100, so that neural network will focus on the up-sampling stage until dimensions are same as expected output (80,80,3).
5) Discriminator model
   Discriminator task is discriminate generated image from the original image in the dataset. Discriminators are trained with two data, there were original data and fake data generated by generator, using the principle of down sampling until flatten process [10]. Flatten process is to make a pixel image from several arrays into a 1-dimensional array for classification purposes [11]. In the development of this model, the final stage will use linear activation function and loss function using Mean Squared Error (MSE).
6) Training LSGAN
   The weights of model are updated if result of image was real, allowing generator to be updated so that it can produce fake image similar as realistic data. The function used to define and compile the model to update generator via discriminator using MSE (Mean Squared Error). The final stage is training with a half batch of real and fake samples to make an update of one batch's worth of weights to the discriminator.
7) Outputs in this system are discriminator-1 loss value, discriminator-2 loss value, generator loss value, and LSGAN Model (generator).

2.3. Contrastive Loss

Contrastive loss is a calculation based on matrix to calculate distance during training with distance-based predictions. The purpose of contrastive loss is to learn vector of input image that representation in the same class will be more similar than the image representation in different classes [12].

$$contrastive\ loss = Y_{true} * D^2 + (1 - Y_{true}) * (margin - D, 0)$$

where Y_{true} is decision and D is output from prediction distance using *Euclidean distance*

3. RESULTS AND DISCUSSION

3.1. Generate Fake Face Image

Latent point used as input into generator to create a fake image. Sample value was number of samples randomly to be used as a pattern in generator to create a fake image for one iteration. This study used four variations of image sample values, there were 100, 200, 300, 400 of the 70.000 images used as datasets. As for the variation of latent point dimension value, there are three variations, there were 100, 200, 300, 400 which will be combined with the sample value so that there are four combinations which can be seen from the table below:

Table 1. Combination of Latent Points and Samples

No	Latent Point	Samples
1	100	100
2	200	200
3	300	300
4	400	400

Every combination is tested with 20 iterations to see best performance based on loss value of each discriminator and generator. Loss of discriminator shows the ability of the discriminator to detect generated image. The smaller loss value of discriminator would give better performance of discriminator compared to the generator. The smaller loss value of generator showed generator's ability

p-ISSN: 2528-1682
e-ISSN: 2527-9165

to fool discriminator in ordering between real image or fake image. Based on testing process, latent point = 200 and samples = 200 was the best combination with loss score as shown in Tabel 2.

Table 2. Latent Point = 200 and Samples = 200

Epoch	Diskriminator-1 Loss	Diskriminator-2 Loss	Generator Loss
1	0,098	0,033	1,055
2	0,007	0,006	0,014
3	0	0,005	0,013
4	0	0,003	0,001
5	0,001	0,003	0
6	0,001	0,002	0,003
7	0,001	0,004	0,009
8	0,002	0,002	0,01
9	0,002	0,003	0,008
10	0,002	0,002	0,008
11	0,002	0,003	0,006
12	0,001	0,002	0,005
13	0,001	0,001	0,004
14	0,001	0,001	0,004
15	0,001	0,001	0,001
16	0	0	0
17	0	0	0,001
18	0,001	0	0,002
19	0	0	0,003
20	0,001	0,001	0,003

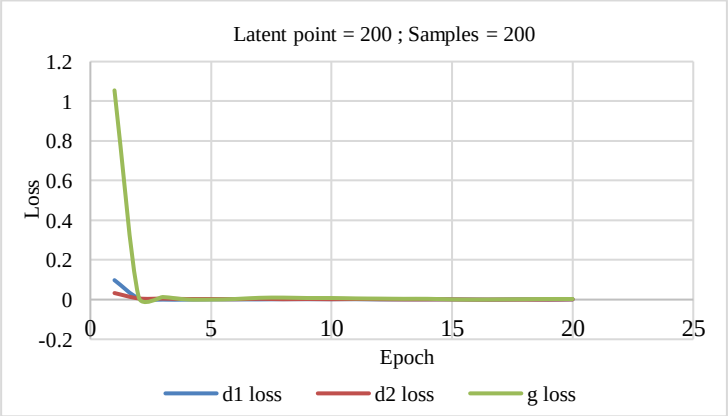
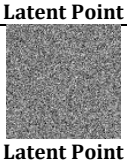


Figure 4. Latent Point = 200 and Samples = 200 (chart)

Based on testing latent point value = 200 and sample value = 200, loss value of discriminator and generator with an average of discriminator-1 loss : 0.0061 ; discriminator-2 loss : 0.0036 ; generator loss: 0.0575. The discriminator's ability to recognize fake images is still more better than original image. This is evidenced by the loss value discriminator-2 is smaller than the loss discriminator-1. Generator produces a slightly larger loss compared to discriminator, then the generator's ability to manipulate the discriminator when the classification is quite maximal because of the difference in the loss value discriminator-2 with a relatively small generator that is 0.054. The visualized discriminator and generator using a graph can be seen in Figure 4.

To get maximum results in training process using 100 iterations with a real classification accuracy value equals to 95% and fake equals to 99%. In Table 3. shows result of generated image with latent point 200 and samples 200

Table 3. Generated Image

Latent Point	Generated Image
	
Latent Point	Generated Image

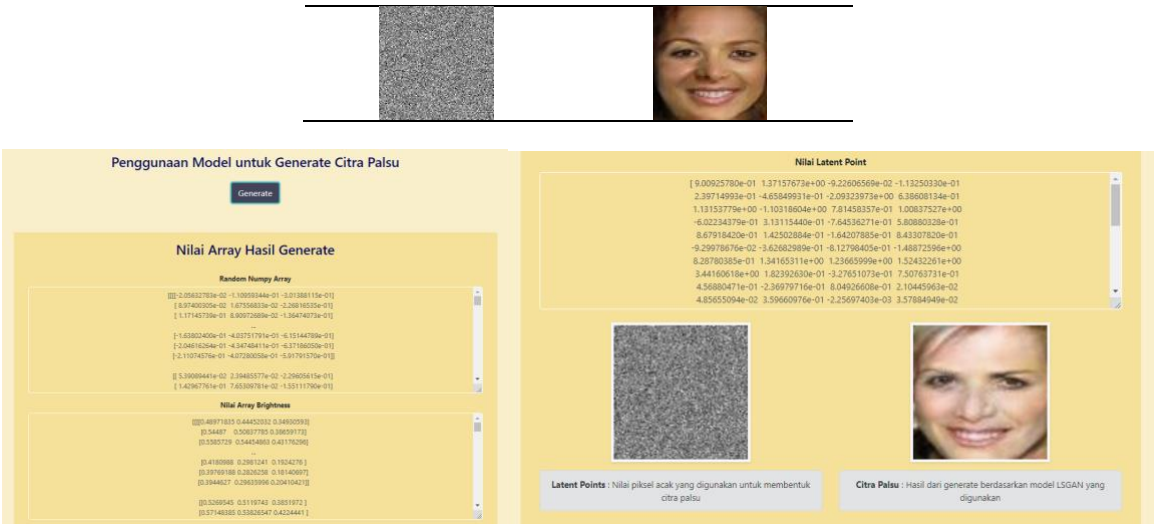


Figure 5. User Interface for generate Fake Face Image

Models trained three times, namely training for models with female-male faces dataset, models with female faces dataset, and models with male faces dataset. The results of three models are as Tabel 4. shows:

Table 4. Generated Image in three models

Model	Generated Image
Female-Male Faces Dataset	
Female Faces Dataset	
Male Faces Dataset	

Every pattern or feature in pixels of image made model learn and create generated image suitable with input in training process. Model with female-male faces dataset generated two possible image, female or male faces. Model with female faces dataset generated image similar with face of female. Otherwise, model with male faces dataset generated image similar with face of male.

3.2. Classification

The results of model using supervised contrastive loss produced an accuracy score 99.93%. In testing step, this classification model uses 10 randomly selected images outside of the training data, five for fake images and five for real data. It is important to note that the results of these tests do not contribute to the training or previous testing processes of the model. These test results are solely used as an additional evaluation of the existing model and as a tool to assess the reliability and effectiveness of the developed GUI. In this context, the testing of these ten images serves as an additional step to ensure the overall performance and functionality of the model and the user interface that has been created. The results of the test are shown in Table 5.

Table 5. Combination of Latent Points and Samples

Image	Expectation	Observation	Accuracy Prediction
	Fake Image	Fake Image	0.9997

	Fake Image	Fake Image	0.9928
	Fake Image	Fake Image	0.7502
	Fake Image	Fake Image	0.9919
	Fake Image	Fake Image	0.9999
	Real Image	Real Image	1.0
	Real Image	Real Image	1.0
	Real Image	Real Image	1.0
	Real Image	Real Image	1.0
	Real Image	Real Image	1.0

Based on testing process on ten samples, classification model had classified fake image and real image without wrong prediction. Highest accuracy prediction equals to 100% and smallest is 75,2 %. This testing process is to prove how good the results of the artificial image produced by the LSGAN model are. With the misclassification of the fake image, it can be proven that the image produced by the LSGAN model is difficult to distinguish from the original image, meaning that the fake image has close similarities to original image. The fake image generated by the LSGAN model can be said to be good because it can outwit the supervised contrastive loss model in classifying images.

4. CONCLUSION

Generate a fake image using LSGAN with a latent point value of 200 and a training sample value of 200 resulting in a loss of the first discriminator with an average of 0.0061; discriminator-2 loss: 0.0036 and generator loss: 0.0575. The smaller the loss value will affect the accuracy of the discriminator in distinguishing the original image and the fake image. The relationship between them is inversely so that a small loss value will produce a large accuracy. After generate fake image is classification using a supervised contrastive loss model. This model can produce an accuracy value is 99.93%. The shortcoming of this research is that the system can only randomly generate fake faces and cannot identify which fake faces are the same as real faces, maybe this can be a future work so that system can generate fake faces and recognize these faces according to real image and can calculate fake percentage of the generating image.

REFERENCES

[1] S. Mahdizadehaghdam, A. Panahi, and H. Krim, "Sparse generative adversarial network," *Proc. - 2019 Int. Conf. Comput. Vis. Work. ICCVW 2019*, pp. 3063–3071, 2019, doi: 10.1109/ICCVW.2019.00369.

[2] Y. Chen, Y. K. Lai, and Y. J. Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9465–9474, 2018, doi: 10.1109/CVPR.2018.00986.

[3] H. Zhang *et al.*, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, 2019, doi: 10.1109/TPAMI.2018.2856256.

[4] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8789–8797, 2018, doi: 10.1109/CVPR.2018.00916.

[5] C. C. Hsu, C. Y. Lee, and Y. X. Zhuang, "Learning to detect fake face images in the wild," *Proc. - 2018 Int. Symp. Comput. Consum. Control. IS3C 2018*, pp. 388–391, 2019, doi: 10.1109/IS3C.2018.00104.

[6] L. Nataraj *et al.*, "Detecting GAN generated fake images using Co-occurrence matrices," *arXiv*, 2019.
[7] C. C. Hsu, Y. X. Zhuang, and C. Y. Lee, "Deep fake image detection based on pairwise learning," *Appl. Sci.*, vol. 10, no. 1, pp. 1–10, 2020, doi: 10.3390/app10010370.
[8] Y. Zhang, P. Lv, and X. Lu, "A Deep Learning Approach for Face Detection and Location on Highway," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 435, no. 1, 2018, doi: 10.1088/1757-899X/435/1/012004.
[9] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2813–2821, 2017, doi: 10.1109/ICCV.2017.304.
[10] H. Thanh-Tung, T. Tran, and S. Venkatesh, "Improving generalization and stability of generative adversarial networks," *arXiv*, pp. 1–18, 2019.
[11] M. Elgendy, "Human-in-the-Loop Machine Learning Version 1 MEAP Edition Manning Early Access Program Copyright 2019 Manning Publications," 2019.
[12] P. Khosla *et al.*, "Supervised Contrastive Learning," no. NeurIPS, pp. 1–23, 2020, [Online]. Available: http://arxiv.org/abs/2004.11362.