Received 15 July 2022, accepted 17 August 2022, date of publication 31 August 2022, date of current version 12 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203065

RESEARCH ARTICLE

LiWGAN: A Light Method to Improve the Performance of Generative Adversarial Network

NURUL AMIRAH MASHUDI[1,2], (Graduate Student Member, IEEE), NORULHUSNA AHMAD[1], (Senior Member, IEEE), AND NORLIZA MOHD NOOR[1], (Senior Member, IEEE)

[1]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur Campus, Kuala Lumpur 54100, Malaysia
[2]Faculty of Information Technology, City University Malaysia, Petaling Jaya, Selangor 46100, Malaysia

Corresponding author: Norliza Mohd Noor (norliza@utm.my)

This work was supported in part by Universiti Teknologi Malaysia under Grant R.K130000.2656.18J05, and in part by the Ministry of Higher Education Malaysia.

ABSTRACT Generative adversarial networks (GANs) gained tremendous growth due to the potency and efficiency in producing realistic samples. This study proposes a light-weight GAN (LiWGAN) to learn non-image synthesis with minimum computational time for less power computing. Hence, the LiWGAN method enhanced a new skip-layer channel-wise excitation module (SLE) and a self-supervised discriminator design for non-synthesis performance using the facemask dataset. Facemask is one of the preventative strategies pioneered by the current COVID-19 pandemic. LiWGAN manipulates a non-image synthesis of facemasks that could be beneficial for some researchers to identify an individual using lower power devices, occlusion challenges for face recognition, and alleviate the accuracy challenges due to limited datasets. The study evaluates the performance of the processing time in terms of batch sizes and image resolutions using the facemask dataset. The Fréchet inception distance (FID) was also measured on the facemask images to evaluate the quality of the augmented image using LiWGAN. The findings for 3000 generated images showed a nearly similar FID score at 220.43 with significantly less processing time per iteration at 1.03s than StyleGAN at 219.97 FID score. One experiment was conducted using the CelebA dataset to compare with GL-GAN and DRAGAN, proving LiWGAN is appropriate for other datasets. The outcomes found LiWGAN performed better than GL-GAN and DRAGAN at 91.31 FID score with 3.50s processing time per iteration. Therefore, LiWGAN could aim to enhance the FID score to be near zero in the future with less processing time by using different datasets.

INDEX TERMS Non-image synthesis, self-supervised discriminator, data augmentation, deep learning, generative adversarial network.

I. INTRODUCTION

Generative adversarial networks (GANs) are among the most significant developments in the deep learning domain, especially for medical imaging applications [1]. GANs have received considerable interest in the computer vision community due to data generation capabilities without directly modeling the likelihood density function. GANs comprise two neural networks: a generator that collects data distribution and a discriminator that predicts sample composition from training data [2]. In addition, the discriminator differentiates between genuine and fake images. Generally, the generator aims to make the discriminator believe that the images are genuine [3]. GANs can be used to generate new data for a limited dataset, which can be challenging and time-consuming. GANs are one of the pre-processing processes before the training phase begins. Comparing and contrasting generative and discriminative algorithms can help better grasp how GANs work.

Deep neural networks perform an unparalleled direction with sufficient data on various tasks such as image detection and classification, voice recognition and synthesis, human

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

learning, reinforcement learning, and others [4] in the past decade. Data augmentation (DA) has proven compelling in image and video classification [5] to identify genuine and fake images and videos. Moreover, data augmentation can be a regularizer to avoid neural network overfitting and enhance the efficiency of imbalanced classes. Simple techniques, including cropping, rotating, changing the image color, and flipping input images, are helpful for data augmentation, which many researchers have performed. In addition, the data-augmentation methods aim to address significant problems. The key approaches are in the data warping group, which is a method to directly augment the input to the data space model [6].

The impressive results of GANs are not only in generating realistic samples [7], [8], [9] but also in object detection [10], [11], [12] and image translation [13]. There are two categories of generative models: conventional machine learning algorithms and deep learning algorithms. The simulated observations based on a probability density function (pdf) built using generative methods can yield large samples. However, several issues arise in generating realistic samples, which cannot monitor the GAN for producing practical samples because of sampling issues in generator input variables. Vanilla GAN can produce realistic test samples, but its impact on features is unclear and straightforward [14]. Therefore, a particular role should have an appropriate structure with a complicated expression, which is unable to develop for each conventional generative model [15].

CELEB-500K, described in [16], is a large training dataset for face recognition. There are 50M images in the dataset, representing 500K different celebrities. As a result, the method is more accurate than other publicly available datasets. On the other hand, our study focuses on working with a limited dataset using CelebA with 10K unique identities. The dataset is great for training and testing models for face recognition and logging in to mobile devices with an individual's face or searching through surveillance images for a particular person.

One possible light-weight computational application of GAN is mobile edge computing (MEC). The MEC was developed as a potential approach to significantly reduce the processing time, and communication resources on mobile devices [17]. Numerous studies have shown that GANs can be implemented in MEC with respect to latency [18], [19], security [20], and low power consumption [21], [22]. The study in [22] proven that their proposed GAN can reduce the time and resources needed for image validation.

This paper proposes light-weight generative adversarial networks (LiWGAN) to generate high quality images for non-image synthesis with minimum computational time for a less power computing. Our study employs a public facemask dataset due to the COVID-19 pandemic that remains debatable, and wearing a facemask is becoming more recommended. Facemasks are an initiative that incorporates the community as a prevention technique. Thus, the generated datasets would be helpful for other researchers to detect and classify facemask with occlusion challenges. We evaluated the quality of the specific image in terms of the Fréchet inception distance (FID) scores with state-of-art methods. For comparison with StyleGAN [23], we repeated the experiment using the same facemask dataset to identify the challenges of data augmentation in increasing the number of images and their quality. The main contributions of this study are as follows:

1) We built the Skip-Layer channel-wise excitation module, which revises the channel responses on a high-scale feature map via low-scale activation. SLE allows for a robust gradient flow across the model weights for faster training. Programmatically, it helps to disentangle styles and content like StyleGAN.
2) We propose a self-supervised discriminator, D, as the feature encoder with an additional decoder. We force D to learn a more descriptive feature map covering more regions from an input image, yielding more details about the setting to train a generator, G.
3) We validate our proposed method concerning other benchmarks utilising CelebA datasets and demonstrating that LiWGAN significantly reduced processing time in applications requiring less computing power.

II. RELATED WORKS

Current works related to GAN approaches mainly improved the image classifier, generation, and detection of fake and genuine images [2], [3], [14], [24], [25], [26], [27], [28], [29], [30]. Some studies have conducted GAN techniques with two processes: 1) focusing on learning augmented images and 2) using the conventional classification network [31]. Goodfellow *et al.* proposed the Vanilla GAN [2] for a generative model. They draw samples directly from the required data distribution without explicitly modeling the underlying density function.

A new approach called BicycleGAN was created from a conditional variational autoencoder GAN and conditional latent regressor GAN [30]. They developed a method to simultaneously implement the relationship between latent encoding and output to boost decoder performance without enforcing a tough decision. Moreover, BicycleGAN can yield diverse and visually pleasing outcomes in several image-to-image conversion problems.

Kodali *et al.* [26] proposed a novel gradient penalty scheme called DRAGAN to eradicate a low local equilibrium in non-convex sports. They proved that they could accomplish asymptotic convergence without requiring the discriminator to be adequately satisfied.

Some authors develop a GAN semi-supervised training scheme for chest anomaly classification, patch-based retinal vessel classification, and cardiac diagnosis [32], [33], [34]. They found that the method can perform better than the conventional supervised CNN. Proposing a softmax GAN, they substitute the classification loss with cross-entropy losses for the generator and discriminator in a single batch of images [27]. The authors proved that implementing importance sampling provides a rigorous approach to global optimization.

Adler *et al.* proposed a Banach Wasserstein GAN (BWGAN) to generate realistic samples from complex image distributions [24]. The strategy is to separate Banach spaces using a gradient penalty and allows practitioners to use the unique feature sets in the generator to determine which features to prioritize. A WGAN and InfoGAN combined to develop the unsupervised cell-level visual representation of images in histopathology [35]. The methods derived the attributes of the discriminator to construct a classifier.

ControlGAN adopts the auxiliary classifier GAN (ACGAN) to generate conditional samples as a classification layer in a discriminator that solves the overfitting [14]. ControlGAN's independent classifier distinguishes the classifier function from the discriminator, using artificial learning algorithms to classify an unlabeled dataset with minimal computational effort. The strategy is to maintain a proper equilibrium between the two issues operated concurrently by two different network modules.

Mode seeking GAN (MSGAN) was applied on the DCGAN and worked as an effective regularization term on the generator [36]. The regularization term drives the generators to investigate more minor modes by increasing the gap between produced images and the distance between the related latent codes. The regularization approach was implemented into existing conditional GAN without training or network structural changes. Regularization improves baseline frameworks' complexity without reducing image quality.

A new co-evolutionary method in [37] designed a CycleGAN that allows two portable generators to remove unnecessary filters, reducing memory consumption concurrently. In addition, the proposed method achieved high compression and optimization rates. The style information in the compressed images can be used on any off-the-shelf platform without affecting visual quality.

The effective DA network's goals are detecting out-of-distribution data and enhancing resilience without compromising classification accuracy [38]. The comparison of the effective DA and the state-of-art WideResNet proposed in [39] showed the effectiveness of both methods in various datasets. The initial novelty of the study is that when the input image is out of distribution, the predicted probabilities of the enhanced images could be inaccurate. As a result, the effective DA is more practical than previous approaches and can be readily adapted to diverse neural networks to increase security in real applications.

Supervised learning sentence compression, SeqGAN, and data screening were proposed as a framework for data augmentation [40]. The framework uses SeqGAN to create text data to address the lack of variety in widely used approaches. However, SeqGAN faces a significant hurdle while training on large texts. Hence the study presented a sentence reduction method. The sentiment words are preserved throughout the sentence reduction procedure to preserve more sentiment information. The produced data that contains inaccurate sentiment information can be removed using the proposed data screening method.

Face Augmentation Generative Adversarial Network (FA-GAN) was proposed in [41] to minimize the effect of deformation attribute distribution imbalances for the CASIA-WebFace dataset. Besides improving face recognition accuracy, the FA-GAN uses disentangled identity representations to manipulate various characteristics of an individual's face. Research on face recognition and synthesis tasks shows that the proposed network preserves identity well in restricted datasets.

An adaptive global and local bilevel optimization model (GL-GAN) proposed in [42] optimized the image from the local and global aspects. The local bilevel optimization model was proposed based on the discriminator's output feature matrix, in which each element evaluates image receptive field quality and determines the area with low quality. However, the GL-GAN needs extension studies for edge computing and mobile devices due to limitations where the proposed method selects a low-quality rectangular receptive field that causes overlapping images.

The author in [43] proposed DCGAN, known as adversarial learning-based data augmentation (Ada), to produce additional malicious users. The DCGAN-based data augmentation approach can produce better user embeddings than simple data augmentation methods, making it better at detecting malicious users in sparse-sample situations. However, although DCGAN can entirely imitate malicious users' dispersion, there is a constraint to the generated fake users. The hostile users are being injected into the system to make them more like genuine users to evade detection. As a result, these malicious users' false users are less likely to include attack characteristics.

A combination of GAN and re-id model called Jot-GAN was proposed to train the generator and re-id model concurrently to obtain their respective optimums using a discriminator [44]. Furthermore, the adversarial training and the produced samples enhance the re-id model's ability to trick the discriminator, thereby boosting its performance. Findings showed that the Jot-GAN surpassed the existing methods with the identification loss and triplets loss.

In 2018, the NVIDIA team introduced a style-based GAN model (StyleGAN) [23]. The normalization of the generator was restructured and regularized to facilitate a good mapping condition from latent to image codes. As a result, the authors enhanced the training performance for superior-quality images. The simplified data flow produced the most significant performance due to weight demodulation, lazy regularization, and algorithm optimization.

The advantages of StyleGAN are that it is easy to allocate an image produced to its root and improve its quality. However, training requires more computational time. Therefore, we take advantage of the weaknesses of StyleGAN to improve the performance of our proposed model, specifically in minimizing the computational time.

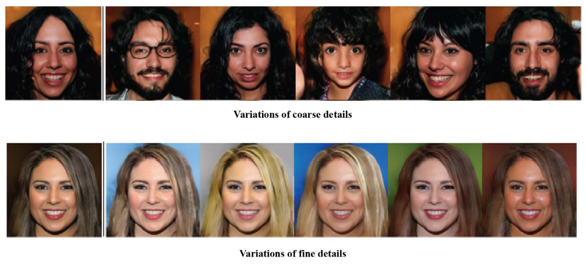


FIGURE 1. The variations of coarse and fine details [39].

III. SYSTEM MODEL

A. StyleGAN

The NVIDIA team developed StyleGAN in 2018 [23], [45], [46]. The authors proposed a new generator architecture that enables them to monitor various levels of data in the samples, from coarse details (e.g., head shape) to finer details (e.g., hair length and eye color), as shown in Fig. 1. StyleGAN applies the concept of ProgressiveGAN, in which the networks initiate at a lower resolution (4^2) and use more data (1024^2) to train the network on more progressively scaled layers of networks. Thus, the training time becomes considerably quicker and more stable. StyleGAN allows simpler things by mapping a network that encodes input vectors into an intermediate latent space, w. In addition, it controls various levels of detail. StyleGAN utilizes bi-linear sampling instead of down-sampling, and later applies on ProgressiveGAN. The bi-linear sampling is carried out by low-pass filtering the activation with a different second-order binomial filter after each upsampling layer and before each downsampling layer.

The comparison of the traditional StyleGAN and StyleGAN architectures [23], as illustrated in Fig. 2, shows the redesign of the style-based generator by omitting the input layer, referring to Fig. 2b. Instead, the latent vector, z in the latent input space, Z mapped to another vector of $w \in W$ with the 512 dimensions. The implementation of 8-fully connected layers, FC aims to simplify the mapping, f. The output of the mapping network, w, is then passed through a learned affine transformation, A, before passing into the synthesis network, an adaptive instance normalization (AdaIN) module. Finally, the model converts encoded mapping into a generated image. The input to AdaIN is styles $y = (\sigma, \mu)$ generated by applying A to w. The styles y controls adaptive instance normalization operations after each convolutional layer of the synthesis network, g. The following equation defines the AdaIN operation as,

$$AdaIN = (x, y) = \sigma(y)(\frac{x - \mu(x)}{\sigma(x)}) + \mu(y) \tag{1}$$

where each input feature map, x is normalized separately and then scaled and biased using the corresponding scalar components from style y. The normalized content input is simply scaled up by $\sigma(y)$ and shifted by $\mu(y)$. Thus, the dimension of y is twice the number of input x on that layer. The synthesis network contained 18 convolutional layers for each resolution: 4^2 and 1024^2.

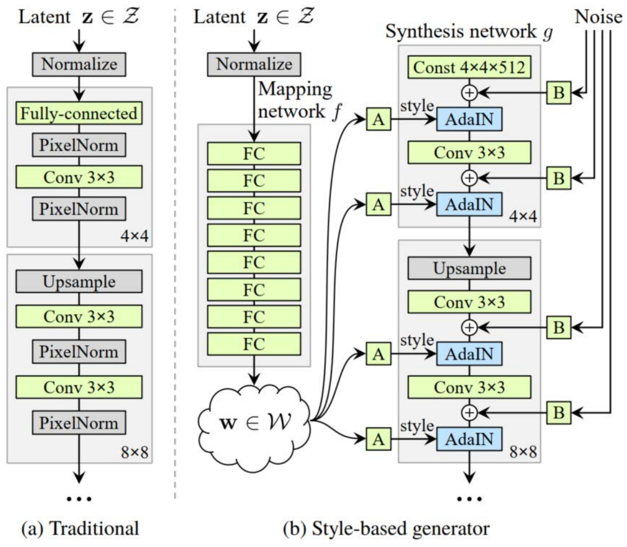


FIGURE 2. The architecture of StyleGAN [39].

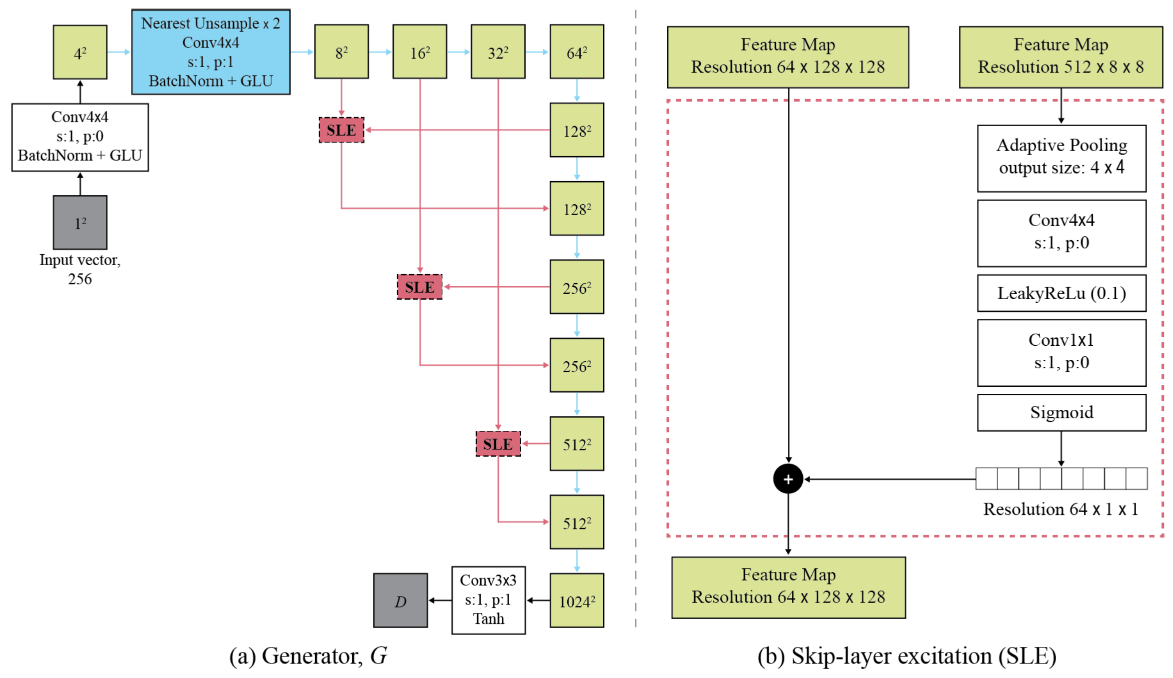
Style generation uses an intermediate vector at each level of the synthesis network, which may cause the network to learn the correlation between different levels. Therefore, the model randomly selects two input vectors, z_1 and z_2, and generates intermediate vectors, w_1 and w_2 to reduce the correlation. It then trains some of the levels using the first and switches method (in a random split point) to the other to train the remaining levels. The random split point switch ensures the networks do not learn the correlation effectively.

Finally, the generator offers a direct mechanism of creating stochastic detail by explicitly incorporating noise inputs. Each layer of the synthesis network receives separate single-channel images formed of uncorrelated Gaussian noise. First, the noise image is transmitted to all feature maps using learned per-feature scaling factors, as illustrated in Fig. 2b. The corresponding convolutional filtering operation result is subsequently subjected to the noise image.

IV. LIGHT-WEIGHT GAN

This study compared our proposed model, a LiWGAN, with StyleGAN. We adopted StyleGAN in [23], [45], and [46], including the model configuration and differentiable data-augmentation, for the best training on few-sample datasets. Furthermore, we compared our proposed model with StyleGAN based on the computing time because StyleGAN requires much more computing time to train. However, for non-image synthesis quality, we compared it regardless of the computing time. Therefore, we developed our proposed model using two proposed techniques, namely a skip-layer excitation module and a self-supervised discriminator.

A LiWGAN requires a generator, G, that can learn fast and a discriminator, D, to provide valuable signals to train G continuously, as illustrated in Fig. 3a. It utilizes a single convolution layer on each resolution in G and applies three input channels: 8^2, 16^2, and 32^2, including three output channels:



4²
Nearest Unsample x 2 Conv4x4 s:1, p:1 BatchNorm + GLU
8²
16²
32²
64²
Conv4x4 s:1, p:0 BatchNorm + GLU
SLE
128²
1²
Input vector, 256
256²
512²
D
Conv3x3 s:1, p:1 Tanh
1024²
(a) Generator, G
Feature Map Resolution 64 x 128 x 128
Feature Map Resolution 512 x 8 x 8
Adaptive Pooling output size: 4 x 4
Conv4x4 s:1, p:0
LeakyReLu (0.1)
Conv1x1 s:1, p:0
Sigmoid
Resolution 64 x 1 x 1
(b) Skip-layer excitation (SLE)

FIGURE 3. The structure of the generator, *G* in (a), and skip-layer excitation module in (b). Feature-maps consists of the spatial size and omits the channel number that presented in the green boxes. The blue box and blue arrows represent the same up-sampling structure. The red boxes represent the SLE component.

128^2, 256^2, and 512^2 for the convolution layer in both G and D. We present the structure of the G and D, referring to in Fig. 3 and Fig. 4, with descriptions of the component layers and forward flow. Generally, our proposed model is similar to StyleGAN. However, the structural designs make our proposed model much smaller than StyleGAN and substantially faster to train with skip-layer channel-wise excitation (SLE) implementation. Meanwhile, our proposed model remains robust on small datasets owing to its compact size with the two proposed techniques.

A. SKIP-LAYER CHANNEL-WISE EXCITATION

Generator, G, requires deepening combined with up-sampling needs, with further convolution layers. A deeper model with more convolution layers leads to a longer training time of the GAN owing to the increased number of model parameters and a weaker gradient flow through G.

SLE was redesigned based on the skip layer connections in [32] to train a deep model and strengthen the gradient signals between layers. As presented in Fig. 3, the architecture of the SLE displays ResBlock implemented skip connections as an element-wise addition between the activations from different convolution layers. In addition, channel-wise multiplications were performed between the activations to eliminate the heavy computation of convolution. The skip connections were used at a similar resolution in previous GAN works. However, skip connections were performed between resolutions into a more extended range in this study, as an equal spatial dimension was no longer necessary. The ResBlock is applicable with a shortcut gradient flow without additional computational cost to ensure that the SLE succeeds. The computation of SLE is as follows,

$$y = \mathcal{F}(x_l, \{W_i\}) \cdot x_h \tag{2}$$

where x and y are the input and output feature maps of the SLE component, function $\mathcal{F}$ comprises the operations on x_l and W_i specifies the learned module weights. The SLE component indicates that x_l and x_h are the feature maps at resolutions of 8^2 and 128^2, as shown in Fig. 3b. First, the adaptive pooling layer, F down-samples x_l into 4^2 along the spatial dimensions, and the convolutional layer later down-samples it into 1^2. A LeakyReLU is used to design nonlinearity, and x_l has the same channel size as x_h for other convolutionary layers. Finally, after a gating operation via a sigmoid function, the output from $\mathcal{F}$ multiplies x_h along the channel dimension yields y with the same shape as x_h.

SLE resembles the squeeze-and-excitation module (SE) proposed in [47]. However, the SE operates within one feature map as a self-gating module. In comparison, SLE performs between feature maps far away from each other. While SLE benefits from channel-wise feature re-calibration similar to SE, it also strengthens the gradient flow of the whole model like ResBlock. The channel-wise multiplication in SLE also coincides with instance normalization [48], [49], widely used in style transfer.

Similarly, we show that SLE enables G to automatically disentangle the content and style attributes, as in StyleGAN [46]. The performance of SLE on high-resolution feature maps, altering these feature maps is more likely to change the style attributes of the generated image [46], [50].

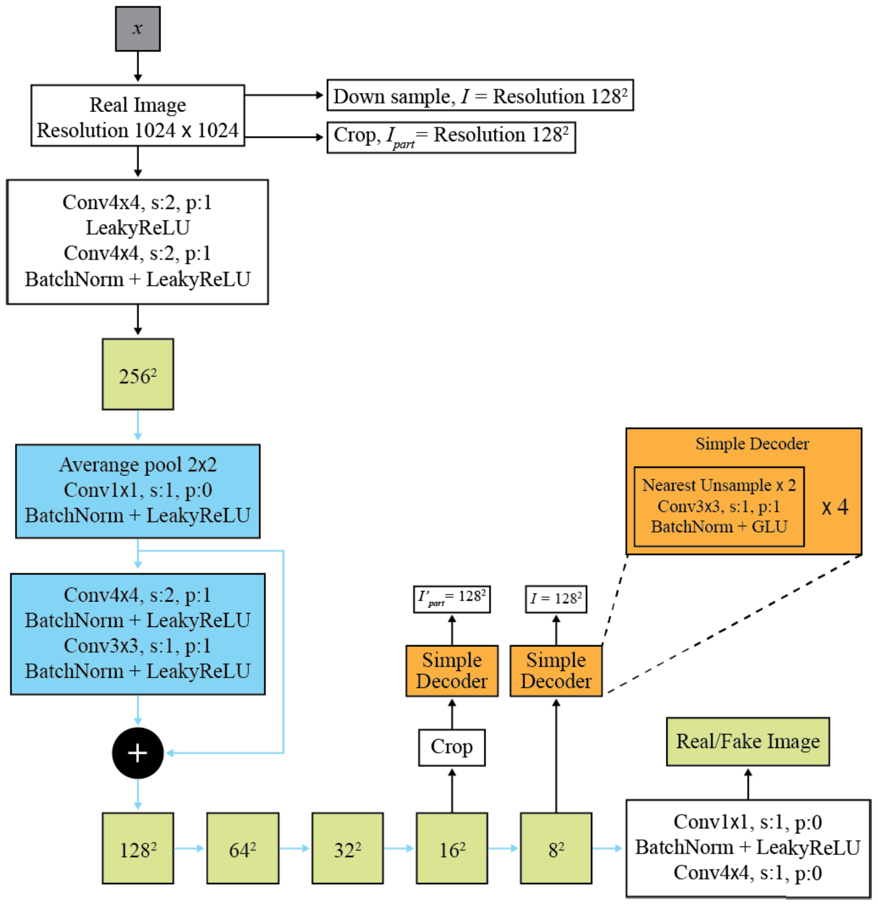


FIGURE 4. The structure and the forward flow of the discriminator. The same residual down-sampling structure is presented in the blue boxes, and the same decoder structure is presented in the orange boxes.

By replacing x_l in SLE with another synthesized sample, our G can generate an image with entire content but in the new replacing image style.

B. SELF-SUPERVISED DISCRIMINATOR

This study presented a simple regularization for D as an encoder and a train with small decoders. An auto-encoding training requires D to extract the image features that the decoder can reconstruct. The decoder and D were optimized to achieve a quick restoration loss trained on real samples. The decoders are optimized together with D on a simple reconstruction loss, trained only on real samples.

$$\mathcal{L}_{recons} = \mathbb{E}_{f\sim D_{encode}(x), x\sim I_a}[||\mathcal{G}(f) - \mathcal{T}(x)||] \tag{3}$$

where we propose the reconstruction loss $\mathcal{L}_{recons}$ between the intermediate feature maps the f, from the discriminator D. The function $\mathcal{G}$ contains the processing on f and the decoder, and the function $\mathcal{T}$ represents the processing of sample x from genuine images, I_a.

The self-supervised, D is illustrated in Fig. 4, where two decoders for the feature maps are employed on two scales: f_1 on 16^2 and f_2 on 8^2. The decoders have four convolution layers only to produce images at 128^2 resolution. Nevertheless, it may require additional computations, which is less than other regularization methods. The f_1 was randomly cropped with $\frac{1}{8}$ of its height and width, then crop the genuine image on the same portion to obtain the cropped image, I_c. Next, the genuine image was resized to obtain a down-sampled image, I. Then, the decoders were produced I'_c from cropped f_1, and I' from f_2. Finally, D and the decoders were trained to reduce the loss for the corresponding of I'_c to I_c and I' to I. The simple decoder has three convolution layers with the nearest unsample layer using gated linear unit (GLU) to alleviate the loss of sample images.

D extracts a more comprehensive representation from the inputs, covering the overall compositions (from f_2) and detailed textures (from f_1). Note that the processing in $\mathcal{G}$ and $\mathcal{T}$ is not limited to cropping; more operations remain to be explored for better performance. The auto-encoding approach is a typical method for self-supervised learning, which improves model robustness and generalization ability [51], [52], [53], [54]. From the perspective of GAN, a regularized D via self-supervision training strategies significantly improves the synthesis quality on G, which enhances the performance of auto-encoding.

Moreover, the method varies radically from merging the GAN and auto-encoder (AE). In previous works, G was primarily trained as a decoder in a latent space from D or

TABLE 1. Data augmentation in LiWGAN.

Image Transforms	Description
color and contrast	randomly change brightness, saturation, and contrast
cutout for occlusion images	creates random black boxes on the image to generate image occlusion
offset-x and offset-y for shift-variant images	randomly moves image by x and y-axis with repeating image
horizontal translation	randomly moves image on the canvas with black background

the adversary training with D as a potential loss compared to the training of AE. However, in this study, a pure GAN with a much simpler training schema was implemented in the model. The auto-coding training regularizes D without involving G. Therefore, hinge loss was implemented to train D and G iteratively. The advantage of hinge loss is that it can compute faster than other GAN losses.

V. AUGMENTATION

Image data augmentation is the best-known data augmentation type. It involves the creation of a transformed image in the training dataset that is the same class as the original image [55]. It requires some effort and cost to collect labeled datasets, particularly in disciplines like facial recognition using non-image synthesis, where the datasets are in millions. On the other hand, data augmentation refers to the practice of enhancing the original datasets.

This study used several basic augmentation techniques, such as rotation, shearing, cropping, zooming in and out, and many more. In this study, by default, the augmentation types were set to horizontal translation and cutout for occlusion images, with color and contrast omitted. The horizontal translation of an image of a person wearing a mask may make sense, as the photo can be taken from the right or left. However, a vertical-horizontal image of a person wearing a mask does not make sense.

It is probably not appropriate, given that it is unfeasible to see an upside-down person. The purpose of color and contrast is to generalize images trained using various lighting spectrums. Thus, an appropriate data augmentation technique must be chosen wisely within the context of the training dataset.

Several types of augmentation presented in this study are listed in Table 1. These four were executed due to the significant output for data augmentation in a low-data setting. A general recommendation is to use proper augmentations for the data and as much as possible; then, after some training, it is most destructive for image augmentation.

VI. SIMULATION PARAMETER

This section explains the datasets and methods used for LiWGAN to generate more facemask images and perform data augmentation. First, the data augmentation is described briefly to understand the LiWGAN implementation better. Then, the process of developing facemask images begins in correspondence with the data augmentation results. Table 2 presents the summary of the simulation parameter used in this study to apply LiWGAN to improve the performance of GAN model.

TABLE 2. Summary of simulation parameter.

Parameter	Description
Dataset	With Facemask: 3725 Without Facemask: 3828
Image Resolution	256^2 1024^2
Batch Size	8, 16, 32
FID Size	1000, 2000, 3000, and 5000
Iteration Training	10,000 iteration
GPU	NVIDIA Tesla P100
Programming Language	Python (Pytorch)
Development Platform	Google Colaboratory Pro

A. FACEMASK DATASET

A facemask detection dataset was utilized in this study to perform data augmentation and classification of face images with and without a mask. The facemask dataset was taken from Kaggle and found in [56]. The dataset consisted of 7553 RGB images in two separate folders with and without a mask. Images of faces with masks contained 3725 images, and faces without masks contained 3828 images. The facemask images were trained for 10,000 iterations for 256^2 resolution images.

B. DEVELOPMENT ENVIRONMENT

Google Colaboratory, the most well-known Google Colab, is used in this study as a development platform to run our proposed model via the Colab Notebook. Google Colab is a research project used for machine learning models on powerful hardware options, namely, the graphics processing unit (GPU) and tensor processing unit (TPU) [57]. Google Colab offers core machine learning and artificial intelligence libraries, such as TensorFlow, Matplotlib, and Keras, with either Python 2 or 3 runtimes pre-configured [58]. We used Google Colab Pro to work faster and with longer runtimes in this study. The GPU of NVIDIA Tesla P100 is utilized with high-memory virtual machines (VMs). We need a faster GPU and high RAM to run our proposed model because our proposed model needs to generate more images that require longer runtimes of more than 24 hours and less disconnection.

C. METRICS

The Fréchet inception distance (FID) computes the total semantic realism of the synthesized image. The FID created by Martin Heusel *et al.* [59] generates genuine images and improves the current inception score (IS). The conditional class predictions for synthetic images and the marginal probability for the predicted classes were combined to obtain the IS. However, the IS does not demonstrate how synthetic images interact with genuine images.

The purpose of the FID score is to measure synthetic images based on synthetic image data compared with the results of a set of genuine images in the target domain [60]. Therefore, a lower FID indicates better quality images; however, a high FID indicates a low-quality image and displays a linear relationship.

In this study, we let the generator G produce 5000 images and measure the FID between the synthesized images and the entire training set for datasets with more than 1000 images. Therefore, we used 1000, 2000, 3000, and 5000 images to compute the FID in this study. By considering the significant performance difference between our proposed model and the comparable models, FID is likely to be consistent with the others; thus, it is unnecessary to implement other metrics. The FID, d^2 provided in [56] is computed as follows,

$$d^2(x, g) = ||\mu_x - \mu_g||_2^2 + Tr(C_x + C_g - 2(C_x C_g)^{\frac{1}{2}} \quad (4)$$

where x is the genuine image, and g is the generated image. While, μ_x and μ_g refer to the feature-wise means of the genuine and generated images, respectively. The C_x and C_g refer to the covariance matrix for the genuine and generated feature vectors, known as sigma. The $||\mu_x - \mu_g||^2$ refers to the sum square difference between the two mean vectors, while Tr refers to the trace of a square matrix, the sum of the elements on the main diagonal (from the upper left to the lower right).

VII. RESULTS AND DISCUSSION

A. QUALITATIVE EVALUATION

We executed image augmentation with a single GPU and three hours of training for the dataset with 10,000 iterations. As explained in Section V, the augmentation results were produced based on the types of augmentations, for both with and without mask data. The augmentations can be combined with more than one augmentation to produce varied results. Fig. 5 shows some images combined all augmentation types, namely a) color and contrast, b) cutout for occlusion images, c) offset-x and offset-y for shift-variant images, and d) horizontal translation with an image size of 256^2 and batch size of 16.

The qualitative comparisons with StyleGAN based on resolutions of 256^2 and 1024^2 highlighted the efficiency of our proposed model as in Fig. 6. StyleGAN either converges slower or suffers from mode collapse, given the same batch size and training time. In contrast, our proposed model consistently generated satisfactory images. Note that the best results from our proposed model on facemask images only took one hour of training time, and the best performance was achieved at training for four hours with a resolution of 1024^2. For StyleGAN on the facemask, the images were from the best epoch, which corresponds to the training time and GPU in Table 3 and Table 4. The resolution of 1024^2 from StyleGAN is also limited, given the increased training time.

LiWGAN has the same features as StyleGAN with a channel-wise excitation module. It learns how to distinguish high images from G's convolution layers on various scales uncontrolled from high-level semantic attributes (style and content). The style-mixing results in Fig.6 show LiWGAN and StyleGAN with 256^2 and 1024^2 resolutions. While StyleGAN suffers from converging on the bottom high-resolution datasets, LiWGAN successfully learns the style representations along the channel dimension on the "excited" layers (i.e., for feature maps with 256^2 and 512^2 resolution).

TABLE 3. Comparison of computational cost of our model and StyleGAN based on batch sizes.

Resolution/Batch-size	StyleGAN	LiWGAN
Resolution: 256^2 Batch size: 8	Training Time: 5 hours Training GPU: 7.62 GB	Training Time: 1 hour Training GPU: 4.67 GB
Resolution: 256^2 Batch size: 16	Training Time: 9 hours Training GPU: 13.21 GB	Training Time: 3 hours Training GPU: 5.81 GB
Resolution: 256^2 Batch size: 32	Training Time: 15 hours Training GPU: 15.34 GB	Training Time: 6 hours Training GPU: 9.12 GB

TABLE 4. Comparison of computational cost based on 256^2 and 1024^2 resolutions.

Resolution/Batch-size	StyleGAN	LiWGAN
Resolution: 256^2 Batch size: 8	Training Time: 5 hours Training GPU: 7.62 GB	Training Time: 1 hour Training GPU: 4.67 GB
Resolution: 1024^2 Batch size: 8	Training Time: 39 hours Training GPU: 15.13 GB	Training Time: 4 hours Training GPU: 6.59 GB

Collecting large-scale image datasets are expensive for a particular character, genre, or subject. On these few-shot datasets, a data-efficient model is valuable for image generation. The computational cost comparison is evaluated and tabulated in Table 3, which presents the normalized models combined with a mask and without a mask on an NVIDIA Tesla P100 GPU and high memory virtual machines (VMs), implemented using PyTorch in Google Colab Pro. The comparison evaluated the training time per 10,000 iterations and trained the GPU with a resolution of 256^2 for multiple batch sizes, such as 8, 16, and 32, for the total images of 1000. We executed StyleGAN into the dataset to obtain a fair comparison with our proposed model.

The results were compared with StyleGAN to determine the effective, efficient, and best training time and training GPU. According to the findings, StyleGAN took five hours of training time and a training GPU of 7.62 GB for batch-size 8, which is considered a longer training time and large training GPU to be compared with our proposed model. our proposed model only took one hour of training time and 4.67 GB of training GPU, with a shorter training time and a small GPU. The differences in training time and training GPU for StyleGAN and our proposed model are varied for all the computed batch sizes. For example, the batch size of 32 for our proposed model took approximately six hours to train

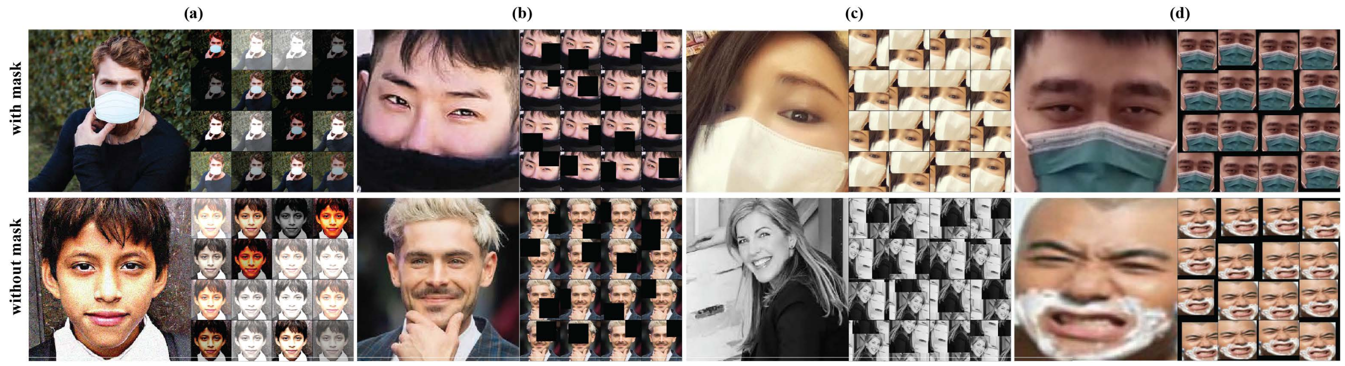


FIGURE 5. Types of augmentations for images with a mask and without a mask dataset: (a) variations of color and contrast, (b) various cutout for occlusion images, (c) offset-*x* and offset-*y* for shift-variant images, and (d) horizontal translation, with an image size of 256^2 and batch-size of 16.

10,000 iterations of images with only 9.12 GB of training GPU. The training GPU for a batch size of 32 is almost similar to batch size 8 for StyleGAN with only 2 GB GPU variance.

We adopt StyleGAN with recent studies from [23], [45], [46], including the model configuration and differentiable data-augmentation, for the best training on few-sample datasets. We show that our proposed model not only achieves superior performance on the few-shot datasets but is also much more computationally efficient than the compared method. The training time and GPU performance are evaluated based on the low resolution of 256^2 and high resolution 1024^2 for a batch size of 8, as tabulated in Table 4. StyleGAN took 39 hours of training time, which is more than 24 hours to execute 100 images for 10,000 iterations. Limited access to high-memory virtual machines, GPUs, and RAM may result in a long training time with frequent disconnections. Hence, our proposed model produced the best training time and GPU as our objective is to reduce the computational time and GPU training size.

The training loss of the model for batch sizes 8 and 32 with an image size of 256^2 was computed to determine the effectiveness and efficiency of the model. Our proposed model's strength is the self-supervised discriminator, which optimizes both the decoder and the discriminator to achieve rapid restoration loss. Therefore, the self-supervised loss is compared with the generator loss and discriminator loss for every 1000 iterations until it reaches 10,000.

Fig. 7 presents the discriminator, generator training loss, and self-supervised loss for a batch size of 8 with an image size of 256^2. The results show the self-supervised loss of our proposed model obtained zero loss for every 1000 iterations for batch-size eight iterations. The generator loss started with a considerable loss value at 45.97 and decreased to 1.85 at 10,000 iterations, which was a good training loss. The discriminator loss began at 6.01 and was reduced to 0.06.

Meanwhile, Fig. 8 presents the training loss for a batch size of 32 with the image size of 256^2. The results showed a similar training loss as batch size 8, exclusive of the generator loss. The generator loss began from the negative value of -35.77, which increased the graph to a positive loss value of 2.37. Nevertheless, the discriminator loss showed a varied gap between batch sizes 8 and 32 for every iteration. At 10,000 iterations of batch size 32, the discriminator loss produced a high discriminator loss at 0.29, beginning at 7.47, slightly higher than the batch size of 8. The self-supervised loss has obtained a 0.00 loss at 10,000 iterations. Therefore, our proposed model can optimize the training loss using the proposed self-supervised discriminator.

Furthermore, self-supervised training results improve our proposed model's robustness and generalization ability. From the viewpoint of GANs, self-supervised methods improve the representation efficiency for machine learning, which results in better auto-coding. The results of the training loss show a better training loss for batch size 8 with a decreasing loss value to zero.

B. QUANTITATIVE EVALUATION

We also tested our model on facemask datasets with sufficient training samples for a more thorough evaluation. We trained the full StyleGAN for approximately four to five days on the facemask dataset with a batch size of 8 on two Tesla P100 GPUs. Instead, we trained our model for only 24 hours, with a batch size of eight on a single GPU. The standard method for calculating FID is to generate 50,000 images and use the entire training set as the reference distribution. We computed the FID for 1000, 2000, 3000, and 5000 images at 1024^2 resolution. Compared with StyleGAN, as tabulated in Table 6, the results show that LiWGAN can work with many datasets with a minimum computer budget. In addition, the FID results proved that LiWGAN boosts distance performance consistently compared to StyleGAN.

We experimented with the two proposed modules in Table 5, where both SLE and decoding-on-*D* (decode) can separately boost the model performance. Results show that the two modules are orthogonal to each other in improving the model performance, and the self-supervised *D* makes the most significant contribution. Significantly, StyleGAN diverges rapidly after the training time. In contrast, our model is less likely to cause mode collapse among the tested datasets. Furthermore, our model maintains good synthesis

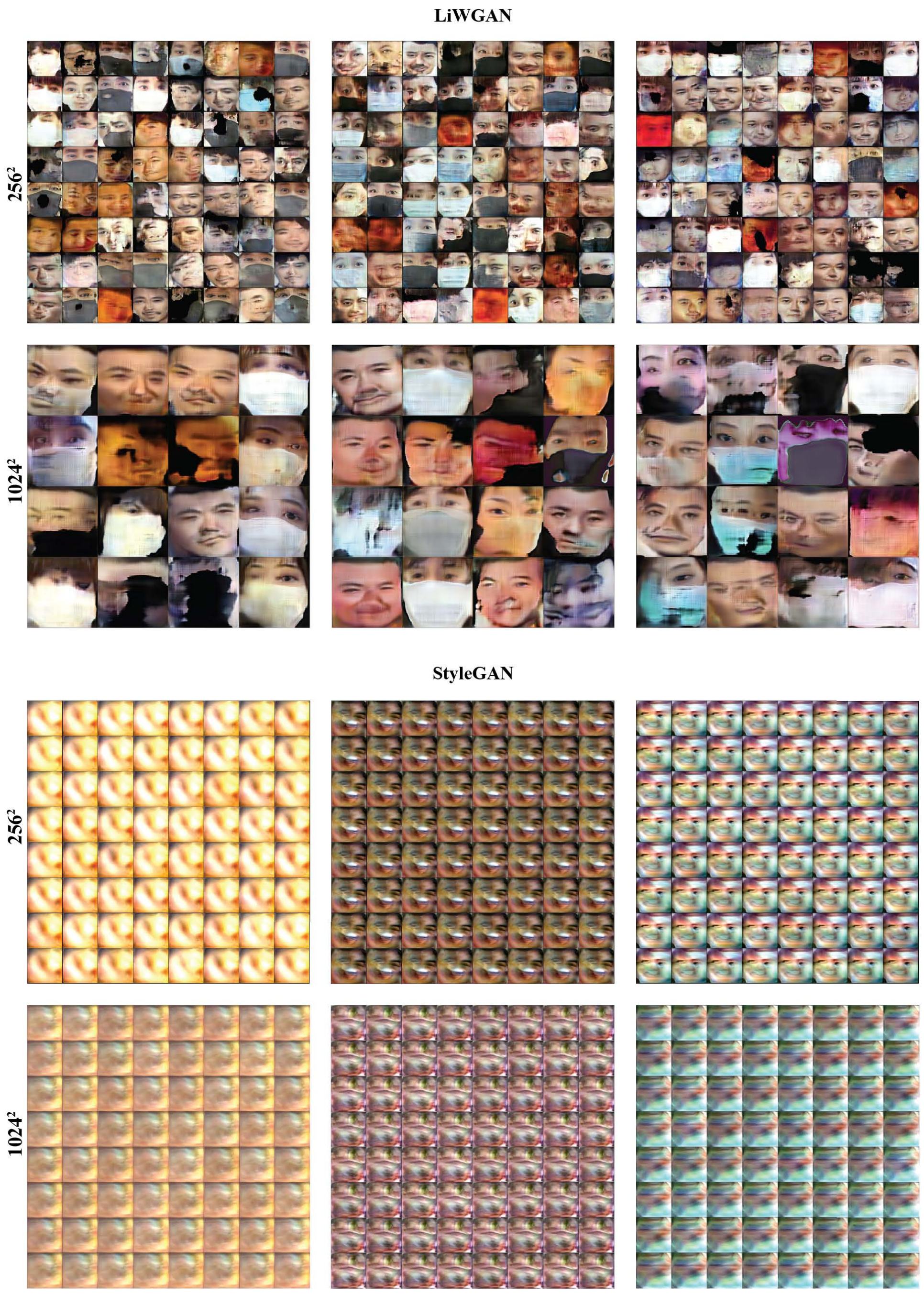


FIGURE 6. Style-mixing results of LiWGAN and StyleGAN with 256^2 and 1024^2 resolutions. LiWGAN is trained for only three to four hours on a single GPU.

quality and does not collapse after training for 20 hours. Finally, the decoding regularization on D prevents the model from diverging.

In addition to the auto-encoding training for D, we show that D with other common self-supervising strategies boosts GAN's performance by treating genuine images as a unique class and classifying them. For instance, we train D to predict the original aspect ratio of the genuine image. Then, it reshaped to square when supplied to D. Auto-encoding causes D to pay attention to more regions of the input image, resulting in a more comprehensive feature map for the input image. In contrast, a classification task does not guarantee that D covers the entire image. Instead, the task drives D to focus only on small regions because the model can find class

TABLE 5. Comparison of FID at 1024^2 resolution for 1000, 2000, 3000, and 5000 images.

Image Number	StyleGAN	LiWGAN
1000	98.85	204.08
2000	111.86	175.28
3000	**219.97**	**220.43**
5000	95.28	142.78

TABLE 6. Comparison with state-of-the-art methods in terms of FID and processing time per iteration.

Ref.	Methods	Dataset	FID	Image Type	Processing Time per Iteration
[26]	DRAGAN	CelebA	157.5	Non-Synthesis	N/A
[36]	MSGAN	CIFAR10	29.65	Synthesis	N/A
		Facades	92.84	Non-Synthesis	
		Maps	152.43	Non-Synthesis	
[37]	CycleGAN	Horse to Zebra	96.15	Synthesis	0.73s
		Zebra to Horse	157.90		
		Summer to Winter	78.58		
		Winter to Summer	79.16		
[42]	GL-GAN	Oxford Flower	51.30	Synthesis	1.63s
		CelebA	12.37	Non-Synthesis	12.24s
[23]	StyleGAN	LSUN Car	3.27	Synthesis	0.61s
		LSUN Cat	8.53		
		LSUN Church	4.21		
		LSUN Horse	3.83		
	LiWGAN	**Facemask**	**142.78**	**Non-Synthesis**	**1.03s**

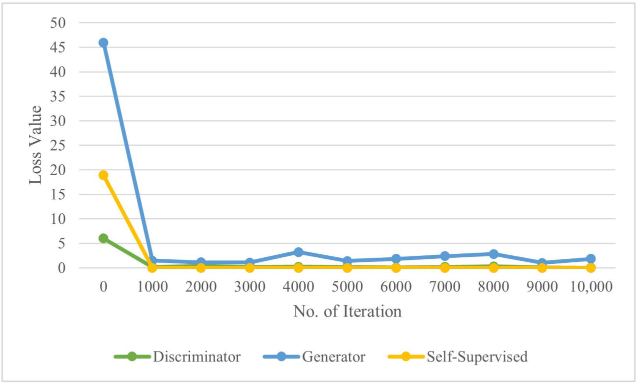


FIGURE 7. The training loss consisted of the discriminator, generator, and self-supervised loss for a batch size of 8 with an image size of 256^2.

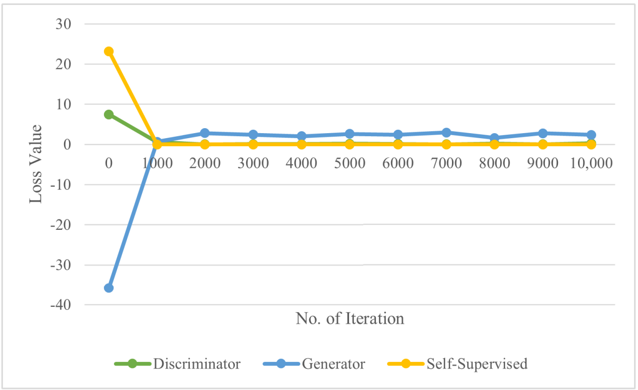


FIGURE 8. The training loss consisted of the discriminator, generator, and self-supervised loss for a batch size of 32 with an image size of 256^2.

cues from small regions of the images. Focusing on limited regions or patterns is a typical overfitting behavior widely occurring for *D* in vanilla GANs.

Furthermore, we compared LiWGAN with the state-of-art methods, including GL-GAN [42], DAG + Dist-GAN [61], CycleGAN [37], MSGAN [36], DRAGAN [26], and StyleGAN [23], in terms of FID score, type of image (synthesis or non-synthesis), and processing time per iteration as tabulated in Table 6. These state-of-art methods utilized various datasets with image synthesis and non-image synthesis. We compared the FID to analyze the methods based on the image quality of the original image. The FID is acceptable when the image quality is almost zero. However, this depends on the dataset used to compute the FID. Therefore, our study analyzed the FID score for the datasets that consist of non-image synthesis to produce a fair comparison with the state-of-art methods.

The findings show LiWGAN generated a 142.78 FID score for a non-image synthesis dataset, which is considered a high score compared to the image synthesis dataset of StyleGAN, with the lowest FID. StyleGAN generated an image synthesis task using LSUN datasets as their training images. Image synthesis has high-resolution images compared to non-image synthesis. However, non-image synthesis embeds the training images in a low dimensional space; thus, it comprises poor quality image synthesis.

LiWGAN produced a slightly lower FID score than CycleGAN using Zebra to Horse, MSGAN using Maps, and DRAGAN using CelebA. In this case, the Zebra to Horse dataset contained image synthesis, while other datasets are non-image synthesis datasets. However, the processing time per iteration shown by LiWGAN takes a few seconds longer than CycleGAN to generate the image, while MSGAN and DRAGAN are not stated the specific processing time by the authors.

On the other hand, the processing time of GL-GAN is longer than other state-of-art methods, including LiWGAN at 12.24s for CelebA datasets, a non-image synthesis with a good FID score. Furthermore, the author optimized the local bilevel for poor image quality. Hence, GL-GAN enhanced the poor-quality images to a high quality non-image synthesis. Moreover, it helps to reduce the FID score nearly to zero in generating the images.

We validated LiWGAN with CelebA dataset for non-image synthesis and compared the FID score and processing time per iteration. The results tabulated in Table 7 show that LiWGAN produced a better FID score at 91.31 than GL-GAN and DRAGAN methods. Furthermore, the processing time per iteration is considerably increased from 1.03s for 142.78 FID to 3.50s for 91.31 FID. Despite this, LiWGAN showed a significant improvement in FID score and processing time compared to GL-GAN.

TABLE 7. FID and processing time per iteration results using CelebA dataset.

Ref.	Method	FID	Processing Time per Iteration
[42]	GL-GAN	12.37	12.24s
[26]	DRAGAN	157.5	N/A
	LiWGAN	**91.31**	**3.50s**

The importance of the dataset selection mainly has a considerable impact on the FID score; we aim to generate good images for non-image synthesis in a low setting. GAN methods require large memory in a real application. Hence, LiWGAN was proposed as one of the best options for light-weight MEC because it requires less computing power than standard GAN methods.

Fine-tuning from a pre-trained GAN [36], [62], [63] has been the norm for image generation on datasets with few samples. However, its performance highly depends on the semantic consistency between the new dataset and the available pre-trained model. According to Zhao *et al.*, fine-tuning performs worse than training from scratch in most cases when the content from the new dataset strays away from the original one [64].

VIII. CONCLUSION

This study proposed a LiWGAN method to produce and generate more non-image synthesis in a low setting. LiWGAN proposed two techniques, a skip-layer channel-wise excitation module (SLE) and a self-supervised discriminator, to boost the performance of the GAN and reduce the computing power for mobile devices, as shown in Table 4. Comparing the resolution and batch size, such as batch sizes of 8, 16, and 32, with a resolution of 256^2, including the highest resolution of 1024^2 into LiWGAN to analyze the FID score in identifying the genuine and fake images while generating high quality images. In addition, all batch sizes and resolutions are analyzed based on the computational time and memory of the GPU. To prove LiWGAN is adequate for light-weight MEC, some analyses have been carried out using the CelebA dataset and compared with other GAN methods in terms of FID score and processing time per iteration. As a result, LiWGAN is considerably fitted for less computing power due to its low processing time, which is suitable for mobile devices. In future work, LiWGAN can improve the FID score and the computational time required to work fast for multiple types of datasets.

REFERENCES

[1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2017.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[4] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, *arXiv:1711.04340*.

[5] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: When to warp?" in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–6.

[6] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.

[7] Y. Li, N. Xiao, and W. Ouyang, "Improved boundary equilibrium generative adversarial networks," *IEEE Access*, vol. 6, pp. 11342–11348, 2018.

[8] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 324–335, May 2020.

[9] C. Wu, L. Herranz, X. Liu, J. van de Weijer, and B. Raducanu, "Memory replay GANs: Learning to generate new categories without forgetting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[10] K. Kim and H. Myung, "Autoencoder-combined generative adversarial networks for synthetic image data generation and detection of jellyfish swarm," *IEEE Access*, vol. 6, pp. 54207–54214, 2018.

[11] Q. Shi, X. Liu, and X. Li, "Road detection from remote sensing images by generative adversarial networks," *IEEE Access*, vol. 6, pp. 25486–25494, 2017.

[12] Y. Shin, H. A. Qadir, and I. Balasingham, "Abnormal colon polyp image synthesis using conditional adversarial networks for improved detection performance," *IEEE Access*, vol. 6, pp. 56007–56017, 2018.

[13] H. Ge, Y. Yao, Z. Chen, and L. Sun, "Unsupervised transformation network based on GANs for target-domain oriented image translation," *IEEE Access*, vol. 6, pp. 61342–61350, 2018.

[14] M. Lee and J. Seok, "Controllable generative adversarial network," *IEEE Access*, vol. 7, pp. 28158–28169, 2019.

[15] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14985–15006, 2019.

[16] J. Cao, Y. Li, and Z. Zhang, "Celeb-500 K: A large training dataset for face recognition," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2406–2410.

[17] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao, "5G on the horizon: Key challenges for the radio-access network," *IEEE Veh. Technol. Mag.*, vol. 8, no. 3, pp. 47–53, Sep. 2013.

[18] K. P. Dirgantoro, J. M. Lee, and D.-S. Kim, "Generative adversarial networks based on edge computing with blockchain architecture for security system," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 039–042.

[19] Q. Cui, Z. Zhou, Z. Fu, R. Meng, X. Sun, and Q. M. J. Wu, "Image steganography based on foreground object generation by generative adversarial networks in mobile edge computing with Internet of Things," *IEEE Access*, vol. 7, pp. 90815–90824, 2019.

[20] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, "Intrusion detection for secure social Internet of Things based on collaborative edge computing: A generative adversarial network-based approach," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 134–145, Feb. 2022.

[21] I. Kaur, E. L. Lydia, V. K. Nassa, B. Shrestha, J. Nebhen, S. Malebary, and G. P. Joshi, "Generative adversarial networks with quantum optimization model for mobile edge computing in IoT big data," *Wireless Pers. Commun.*, vol. 2021, pp. 1–21, Jun. 2021.

[22] Q. Zeng, L. Zhou, Z. Lian, H. Huang, and J. Y. Kim, "Privacy-enhanced federated generative adversarial networks for Internet of Things," *Comput. J.*, May 2022, Art. no. bxac060, doi: 10.1093/comjnl/bxac060.

[23] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8107–8116.

[24] J. Adler and S. Lunz, "Banach Wasserstein GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[25] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*.

[26] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," 2017, *arXiv:1705.07215*.

[27] M. Lin, "Softmax GAN," 2017, *arXiv:1704.06191*.

[28] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, *arXiv:1606.01583*.

[29] C. Zhang, X. Ouyang, and P. Patras, "ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 363–375.

[30] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

[31] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101552.

[32] B. Lecouat, K. Chang, C.-S. Foo, B. Unnikrishnan, J. M. Brown, H. Zenati, A. Beers, V. Chandrasekhar, J. Kalpathy-Cramer, and P. Krishnaswamy, "Semi-supervised deep learning for abnormality classification in retinal images," 2018, *arXiv:1812.07832*.

[33] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Chest X-ray generation and data augmentation for cardiovascular abnormality classification," *Proc. SPIE*, vol. 10574, Mar. 2018, Art. no. 105741M.

[34] A. Lahiri, V. Jain, A. Mondal, and P. K. Biswas, "Retinal vessel segmentation under extreme low annotation: A GAN based semi-supervised approach," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 418–422.

[35] B. Hu, Y. Tang, E. I.-C. Chang, Y. Fan, M. Lai, and Y. Xu, "Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 3, pp. 1316–1328, May 2019.

[36] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1429–1437.

[37] H. Shu, Y. Wang, X. Jia, K. Han, H. Chen, C. Xu, Q. Tian, and C. Xu, "Co-evolutionary compression for unpaired image translation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3234–3243.

[38] C.-H. Lin, C.-S. Lin, P.-Y. Chou, and C.-C. Hsu, "An efficient data augmentation network for out-of-distribution image detection," *IEEE Access*, vol. 9, pp. 35313–35323, 2021.

[39] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

[40] J. Luo, M. Bouazizi, and T. Ohtsuki, "Data augmentation for sentiment analysis using sentence compression-based SeqGAN with data screening," *IEEE Access*, vol. 9, pp. 99922–99931, 2021.

[41] M. Luo, J. Cao, X. Ma, X. Zhang, and R. He, "FA-GAN: Face augmentation GAN for deformation-invariant face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2341–2355, 2021.

[42] Y. Liu, H. Fan, X. Yuan, and J. Xiang, "GL-GAN: Adaptive global and local bilevel optimization for generative adversarial network," *Pattern Recognit.*, vol. 123, Mar. 2022, Art. no. 108375.

[43] J. Wang, M. Gao, Z. Wang, C. Lin, W. Zhou, and J. Wen, "Ada: Adversarial learning based data augmentation for malicious users detection," *Appl. Soft Comput.*, vol. 117, Mar. 2022, Art. no. 108414.

[44] Z. Zhao, R. Song, Q. Zhang, P. Duan, and Y. Zhang, "JoT-GAN: A framework for jointly training GAN and person re-identification model," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 1s, pp. 1–18, Feb. 2022.

[45] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12104–12114.

[46] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4217–4228, Dec. 2021.

[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[48] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[49] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.

[50] B. Liu, K. Song, Y. Zhu, G. de Melo, and A. Elgammal, "TIME: Text and image mutual-translation adversarial networks," 2020, *arXiv:2005.13192*.

[51] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6391–6400.

[52] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[53] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[54] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2020.

[55] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[56] O. Gurav. (2020). *Face Mask Detection Dataset*. Accessed: Nov. 30, 2020. [Online]. Available: https://www.kaggle.com/omkargurav/face-mask-dataset

[57] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. New York, NY, USA: Apress, 2019.

[58] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance analysis of Google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[59] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

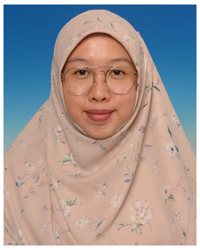
[60] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[61] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, "On data augmentation for GAN training," *IEEE Trans. Image Process.*, vol. 30, pp. 1882–1897, 2021.

[62] A. Noguchi and T. Harada, "Image generation from small datasets via batch statistics adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2750–2758.

[63] Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. van de Weijer, "MineGAN: Effective knowledge transfer from GANs to target domains with few images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9329–9338.

[64] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient GAN training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7559–7570.



NURUL AMIRAH MASHUDI (Graduate Student Member, IEEE) received the Diploma degree in computer science from the International Islamic College, Kuala Lumpur, and the B.Sc. degree in graphics and multimedia software and the M.Sc. degree in systems engineering from Universiti Teknologi Malaysia (UTM), Kuala Lumpur, Johor Bahru. Currently, she is working as an Information Technology (IT) Lecturer at City University Malaysia, Johor Bahru Campus. Her current research interests include image recognition, pattern recognition, image processing, image analysis, and machine learning applications.

NORULHUSNA AHMAD (Senior Member, IEEE) received the master's degree in electrical engineering (telecommunication) and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia (UTM), Kuala Lumpur, in 2003 and 2014, respectively. Currently, she is a Senior Lecturer at the Razak Faculty of Technology and Informatics, UTM. Her current research interests include future communication theory, the massive IoT technologies, UAV communication, machine learning application, multiple access techniques, image processing, and green energy communication.

NORLIZA MOHD NOOR (Senior Member, IEEE) received the B.Sc. degree in electrical/electronic engineering from Texas Tech University, Lubbock, TX, USA, and the master's degree in electrical engineering (by research) and the Ph.D. degree in electrical engineering (signal processing) from Universiti Teknologi Malaysia. She is currently a Professor at the Razak Faculty of Technology and Informatics, UTM, Kuala Lumpur Campus. Her research interests include image analysis and machine learning.