GENERATING SYNTHETIC TRAINING DATASETS USING CONDITIONAL GENERATIVE ADVERSARIAL NETWORK TO IMPROVE IMAGES SEGMENTATION

Iffati Uzma[1], Rani Nooraeni[2], Takdir[3]
[1]Badan Pusat Statistik Kabupaten Bungo, Jambi
[2]Politeknik Statistika STIS
[3]University of Tsukuba
e-mail: [1]iffati.uzma@bps.go.id, [2]raninoor@stis.ac.id, [3]takdir-kde@kde.cs.tsukuba.ac.jp

ABSTRACT

A limited amount of training datasets in *deep learning* research could impact the accuracy of the resulting models. This situation can cause *overfit*, so the model cannot work correctly. Conditional Generative Adversarial Network (CGAN) was introduced to generate synthetic data by considering certain conditions. This study aims to generate additional synthetic training datasets to improve the accuracy of the object segmentation model of images. Firstly, we evaluated CGAN-based dataset generator accuracy against several open datasets. Then, we applied the generator to train two object segmentation models, i.e., FCN and CNN U-Net. Our evaluation shows that CGAN can generate synthetic datasets well. Complex datasets require more training iterations. It also improves both segmentation models' validation loss and accuracy, although other metrics still need further improvement.

***Keywords*— *deep learning*,** synthetic data, cGAN, U-net, segmentation

ABSTRAK

Dataset pelatihan dalam jumlah terbatas dalam penelitian pembelajaran mendalam dapat memengaruhi keakuratan model yang dihasilkan. Situasi ini dapat menyebabkan overfit, sehingga model tidak dapat bekerja dengan baik. Conditional Generative Adversarial Network (CGAN) diperkenalkan untuk menghasilkan data sintetik dengan mempertimbangkan kondisi tertentu. Penelitian ini bertujuan untuk menghasilkan dataset training sintetik tambahan untuk meningkatkan akurasi model segmentasi objek citra. Pertama, kami mengevaluasi akurasi generator dataset berbasis CGAN terhadap beberapa dataset terbuka. Kemudian, kami menerapkan generator untuk melatih dua model segmentasi objek, yaitu FCN dan CNN U-Net. Evaluasi kami menunjukkan bahwa CGAN dapat menghasilkan dataset sintetik dengan baik. Kumpulan data yang kompleks memerlukan iterasi pelatihan yang lebih banyak. Ini juga meningkatkan kehilangan dan akurasi validasi kedua model segmentasi, meskipun metrik lainnya masih memerlukan peningkatan lebih lanjut.

Kata kunci— *deep learning*, data sintetik, cGAN, U-net, segmentasi

INTRODUCTION

Deep Learning Deep is a machine learning concept gaining popularity nowadays. It can automatically extract required features, an essential step in preparation for model training. However, deep learning requires many training datasets to perform well (Mahapatra, 2018; Papernot et al., 2016; Roh, Heo, & Whang, 2019).

The limitation of data is an essential issue in deep learning. Thousands to millions of labeled datasets for each category shall be required to train a model (Goodfellow et al., 2014). In fact, we are often faced with the problem of a need for more data (Antoniou, Storkey, & Edwards, 2017). In this situation, the resulting model tends to overfit, i.e., it cannot be generalized for the testing datasets and leads to an inaccurate model (Antoniou et al., 2017; Ministry of Defence, 2020).

A common solution to improve the accuracy and reduce the biases of a deep learning model is to add more datasets (Ministry of Defence, 2020; Zhu et al., 2020). Generating new synthetic datasets is one of many available solutions to enlarge dataset size (Nyberg, 2021). According to (Antoniou et al., 2017), an automatic data generator is used to add more training datasets to improve the performance of the corresponding classification model.

Image segmentation is a fundamental method in object-based image analysis (Hossain & Chen, 2019). It aims to recognize objects' existence or location by labeling each pixel in images into a class or category (Kattenborn, Leitloff, Schiefer, & Hinz, 2021). *Convolutional Neural Network (CNN)* is commonly used to segregate different objects semantically (Kulkarni, Mohandoss, Northrup, Mwebaze, & Alemohammad, 2020). However, CNN often fails to generalize accurate segmentation when processing satellite images from other areas or weather conditions (Rezaei et al., 2018). In practice, *Fully Convolutional Networks (FCN)* and *U-Net* architectures are the most used CNN architectures to perform semantic segmentations (Kattenborn et al., 2021).

Generative Adversarial Network (GAN) is one of the deep learning methods to generate synthetic data. GAN learns the distributions of the given data or images and creates new ones that have equivalent patterns and characteristics to the existing datasets (Nyberg, 2021).

This study aims to evaluate CGAN accuracy in generating synthetic datasets. We also assess the accuracy of using CGAN-generated datasets to train CNN-based image segmentation models.

OBJECTIVE

This study was conducted with the following objectives:

1. To apply and evaluate CGAN to generate synthetic data from publicly available image datasets.
2. To evaluate the accuracy of image segmentation models trained with additional CGAN-generated synthetic datasets.

RELATED WORK

A sufficient amount of datasets is a common problem in deep learning research. Many studies have been done to overcome this issue. The study in (Roh et al., 2019) published a literature review to evaluate methods in data acquisition, labeling, and increasing data availability, and each was provided with approaches. One of the mentioned methods to improve data availability is automatic synthetic data generation using GAN.

Many studies related to data generation using GAN also have been conducted. In 2018, Yu, Song, & Lu reported that CGAN reduces data-generating costs and increases robustness compared to other methods. Heilemann et al., 2022 have also conducted a study using CGAN and U-Net related to data unavailability. Their study found that increasing the number of training datasets improves object segmentation accuracy.

The studies above indicate that CGAN is a potential candidate for the object segmentation model. Rezaei et al., 2018) applied CGAN for the semantic segmentation of brain tumours. In 2018, Frangi et al. also conducted another medical imaging using CGAN by segmenting mammogram images.

The result showed that CGAN significantly improved segmentation accuracy.

METHODOLOGY

A. *Data Collection*

We use open image datasets, i.e., MNIST-fashion, MNIST-digit, CIFAR-10, and Oxford-IIIT Pet, in this study. The first three datasets are available in the Keras TensorFlow library. MNIST-fashion contains cloth pictures of 10 categories: shirts, trousers, sweaters, dresses, coats, sandals, suits, shoes, bags, and ankle boots. Each category contains 6000 pictures. MNIST-digit is a collection of digit images (‘0’ to ‘9’ characters). CIFAR-10 includes ten categories of pictures: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each category contains 6000 pictures.

Figures 1, 2, and 3 are sample images from each category.

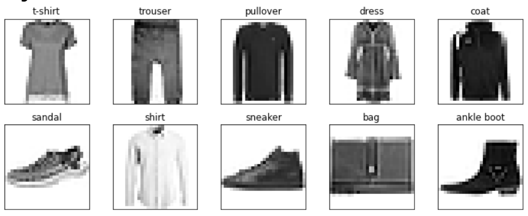


Figure 1. MNIST-Fashion sample images

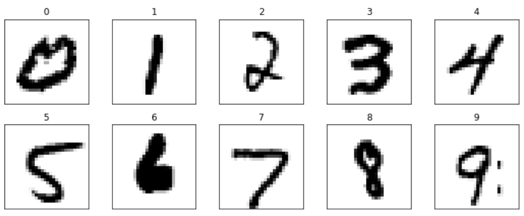


Figure 2. MNIST-Digit sample images

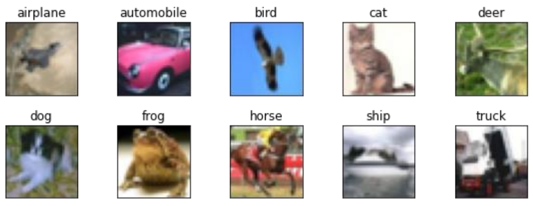


Figure 3. CIFAR-10 sample images

Oxford-IIIT Pet dataset is fetched from Kaggle. It contains 37 pet categories, each with 200 pictures with various scales, layouts, and lighting. We choose 10 out of 37 categories, as shown in Figure 4. Besides synthetic data generation, the Oxford-IIIT Pet dataset was also used to evaluate the segmentation models.

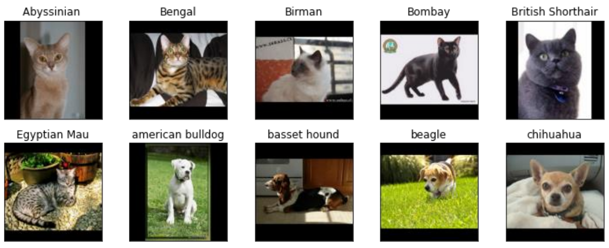


Figure 4. Oxford III-T Pet sample images

B. *CGAN Model*

GAN is formed from *generator* and *discriminator* architectures. The generator identifies the distribution of data, whereas the discriminator estimates the possibility of whether a sample originated from real datasets or generators. Conditional GAN (CGAN) is an extended version with an additional feature *y*, e.g., data label, in both generator and discriminator. We implemented it by adding *y* to the generator and discriminator as a new input layer. Figure 5 depicts the CGAN model training process.

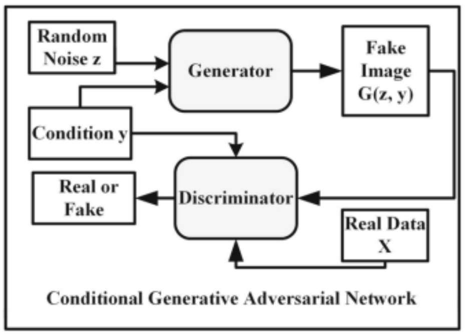


Figure 5. Conditional GAN [17]

The *loss function* of CGAN is formulated as:

$$L_{cGAN} = \min_{G}\max_{D} V_{CGAN}(D,G) = E_{x\sim p_{data}(x)}[logD(y)] + E_{z\sim p_z(z)}\left[log\left(1 - D\big(G(y)\big)\right)\right] \quad ...(1)$$

$V(D,G)$: is the value function of CGAN
y: condition
D: discriminator
G: generator
x: original data item
$p_{data}(x)$: data distribution of x
z: random noise vector as the inputs to generator
$p_z(z)$: prior noise distribution

$D(y)$ means discriminator takes two inputs x and y to be used with discriminator function. D maximizes the accuracy of the classification of samples in detecting whether they are original or generated. $G(y)$ means the generator produces images based on a condition y. G only influence $1 - D(G(z|y)$, which is the probability that D recognizes generated sample, by minimizing its value (Cheng, Tahir, Eric, & Li, 2020; Mirza & Osindero, 2014; Nyberg, 2021).

A CGAN model is constructed as follows (Brownlee, 2019):

- Load dataset
- Define generator and discriminator functions
  - Generator:
    - Input: *point* in latent space
    - Output: images
    - Uses inverse of convolution (transposed convolution)
    - Activation function: LeakyReLU
    - Optimizer: Adam Optimizer
    - Activation function in the output layer: tanh
  - Discriminator
    - Input: images dan class labels
    - Output: binary (Original=1, Generated=0)
    - Implemented using Convolutional Neural Network
    - Activation function: LeakyReLU
    - Optimizer: Adam optimizer
- Define CGAN model
  - Combination of the generator and discriminator
  - Input: point in *latent space*, uses the generator to produce images. The resulting images are the inputs of the discriminator.
  - Output: a classification, i.e., original or generated
- Define a function to select random samples from the dataset for each update to the CGAN model
- Define input function for the generator from *latent space*
- Perform CGAN training process with predefined parameters.

Generating synthetic datasets using CGAN

CGAN model was applied to the MNIST-fashion, MNIST-digit, and CIFAR-10 datasets to evaluate its ability and accuracy in generating synthetic datasets before utilizing it to train object segmentation. This experiment also can show the characteristics of results among different datasets.

The results are evaluated using plot graphs of training loss and training accuracy. CGAN has two loss functions, each in generator and discriminator, respectively. There are two loss values in the discriminator, i.e., loss in detecting that the evaluated data is *original* and loss in detecting that the evaluated data is *generated*.

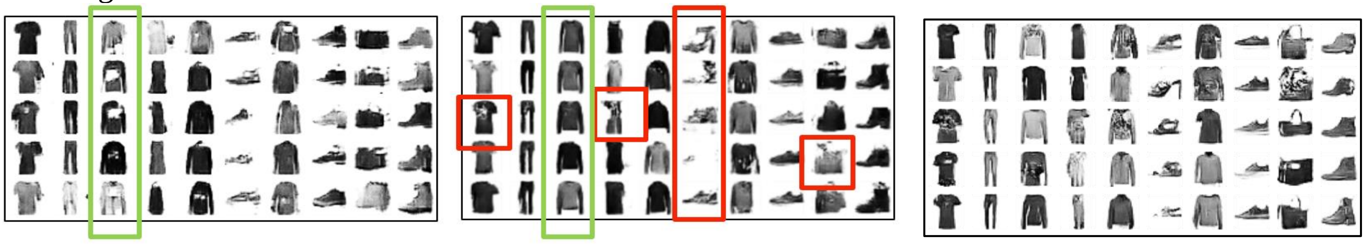
Image Segmentation

We performed image segmentation on the Oxford IIIT Pet dataset. Before segmentation, additional training data is generated using CGAN. We use FCN and CNN U-Net as segmentation models and compare the results with and without synthetic training data on both segmentation models.

The segmentation models are tested and evaluated against the ground truth by collecting the metrics of accuracy, training loss, validation loss, Intersection over Union (IoU), and Dice Score.

RESULT AND DISCUSSION

A. *Visual Evaluation of Synthetic Data Generation*



(a) (b) (c)

Figure 6. Sample of generated synthetic MNIST-Fashion data using (a) epoch=10, (b) epoch=20, dan (c) epoch=40

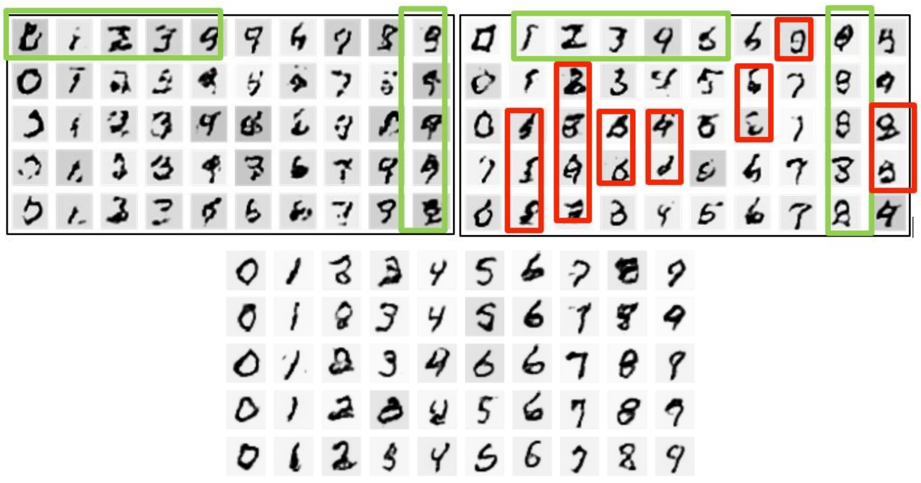

(a) (b) (c)

Figure 7. Sample of generated synthetic MNIST-Digit data using (a) epoch=10, (b) epoch=20, dan (c) epoch=40

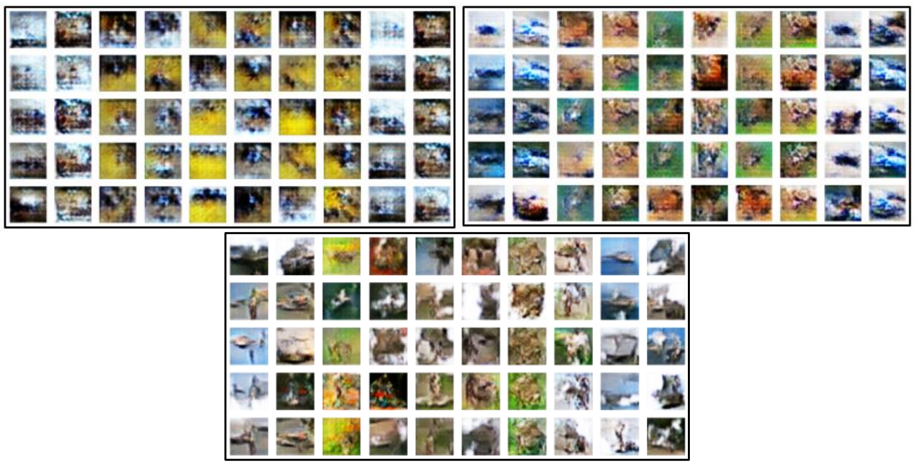

(a) (b) (c)

Figure 8. Sample of generated synthetic CIFAR-10 data using (a) epoch=10, (b) epoch=20, dan (c) epoch=40

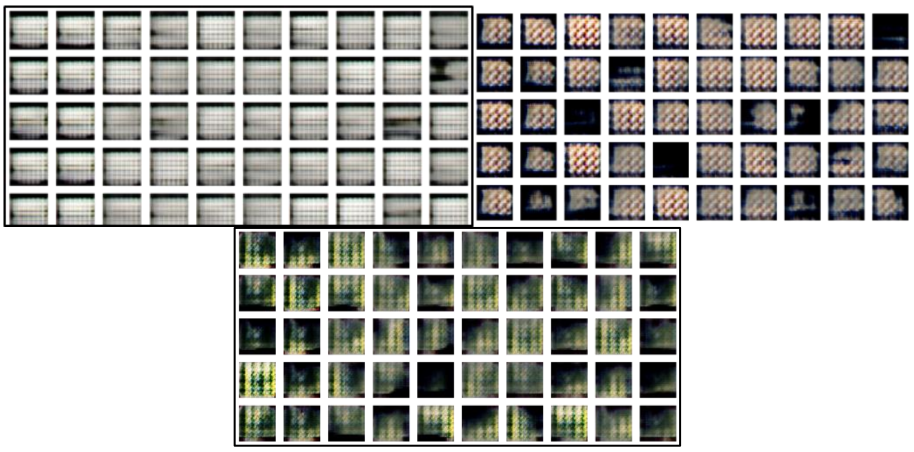

(a) (b) (c)

Figure 9. Sample of generated synthetic Oxford IIIT Pet data using (a) epoch=10, (b) epoch=20, dan (c) epoch=40

We used 6000 images as training data from MNIST-Fashion and MNIST-Digit, respectively. There are 5000 images from CIFAR-10 dataset. In the Oxford III dataset, there are 10 categories, and we took 200 images from each category.

MNIST-Fashion generated data, as visually shown in Figure 6, has a good similarity with the original. Increasing the epoch from 10 to 20 improves the quality of generated images, especially in the green rectangle-marked images. These images are transformed from images containing incomplete parts to complete images. However, incomplete prints, such as in the red rectangle, remain found. It means that the model can generate synthetic images properly with the configured parameters. Still, more training iterations are needed to improve the quality, as shown in the results of 40 epochs.

A similar process on MNIST-Digit datasets showed equivalent results, as depicted in Figure 7. In the ten epochs, '8' characters offer the best quality. Meanwhile, most of the '1' and '2' characters cannot be seen clearly. Increasing the epoch to 40 improves the whole image quality.

In the CIFAR-10 dataset, generated images in 10 epochs are still blurry, and their class cannot be recognized visually, as shown in Figure 8. Objects can be recognized well if the epoch is increased to 40. The CIFAR-10 dataset contains color images instead of black and white in the former datasets. Therefore, features in this kind of image are more variable and require more training iterations as well.

The lowest quality of generated images is shown in the Oxford IIIT Pet dataset. This dataset has the highest resolution and various scales and lighting. Training the generator until 40 epochs is insufficient to produce good images, as shown in Figure 9.

Generally, increasing the number of training iterations improves generated image quality, as demonstrated by the above experimental results. More iterations are required for richer image features, such as color, resolution, and brightness. Considering this, we increased the number of iterations to 100 for CIFAR-10 and Oxford IIIT Pet datasets. The sample results are presented in Figures 10 and 11, respectively.

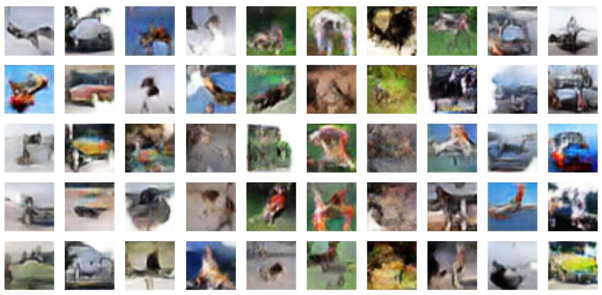


Figure 10. Sample of generated synthetic CIFAR-10 data using epoch=100

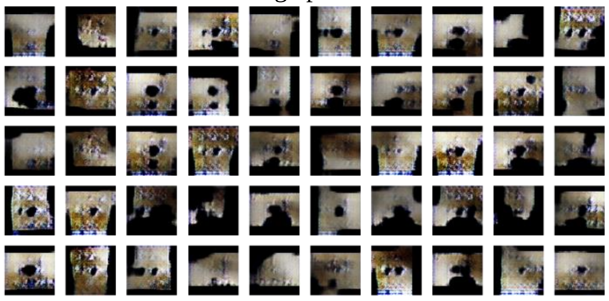
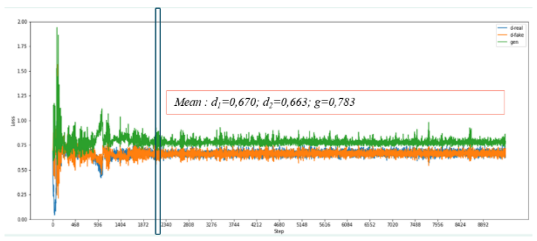


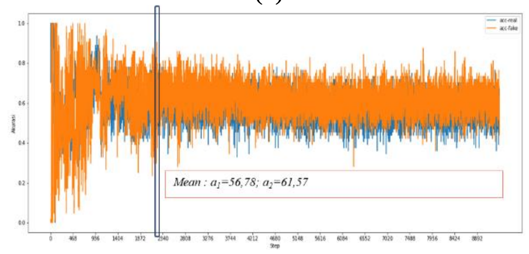
Figure 11. Sample of generated synthetic Oxford IIIT Pet data using epoch=100

The resulting images become clearer visually. Both datasets have various colors and backgrounds, meaning more features that need to be accommodated. This led to more iterations, i.e., longer training times, to achieve representative outputs.

B. *Evaluation Metrics of Synthetic Data Generation*



(a)

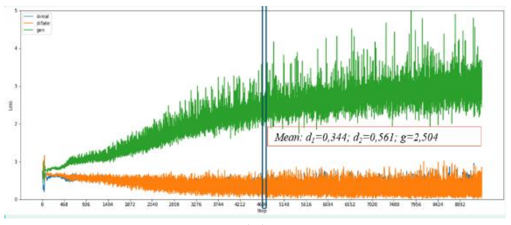


(b)

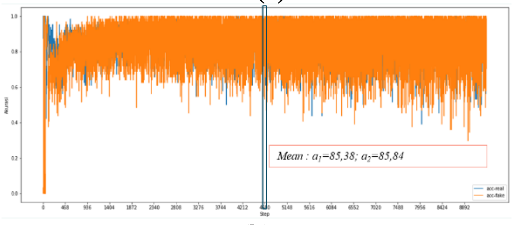
Figure 12. Evaluation metrics of CGAN on MNIST-Fashion dataset: (a) training loss, and (b) discriminator accuracy

When applying CGAN on the MNIST-Fashion dataset, the generator and discriminator were stable at iteration 2000

(5th epoch), as shown in the line graph in figure 12 (a). Loss values of the original data discriminator, generated data discriminator, and generator at that point are 0.670, 0.663, 0.783, consecutively. Figure 12 (b) depicts the accuracy of the discriminator in identifying whether data is original or generated. In line with the loss values, the accuracy of either original or generated data identifications becomes stable at iteration 2340 (5th epoch). In this stable condition, the accuracy of the original and that of generated data identifications are 56.78% and 61.56% on average, respectively.



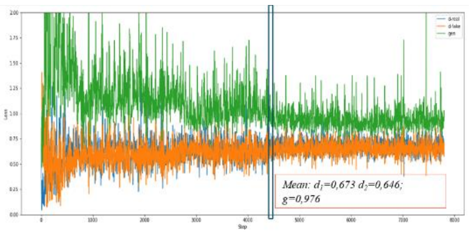
(a)



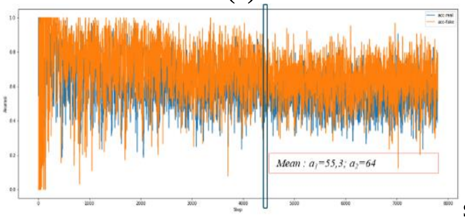
(b)

Figure 13. Evaluation metrics of CGAN on MNIST-Digit: (a) training loss, and (b) discriminator accuracy

In MNIST-Digit dataset generation, as shown in figure 13, loss values of the generator are gradually increased. The averages of loss values for the original data discriminator, generated data discriminator, and generator in the last ten epochs are 0.344, 0.561, and 2.504, respectively. The discriminator accuracy graph, i.e., figure 13 (b), shows great averages of both original and generated data which are 85.37% and 85.83% in the last ten epochs. It means that the discriminator model can classify original and generated data well.



(a)

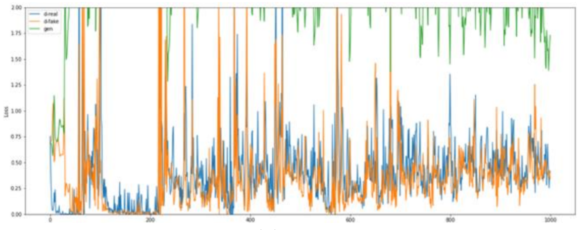


ss

(b)

Figure 14. Evaluation metrics of CGAN on CIFAR-10: training loss, and (b) discriminator accuracy

Both graphs in Figure 14 show fluctuations in all evaluation metrics of CIFAR-10 dataset compared to the former datasets.


(a)

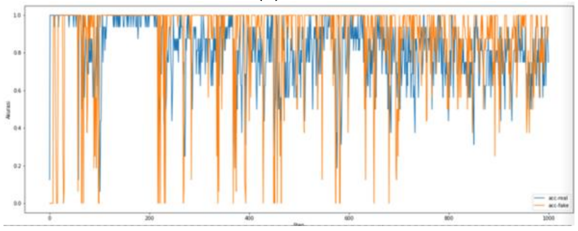

(b)

Figure 15. Evaluation metrics of CGAN on Oxford IIIT Pet dataset: (a) training loss, and (b) discriminator accuracy

Oxford IIIT Pet metrics in Figure 15 show unstable values. In the beginning, the loss values of the generator were high and decreased gradually. It means that more training iterations are required to stabilize the metrics.

There is a general pattern from all the above experimental results of loss and accuracy from four different datasets, i.e., the discriminator and generator are opposite. Their loss values would lead to a stable state if iterations were increased. An optimal GAN model can be achieved when the loss values are stable and convergent, i.e., the model cannot increase or decrease its loss value.

TABEL I
SUMMARY OF CGAN EVALUATION METRICS

Indikator	*Epoch*	*MNIST-fashion*	*MNIST-digit*	*CIFAR-10*	*Oxford IIIT Pet*
Discriminator Loss (Original)	10	0,670	0,368	0,636	0,489
	20	0,672	0,341	0,685	0,488
	40	0,681	0,319	0,70	0,526
Discriminator Loss (Generated)	10	0,665	0,365	0,608	0,379
	20	0,666	0,339	0,663	0,431
	40	0,675	0,302	0,695	0,420
Generator Loss	10	0,779	2,087	1,125	3,091
	20	0,782	2,669	0,917	1,827
	40	0,770	3,22	0,757	1,969
Accuracy(Original)	10	57,56	85,69	59,43	77,52
	20	56,53	84,87	52,87	73,5
	40	54,18	84,7	47,04	70,82
Accuracy (Generated)	10	61,21	87,047	69,648	86,92
	20	60,28	85,56	60,956	89,12
	40	57,70	87,02	52,88	83,04

Table 1 above shows that from all data, the values in the MNIST-fashion data tend to be the most stable. MNIST-digit data is different from other data. In the MNIST digit, the value of the loss discriminator tends to decrease, while the value of the loss generator tends to increase with a fairly high value. However, for the value of discriminator accuracy, it can be seen that the three data seem to have succeeded in reducing the value of discriminator accuracy along with the ongoing training process, meaning that the generator carries out a learning process to make it difficult for the discriminator to classify the original and generated images. Then for the CIFAR-10 data, it can be seen that the loss value and discriminator accuracy are stable. However, the generator needs to produce a better image in the previous visual evaluation. It means that the value that appears in the discriminator loss or accuracy cannot be used to evaluate the resulting image but must be visually checked to determine whether the resulting image is as desired and whether training is still needed. It is proven that when the number of iterations in training is increased again on CIFAR-10 data, the resulting image generator gets better. The resulting values appear unstable for the Oxford IIIT pet data, meaning more training is needed.

C. *Segmentation*

Segmentation without Additional Data

Further segmentation process is performed on Oxford IIIT pet data. We used 920 images in the training phase. The original data is shown in the following Figure 17.

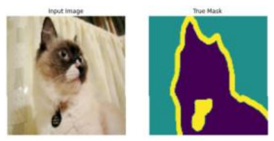
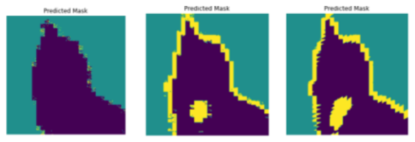


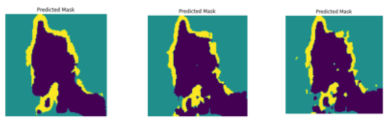
Figure 17. The original image to be predicted

The segmentation images generated during the training process, i.e. at each of the 5th, 10th, and 20th epochs are depicted in Figures 18 and 19 below.



(a) (b) (c)

Figure 18. FCN segmentation results without additional data for (a) epoch=5, (b) epoch=10, and (c) epoch=20



(a) (b) (c)

Figure 19. U-Net segmentation results without additional data for (a) epoch=5, (b) epoch=10, and (c) epoch=20

The pictures in Figure 17-19 show that the training process can perform better segmentation and resemble the actual segmentation along with the training process. Visually, with epoch 20 above, it can be obtained that the segmentation results on the U-Net method are better than the FCN method.

Segmentation with Additional Data

Before segmentation, the Oxford IIIT Pet dataset is used to train the conditional-GAN model to generate synthetic data. Figure 20 shows the results of the data training process on cGAN after training with epoch=1000:

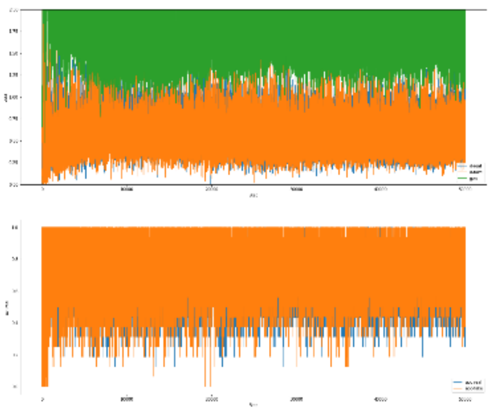


Figure 20. Graph of cGAN Training Loss and Accuracy on the Oxford IIIT Pet Dataset

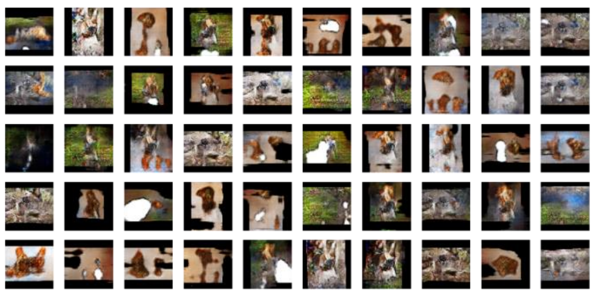


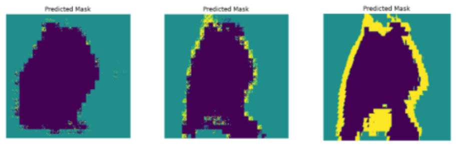
Figure 21. Oxford IIIT cGAN Revival Data Pet Dataset Epoch=1000

TABLE II
EVALUATION OF REGIONAL DATA FOR THE OXFORD IIIT PET DATASET

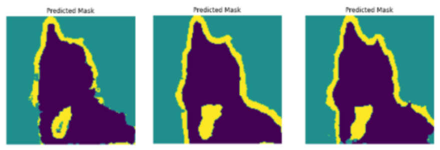
Indicator	*Epoch 10*	*Epoch 20*	*Epoch 40*	*Epoch 100*	*Epoch 1000*
FID	381,152	355,103	332,625	221,914	205,193
IS	1,000817 ±0,0000685	1,00148±0,00009	1,001696±0,000111	1,008715± 0,00101	1,00701± 0,00059

Figure 20 shows that the resulting values have not converged and have a wide range. Most animal images (cats and dogs) do not resemble the desired shape, as shown in Figure 21. The evaluation in Table 2 is also in line with those results. In Table 2, the FID value up to epoch=1000 is 205.193. This value is high, meaning the actual and generated images differ. The resulting Inception Score (IS) value is small, meaning the generated data is homogenous and lacks quality. It is likely because the datasets used have varied in size, color, and lighting, known as complex datasets. The complexity of the dataset in this paper means the complexity of image attribute compositions contained in an image/picture, such as lighting, coloring, and resolution. Each attribute contributes to the computations required to process the images. For instance, higher-resolution images require more computational resources to process that the lower-resolution images. Another example is that rich-colored images are more expensive to process than black-and-white images.

Next, 1000 synthetic data are generated using the cGAN model that has been trained. This data is then fed into segmentation training. The segmentation results are shown in Figures 22 and 23 below.



(a) (b) (c)
Figure 22. FCN segmentation results with additional data for (a) epoch=5, (b) epoch=10, and (c) epoch=20



(a) (b) (c)
Figure 23. U-Net segmentation results with additional data for (a) epoch=5, (b) epoch=10, and (c) epoch=20

D. *Comparison of Segmentation between with and without Synthetic Data*

The difference in segmentation results between the two conditions with and without adding synthetic data can be seen in Table 3 and Table 4 below:

TABLE III
COMPARISON OF SEGMENTATION WITH AND WITHOUT ADDITIONAL DATA USING THE FCN MODEL

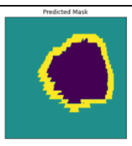
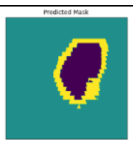
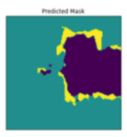
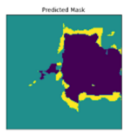
Differentiator	*Without Additional Data*			*With Additional Data*		
Segmentation image						
Loss training	0.2592	0.1751	0.1300	0.2896	0.2110	0.1330
Training accuracy	0.8678	0.9186	0.9396	0.8607	0.9003	0.9394
Loss validation	0.5162	0.7152	0.9647	0.4624	0.5807	0.7962
Validation accuracy	0.8153	0.8156	0.8171	0.8198	0.8281	0.8272
IoU	0,53066			0,50158		
Dice score	0,68536			0,6564		

TABLE IV
COMPARISON OF SEGMENTATION WITH AND WITHOUT ADDITIONAL DATA USING THE U-NET MODEL

Differentiator	*Without Additional Data*			*With Additional Data*		
Segmentation image						
Loss training	0,1121	0,0580	0,0311	0.1646	0.0933	0.0511
Training accuracy	0,9489	0,9687	0,9758	0.9273	0.9557	0.9706
Loss validation	0,3945	0,5080	0,7235	0.3202	0.3898	0.6211
Validation accuracy	0,8665	0,8715	0,8672	0.8786	0.8806	0.8718
IoU	0,70878			0,68498		
Dice score	0,82656			0,80858		

From the two comparisons in Tables 3 and 4, the segmentation results on additional data have generally decreased. Similarly, in FCN and U-Net, the value of training loss increased, training accuracy decreased, and IoU and Dice Score decreased. Only the validation loss and validation accuracy show better results, as shown in the yellow highlights. It is because the generated data that was previously used lacks similarity to the original data. However, there is still an influence on the results obtained. Decreasing the validation loss value and increasing validation accuracy when data is added shows that the model is getting better at determining model parameters. Therefore, in the future, better data generation can be done to obtain better segmentation results.

CONCLUSIONS

In this study, the following conclusions are obtained:

1. The cGAN method can generate synthesis data well through training. More complex data requires training with more iterative processes.
2. Adding generated synthetic data with cGAN positively affects loss and validation accuracy, but vice versa for other evaluation indicators. It means there is an effect from the addition of synthetic data, even though not on all evaluation indicators. The insufficient similarity of generated synthetic images from the original images causes it.

REFERENCES

Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Brownlee, J. (2019). *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery.

Cheng, K., Tahir, R., Eric, L. K., & Li, M. (2020). An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Multimedia Tools and Applications, 79*, 13725–13752. Springer.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., et al. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

Heilemann, G., Matthewman, M., Kuess, P., Goldner, G., Widder, J., Georg, D., & Zimmermann, L. (2022). Can Generative Adversarial Networks help to overcome the limited data problem in segmentation? *Zeitschrift für Medizinische Physik*, *32*(3), 361–368. Elsevier.

Hossain, M. D., & Chen, D. (2019). Segmentation for Object-Based Image Analysis (OBIA): A review of algorithms and challenges from remote sensing perspective. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 115–134. Elsevier.

Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing*, *173*, 24–49. Elsevier.

Kulkarni, A., Mohandoss, T., Northrup, D., Mwebaze, E., & Alemohammad, H. (2020). Semantic segmentation of medium-resolution satellite imagery using conditional generative adversarial networks. *arXiv preprint arXiv:2012.03093*.

Mahapatra, S. (2018). Why Deep Learning over Traditional Machine Learning? | by Sambit Mahapatra | Towards Data Science. Retrieved November 26, 2021, from https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063

Ministry of Defence. (2020). *Machine Learning with Limited Data*. London: Defence Science and Technology Laboratory.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Nyberg, D. (2021). Exploring the Capabilities of Generative Adversarial Networks in Remote Sensing Applications. Master thesis.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. *2016 IEEE European symposium on security and privacy (EuroS&P)* (pp. 372–387). IEEE.

Rezaei, M., Harmuth, K., Gierke, W., Kellermeier, T., Fischer, M., Yang, H., & Meinel, C. (2018). A conditional adversarial network for semantic segmentation of brain tumor. *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3* (pp. 241–252). Springer.

Roh, Y., Heo, G., & Whang, S. E. (2019). A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1328–1347. IEEE.

Singh, V. K., Romani, S., Rashwan, H. A., Akram, F., Pandey, N., Sarker, M. M. K., Abdulwahab, S., et al. (2018). Conditional generative adversarial and convolutional networks for X-ray breast mass segmentation and shape classification. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11* (pp. 833–840). Springer.

Yu, P., Song, K., & Lu, J. (2018). Generating adversarial examples with conditional generative adversarial net. *2018 24th international conference on pattern recognition (ICPR)* (pp. 676–681). IEEE.

Zhu, D., Xia, S., Zhao, J., Zhou, Y., Jian, M., Niu, Q., Yao, R., et al. (2020). Diverse sample generation with multi-branch conditional generative adversarial network for remote sensing objects detection. *Neurocomputing*, *381*, 40–51. Elsevier.