# A novel generalized adversarial image method using descriptive features

**Truong Phi Ho, Pham Duy Trung, Bui Thu Lam**

***Abstract*— Currently, machine learning not only solves simple problems such as object classification but also machine learning is widely applied in the field of computer vision such as identification systems, object detection, and modules in the authentication system, intelligent processing algorithms such as automatic driving, chatbot, etc. Deep learning models on GAN networks can automatically generate image data such as objects, animals, human faces, or the like by learning the word features of images in datasets such as MS-COCO, ImageNET, CUB, etc. Using this technique, attackers can fake images in some cases with malicious intent. In this paper, the authors propose to build a Generative Adversarial Network to create images that fool the target model YOLOv7, INCEPTIONv3. Experimental results on the CUB dataset show our proposed model's ability to generate adversarial examples is highly effective with an average image generation time equal to 0.16 seconds/an image. The successful rate of fooling the model reached over 85%, average recognition rate reached over 45% for the YOLOv7 model. When experimenting on the INCEPTIONv3 model, the successful rate of fooling the model reached over 95%, average recognition rate reached over 50%. The image fidelity evaluated by the PSNR index reached an average of greater than 29.**

***Tóm tắt*— Hiện nay, học máy không chỉ giải quyết các vấn đề đơn giản như phân lớp đối tượng mà còn được ứng dụng rộng rãi trong lĩnh vực thị giác máy tính như hệ thống nhận dạng, phát hiện đối tượng và trong các mô đun của các hệ thống xác thực, các thuật toán xử lý thông minh như lái xe tự động, rô bot chat,... Các mô hình học sâu dựa trên GAN có thể tự động tạo ra dữ liệu hình ảnh như đồ vật, động vật, khuôn mặt người hoặc những thứ tương tự bằng cách học các đặc điểm từ của hình ảnh trong các bộ dữ liệu như MS-COCO, ImageNET, CUB,... Bằng kỹ thuật này, kẻ tấn công có thể giả mạo hình ảnh trong một số trường hợp với mục đích xấu. Trong bài báo này, nhóm tác giả đề xuất xây dựng một mạng sinh đối kháng để tạo ra hình ảnh thực hiện đánh lừa mô hình mục tiêu là YOLOv7, INCEPTIONv3. Kết quả thử nghiệm trên tập dữ liệu CUB cho thấy khả năng tạo hình ảnh đối kháng do mô hình chúng tôi đề xuất đạt hiệu quả cao với thời gian tạo ảnh trung bình là 0,16 giây/một ảnh. Tỷ lệ đánh lừa mô hình thành công đạt trên 85%, tỷ lệ nhận dạng trung bình đạt trên 45% đối với mô hình YOLOv7. Khi thử nghiệm trên mô hình INCEPTIONv3, tỷ lệ đánh lừa mô hình thành công đạt trên 95%, tỷ lệ nhận dạng trung bình đạt trên 50%. Độ trung thực của hình ảnh được đánh giá bằng chỉ số PSNR đạt mức trung bình lớn hơn 29.**

***Keywords*— *Deep learning; Generative adversarial networks; adversarial examples.***

***Từ khóa*— *Học sâu; mạng sinh đối kháng; mẫu đối kháng.***

## I. INTRODUCTION

Deep learning plays a central role in propelling the latest breakthroughs in artificial intelligence and Machine learning, yielding significant repercussions across various domains of technology and everyday existence. Its profound influence stems from being a foundational tool that drives innovation in AI and ML, offering unprecedented capabilities and driving progress in fields ranging from computer vision to natural language processing. Given its wide-ranging impact, deep learning stands as an important force that contributes to the trajectory of modern technology and significantly impacts our daily lives [1, 2]. Recently, deep neural networks have become susceptible to input patterns, known as adversarial examples. The changes are invisible to us but can easily fool the neural network during the test or deployment phase. Vulnerabilities to adversarial examples

This manuscript is received on November 23, 2023. It is commented on December 17, 2023 and is accepted on December 20, 2023 by the first reviewer. It is commented on November 24, 2023 and is accepted on December 20, 2023 by the second reviewer.

become one of the major risks to the application of DNN in security-critical areas.

DNNs have been shown to significantly improve performance in many applications such as computer vision [3, 4], biometrics [5, 6], text processing [7], data social networking data [8], natural language processing [9], information security [10, 11], health [12], economics, finance [13], and many more applications [14], etc. However, AI-based systems are prone to safety and security threats at every stage of the process, from collecting and preparing initial training data to inference and deployment. Attacks and defenses based on adversarial examples pose a big problem.

The gap between AI technology creation and its implementation in practice is widening. In order to equip AI systems that facilitate deep learning for real-world applications and to acquaint researchers with the inherent risks in the complete lifecycle of AI model development and deployment, it is crucial to address the various stages involved. These stages encompass data collection and processing, model selection and training, as well as model deployment and system integration [15].

The survey shows that the issue of security vulnerabilities of DNN networks against attacks using adversarial examples is of research interest. In addition, the major challenges in adversarial examples and the development of solutions are also of interest to many researchers. Most DNNs are fooled by adversarial examples [16]. These are images that are noisy by adding small noises in the input image that are unrecognizable to the human eye, but strongly affect the output of the deep learning model, causing it to be skewed in recognition. This drives the creation of more and more adversarial examples to identify the model's vulnerabilities before putting them to use. Besides, AE also facilitates different DNN algorithms to achieve stability by providing more diverse training data. With the widespread implementation of AI and DNN-based technologies, there are numerous incentives to increase the security of any DNN-based technology and to protect against Malicious attacks to falsify the DL model. The issue of adversarial examples has become a very active arena in recent years that has attracted enormous attention and effort from both academia and industry.

Most of the existing attack algorithms rely on optimizing the $L_{norm}$ parameter to fool the model by the FGSM method [16]. Most of the previously published development methods explicitly compute a perturbation vector to generate the adversarial examples. Towards finding an alternative to optimization-based methods, we have since studied Generating Adversarial Examples with the aim of mapping from an input image that is considered safe, and close to the natural image. Instead of generating a perturbation intermediate vector, we construct the GAN network detailed in the following III to directly generate the so-called adversarial example image.

In addition, the effectiveness of deep learning models depends greatly on the amount of data collected and the computing power of the training server. The utilization of larger datasets for training deep learning models has demonstrated a noteworthy enhancement in their accuracy [17]. However, in the above cases, the parties involved need to share their local data. This is a big obstacle because this data can be sensitive and private data for each party.

In much previous research [18, 19], the authors proposed to build a data generation network but most used it on small data sets with a small number of classifications. Our method is different from previous studies, which use datasets with large real-world images, or natural samples. The main contribution of the author team is not only in creating images to check the accuracy of the model but also these images can be used during training to enhance the robustness of the deep learning model.

The rest of the paper is organized as follows. We first present the Generative Adversarial Network in section II. Section III describes the proposed GAN model for generating adversarial images. Section IV mentions experiments and results. We conclude the paper with conclusions and future works in Section IV.

## II. GENERATIVE ADVERSARIAL NETWORKS FOR GENERATE ADVERSARIAL IMAGES

Adversarial example - AE is an input data type used in machine learning and artificial intelligence where a small perturbation in the input data can cause a trained model to make inaccurate predictions corpse.

Generative adversarial networks - GANs aim to generate new data after the learning process. GANs can spawn a new face, person, text, script, symphony, or the like without knowing it beforehand completely non-existent in reality. These images can be defined as adversarial examples according to the conceptual understanding of AE just presented.

GANs employ a versatile modeling approach that autonomously learns and uncovers patterns within input data. This innovative technique facilitates the generation of coherent and logical outputs by leveraging the knowledge acquired from the original dataset. GANs can train general models by simulating a supervised approach to training problems. GAN comprises two interconnected sub-models that engage in a dynamic interplay, where they both oppose and support each other, ultimately leading to the production of more realistic outputs:

- *Generator model:* Trained to generate new patterns.

- *Discriminator model:* It endeavors to ascertain the source of the input data, specifically whether it originates from the generator model's original dataset or if it is a generated sample.

The idea of GAN originates from a zero-sum non-cooperative game, simply understood as a two-player fighting game like chess, if one person wins, the other will lose. At each turn, both want to maximize their chances of winning and minimize the opponent's chances of winning. The Discriminator $(D)$ and Generator $(G)$ in the GAN network are like two opponents in the game. More interestingly, GAN is also compared by author groups to a minimax game, a crime police game: criminal G creates fake money, and police D learns to distinguish real from fake. The more police try to distinguish real money from fake, the more criminals rely on police feedback to improve their ability to create fake money, trying to confuse the police. Access research on GAN from previous works. From there, we conceived the idea of building a GAN that generates data for use in resistance training to increase the robustness of the machine learning model.

We also propose to use multi-stage GAN with some benefits as: *(i)* High-resolution image generation: after each stage, the generated dummy image will be improved sharper and clearer; *(ii)* Controlling the learning process and

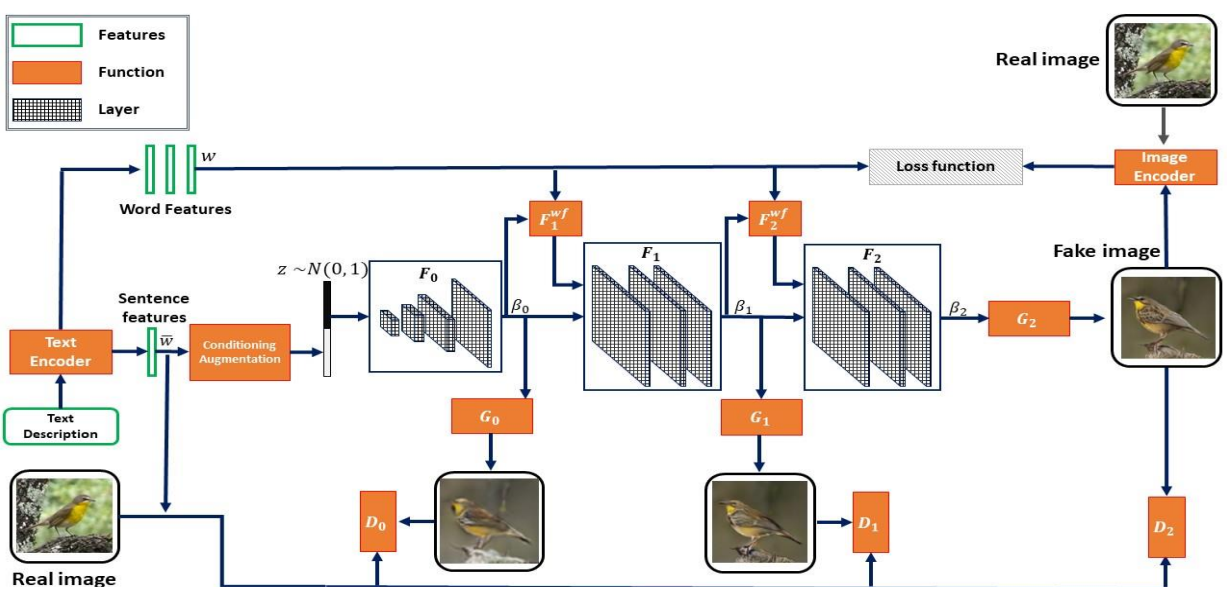



Figure 1. The overview architecture of proposed GAN model for generating adversarial images

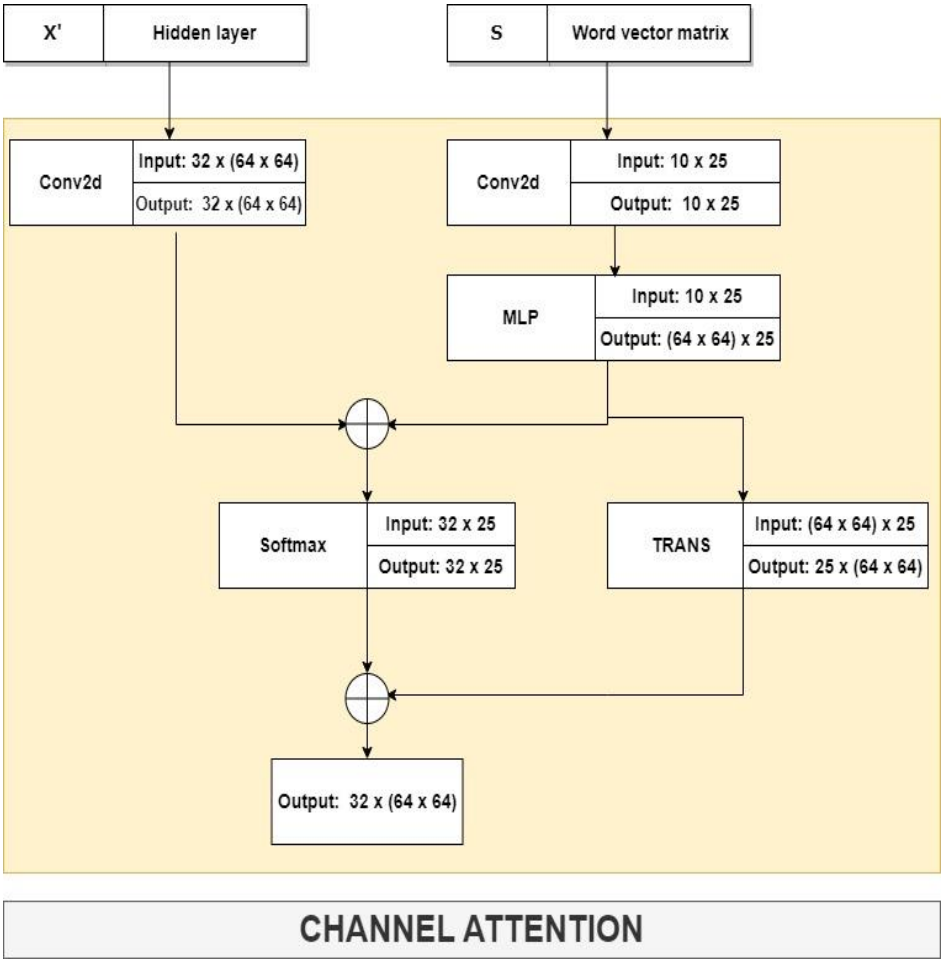



Figure 2. The proposed channel attention architecture

reducing workload: Making model training easier and allows the model to be adjusted as needed; *(iii)* Increased diversity of input data: Staged image generation also allows the GAN model to use a variety of input data types, such as from text, audio, or images low resolution.

The structural parameters and the specific adversarial image-generating network model proposed by the research team in this publication will be presented in section III.

## III. THE PROPOSED GAN MODEL FOR GENERATING ADVERSARIAL IMAGE

### *A. Overview architecture of the proposed GAN*

The idea of the method is to build a model that creates an image from the text. The model will repeat this process many times to improve its predictability, specifically:

- *Sentence feature:* is that all the words in the description sentence are converted to a numeric form combined with a random noise vector to produce a basic image, with the overall color and shape of the object described in the text.

- *Characteristic words:* used in the following stages, it is simply a word in a descriptive sentence. Each of these words will go through the Attention Mechanism ($F_i^{wf}$ in this paper). Attention synthesizes information about the context corresponding to a word and encodes that context in the word vector itself, helping the model focus on the important features of the text, and at the same time helping to generate corresponding images. than with text description. This attention mechanism will be applied to the layers of the neural network $F_1$, $F_2$ to create more detailed images. The model employs attention mechanisms to modify the layer's weights in the neural network adaptively.

In the context of generating images from descriptive text inputs, the initial phase of the process concentrates on outlining the basic shapes and colors of the object described, resulting in the production of low-resolution images. In the latter stage, the input is the image generated from the previous stage and follows the pgGAN [20], and then produces a high-resolution image.

We over-generalize the model detailed in Figure 1. Surveying previously published [21], the proposed model achieves better accuracy for recognition rate performance with inception score in experimental results on CUB dataset.

Firstly, a low-resolution image is produced from the given text using a conditional GAN. The initial image serves as a starting point for the second stage. It combines with the text to create a high-quality image using a CGAN [22-24]. The two stages work together to capture the overall structure and layout of the image in the first stage and then add details and textures in the second stage to create a final high-resolution image.

Our model employs an attention mechanism [25] to create high-resolution images based on textual descriptions. The model consists of a text encoder and a visual encoder. Each encoder is responsible for encoding the low-resolution image or text. These encodings are then utilized to generate an attention map, highlighting the input image's pertinent areas. The attention mechanism enables the model to concentrate on crucial regions of the image, resulting in more authentic images with improved details.

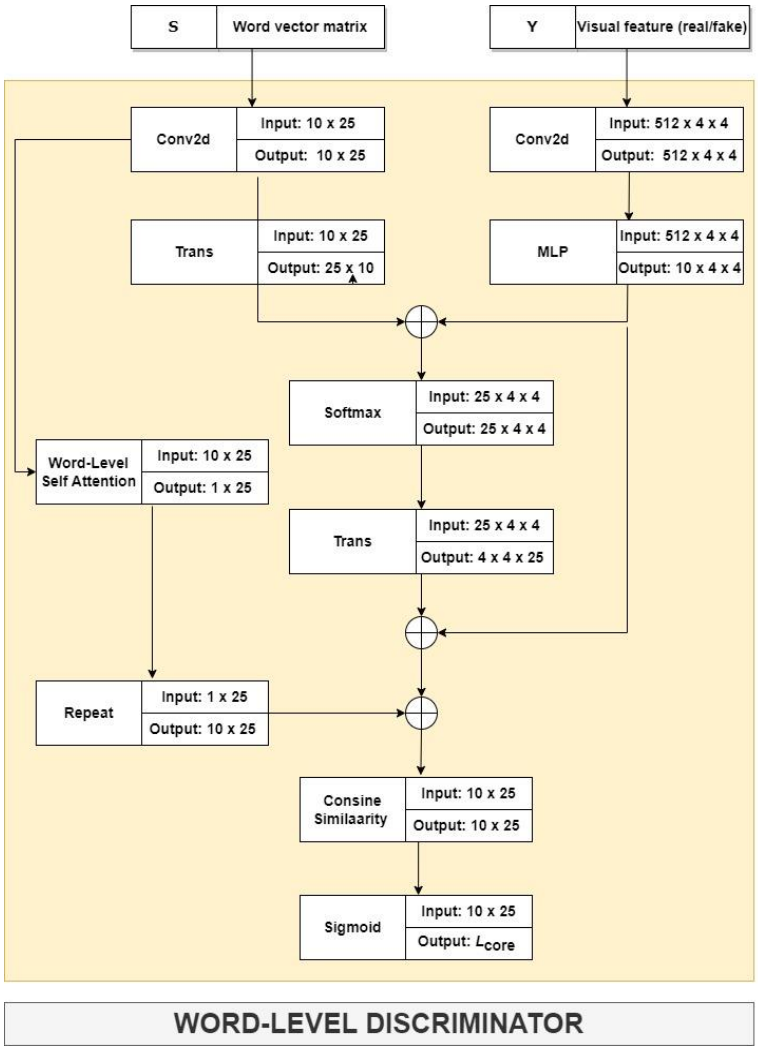



Figure 3. The proposed word-level Discriminator architecture

*B. Architecture*

With our method, some additional information will be required where $z$ is a noise vector taken from a normal distribution, $w$ is the word vector matrix and $\overline{w}$ is the global sentence matrix, hidden states $(\beta_0,\ \beta_1, \ldots, \beta_{m-1})$ as input and generate small to the large-scaled image by Equations (1), with:

$i = 1, 2, \ldots, m-1$:

$$\beta_0 = F_0\big(z, f(\overline{w})\big), \qquad \beta_i = F_i\left(\beta_{i-1}, F_i^{wf}(w, \beta_{i-1})\right), \tag{1}$$

where $f$ is a Conditioning Augmentation that converts $\underline{\overline{w}}$ into a conditioning vector via the formula $f(\overline{w}) = \mu(\overline{w}) + (\sigma(\overline{w}) \odot \varepsilon)$ where $\mu$ is the mean of the vector $\overline{w}$, $\sigma$ is the matrix diagonal covariance and $\varepsilon$ is a normal distribution $N(0,1)$. Figure 2 and Figure 3 are show the channel attention architecture and our proposed word-level Discriminator architecture.

The function $F_i^{wf}(w, \beta)$ is a learned correspondence function (this function is used to find the correlation between two different data sets) trained by vectors from $w$ and image features from the previous $\beta$ hidden layer. The columns in $\beta$ depict the feature vectors of the image's subregions. These vectors are dynamically calculated using the word vectors associated with $\beta_j$, as Equation (2):

$$c_j = \sum_{i=1}^{T-1} \theta_{j,i} w_i', \tag{2}$$

where:

$$\theta_{j,i} = \frac{exp(u'_{j,i})}{\Sigma_{k=0}^{T-1} exp(u'_{j,k})},$$

$$u'_{j,i} = \beta_j^T w'_i, w'_i$$

is the perceptron layer $w'_i = \beta_i w_i$ and indicates the weight of model. We then assign the word context matrix to the image feature set $\beta$ by $F_i^{wf}(w, \beta) = (c_0, c_1, \dots, c_{N-1})$. Subsequently, the generated image features are combined with the corresponding word context features to produce the final image in the subsequent stages. In which, $F_i^{wf}$, $f$, $F_i$, $G_i$ are neural network models. The $F_i$ is a type of Convolutional Neural Network used to map images into semantic vectors. The intermediate layers of the CNN can learn the local features of various sub-regions present within the image. On the other hand, the later layers know the global parts of the image. During the initial stage, GAN produces low-resolution images that encompass the general characteristics outlined in the text description.

The primary objective of this stage is to classify the generated images based on their alignment with the provided text description. Therefore, generator $G_i$ and discriminator $D_i$ (the architecture of these two networks is shown in

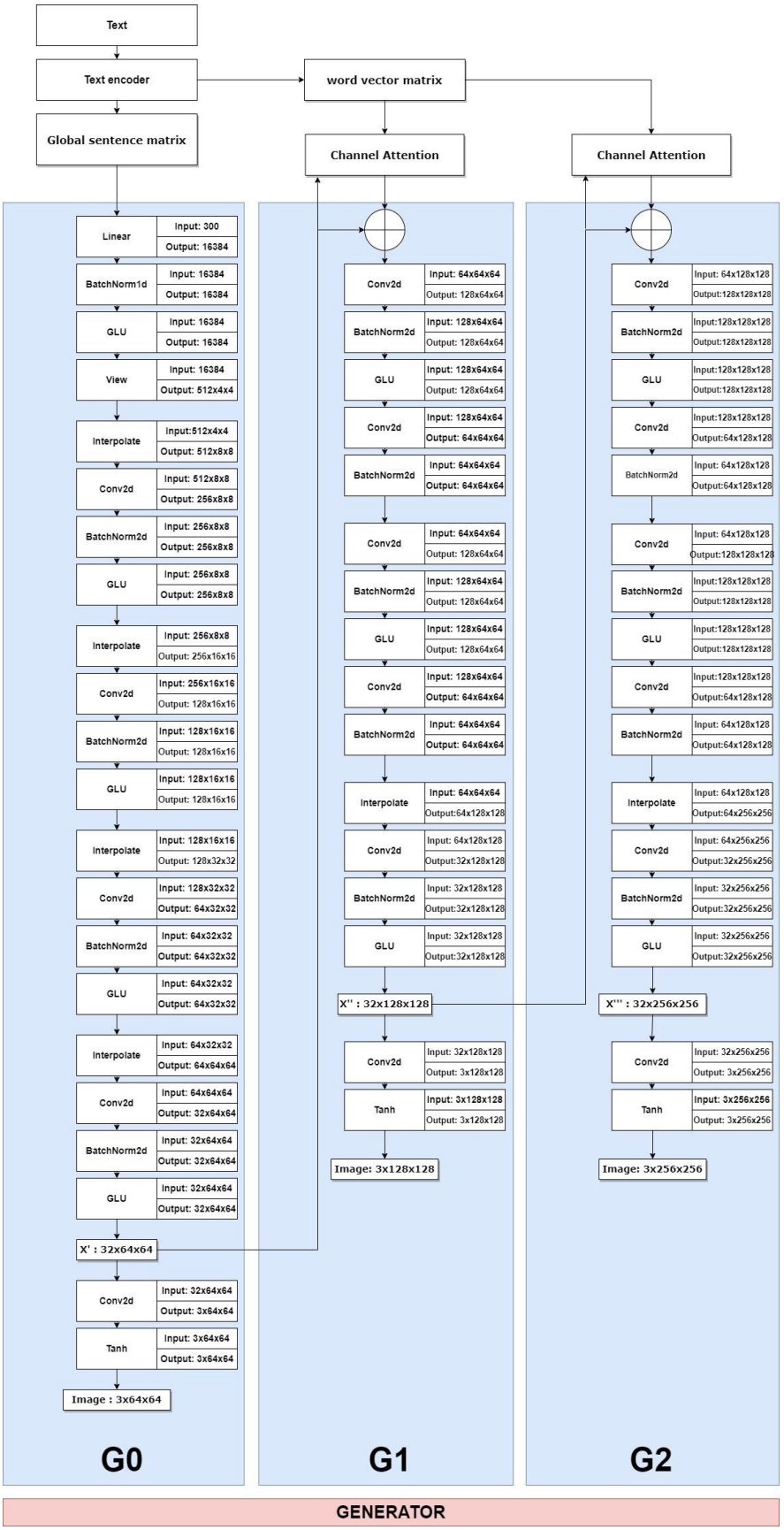



Figure 4. The proposed Generator architecture

Figure 4 and Figure 5 are calculated according Equation (3):

$$\mathcal{L}_{G_i} = E_{\beta_i \sim pF_i}\left[log\ D_i\left(G_i(\beta_i)\right)\right] - E_{\beta_i \sim pF_i}\left[log\ D_i(G_i(\beta_i), \overline{w})\right] \quad (3)$$

The overall loss function determines if an image is real or fake, while the conditional loss function checks if the image matches the accompanying text. During training, the generator $G_i$ and each discriminator $D_i$ are trained to distinguish between real and fake inputs. This is accomplished by minimizing the cross-entropy loss function specified in (4):

$$\mathcal{L}_{D_i} = unconditional\ loss + conditional\ loss, \quad (4)$$

where:

$$unconditional\ loss = -E_{\hat{x}_i \sim p_{data_i}}[log\ D_i(x_i)] - E_{\beta_i \sim pF_i}\left[log\ \left(1 - D_i(G_i(\beta_i))\right)\right]$$

$$conditional\ loss = -E_{\hat{x}_i \sim p_{data_i}}[log\ D_i(x_i, \overline{w})] - E_{\beta_i \sim pF_i}\left[log\ \left(1 - D_i(G_i(\beta_i), \overline{w})\right)\right],$$

where $x_i$ is a distribution of true images from $p_{data_i}$ at $i^{th}$, and $\beta_i$ is a distribution of $p_{F_i}$ also at $i^{th}$. Methods to improve image quality will be presented in next section.

### *C. The methods to improve image quality*

Our method trains a model that learns to capture useful information omitted from the previous stage producing a high-resolution image and displaying detailed features. In the following 2 stages the loss functions of generator $G_i$. The Generator network has three main components are G0, G1, and G2 as Figure 4.

Loss function of $G$ is calculated according to the Equation (5):

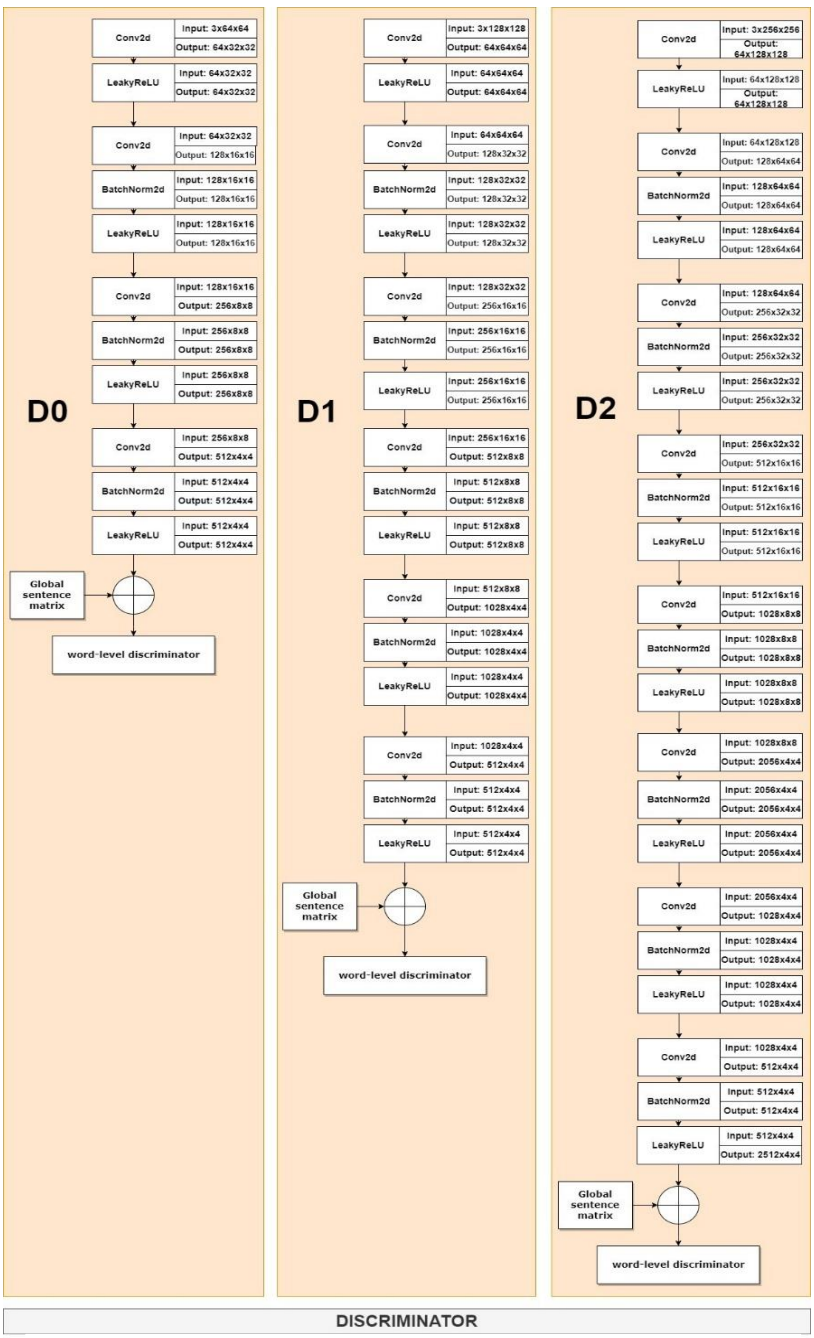



Figure 5. The proposed Discriminator architecture

$$\mathcal{L}_G = \sum_{i=0}^{m-1} \mathcal{L}_{G_i} + \mathcal{L}_\Lambda, \quad (5)$$

where $m$ is the stage number, $\mathcal{L}_{G_i}$ is defined:

$$\mathcal{L}_{G_i} = \mathbb{E}_{\beta_i \sim pF_i}\left[\log D_i\left(G_i(\beta_i)\right)\right] - \mathbb{E}_{\beta_i \sim pF_i}[\log D_i(G_i(\beta_i), \bar{w})]$$

The Discriminator network has three main components are *D0*, *D1*, and *D2* as Figure 5. Loss function of $D$ is calculated according (6):

$$\mathcal{L}_D = \sum_{i=0}^{m-1} \mathcal{L}_{D_i}, \quad (6)$$

where $m$ is the stage number, $\mathcal{L}_{D_i}$ is defined:

$$\mathcal{L}_{D_i} = -\mathbb{E}_{\hat{x}_i \sim p_{data_i}}[\log D_i(x_i)] - \mathbb{E}_{\beta_i \sim pF_i}\left[\log\left(1 - D_i\left(G_i(\beta_i)\right)\right)\right] - \mathbb{E}_{\hat{x}_i \sim p_{data_i}}[\log D_i(x_i, \bar{w})] - \mathbb{E}_{\beta_i \sim pF_i}\left[\log\left(1 - D_i(G_i(\beta_i), \bar{w})\right)\right].$$

To determine how similar text and image features are, we use a comparison method that calculates the fine-grained image-text matching loss $\mathcal{L}_\Lambda$ at the level of individual words. We conduct the reference according to the surveyed study [26]. Similarities for matching image and text pairs between full image ($Img$) and full text in $D$ are defined by the Equation (7):

$$R(Img, D) = \log\left(\sum_{i=1}^{T-1} \exp\left(\gamma_1 R(c_i, w_i)\right)\right)^{\frac{1}{\gamma_1}} \quad (7)$$

The parameter $\gamma_1$ is a factor that determines the degree to which the importance of word pairs is exaggerated concerning the most relevant context region. When $\gamma_1 \to \infty, R(Q, D)$ is approximately equal to $max_{i=1}^{T-1} R(c_i, w_i)$.

In which, $R(c_i, w_i)$ is the relationship between word and image $i^{th}$ using cosine similarity between $c_i$ and $w_i$, $R(c_i, w_i) = \frac{(c_i^T w_i)}{(\|c_i\| \|w_i\|)}$. $\mathcal{L}_\Lambda$ is designed for Semi-Supervised Learning, where single supervision is between the whole image and the whole sentence (sequence of words). The loss function is defined to measure the probability of the image matching the corresponding text description with the equation (8):

$$\mathcal{L}_1^w = -\sum_{i=1}^{M} \log \frac{\exp\left(\gamma_2 R(Img_i, D_i)\right)}{\sum_{j=1}^{M} \exp\left(\gamma_2 R\left(Img_i, D_j\right)\right)} \quad (8)$$

The loss function whose text description matches the corresponding image by the equation (9):

$$\mathcal{L}_2^w = -\sum_{i=1}^{M} \log \frac{\exp\left(\gamma_3 R(Img_i, D_i)\right)}{\sum_{j=1}^{M} \exp\left(\gamma_3 R\left(Img_j, D_i\right)\right)}. \quad (9)$$

We can further obtain the loss function $\mathcal{L}_1^{\bar{w}}$ and $\mathcal{L}_2^{\bar{w}}$ using the sentence vector $\bar{w}$ and the global image vector $\bar{v}$ is the global vector for the whole image) are redefined by the equation with $R(\bar{v}, \bar{w}) = \frac{(\bar{v}^T \bar{w})}{(\|\bar{v}\| \|\bar{w}\|)}$. Loss function of proposed model as Equation (10):

$$\mathcal{L}_\Lambda = \mathcal{L}_1^w + \mathcal{L}_2^w + \mathcal{L}_1^{\bar{w}} + \mathcal{L}_2^{\bar{w}}. \quad (10)$$

Experiments to demonstrate the effectiveness of the model are presented in section IV.

## IV. Experiments and results

### A. Experiments

We conduct extensive tests on the CUB dataset [27]. The CUB dataset consists of 8,855 images for training and 2,933 images for testing, and ten text descriptions accompany each image.

The authors have trained in 600 epochs with 64 batches. The first stage uses the Text-to-image Generation method to convert from text to a 64 × 64 image. The first stage uses an Image-to-image translation method that takes input from the image generated from the first stage and the real image produces a 128 × 128 image. Similar to the next stage with input is the image generated from the first stage, and using our

proposed method to improve the loss function produces a 256 × 256 image. A pre-trained bidirectional LSTM (as described in [28]) is used to encode text descriptions, resulting in a sentence feature of size 256 and a word feature of length 18 and size 256. Meanwhile, images are encoded using a Convolutional Neural Network pre-trained on ImageNet, resulting in semantic vectors. Based on tests on organized validation sets, we use Adam's optimizer by Yi et al. [29] with learning rate 0.0002; setting $\gamma_1 = 5, \gamma_2 = 0.5, \gamma_3 = 1, M = 64$ with CUB dataset.

We use the proposed model with the parameters just mentioned to generate images from given sentences. Then use this image dataset to class identification on target models. The authors statistic and evaluation of the results. Specific results of testing on the YOLOv7 model [30] are shown in Table 1; testing on the INCEPTIONv3 model [31] is shown in Table 2. The authors also evaluated the data set generated through as PSNR values. Specific results are shown in Table 3. We provide some discussion about our proposed model in the next section.

*B. Results and discussions*

To examine the effect of the proposed model, we use the rate successes of adversarial images to relabel the original images with target labels $P_s$ (%), $N_{in}$ is the number of images generated according to the text description in the Text column. $P_s$ is calculated as Equation (11):

$$P_s = \frac{N_s}{N_{in}} \times 100, \quad (11)$$

where $N_s$ is the number of generated images that successfully fool the model according to the specified label "bird". We do not show the results of $N_s$ in the table. $R_{avg}$ (%) is the average recognition rate on each set of images generated by sentences of different sizes. Specific experimental results on the two target models: YOLOv7 and INCEPTIONv3 are shown in Table 1 and Table 2.

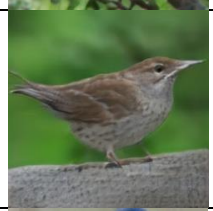
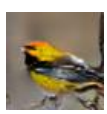
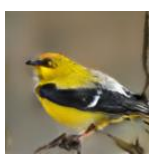
TABLE 1. RESULT OF EXPERIMENT ON YOLOV7 MODEL

| No | *Text description* | $N_{in}$ | 64 × 64 | | 128 × 128 | | 256 × 256 | |
|---|---|---|---|---|---|---|---|---|
| | | | $P_s$ | $R_{avg}$ | $P_s$ | $R_{avg}$ | $P_s$ | $R_{avg}$ |
| 1 | small bird with yellow body feathers, black wings and a small orange beak | 1090 | 2.57 | 33 | 27.34 | 38 | 83.76 | 49 |
| 2 | this bird has a grey breast and abdomen with darker brown on the back and head and those feathers are lightly speckled | 1100 | 0.27 | 29 | 8.64 | 37 | 71.82 | 51 |
| 3 | this bird has a white and tan belly and a blue crown and dark blue grey secondaries and primaries | 1100 | 3.36 | 32 | 38.18 | 37 | 83.82 | 47 |
| 4 | this is a slender bird with a yellow belly, a hint of orange in the tail feathers and rectrices, and brown wings with light wing bars | 1100 | 1.27 | 31 | 28.64 | 38 | 95 | 54 |
| 5 | the bird has a red overall body color aside from a black patch at the base of it's bill | 1100 | 4.18 | 34 | 24.36 | 38 | 66.36 | 46 |
| 6 | this small bird has and open beak with black and white striped wing bars and a brown breast | 1090 | 1.19 | 33 | 27.25 | 37 | 83.94 | 52 |
| 7 | small bird with yellow body feathers, black wings and a small orange beak | 1000 | 2.3 | 34 | 17.1 | 38 | 82 | 47 |
| **Average** | | $\sum = 7580$ | 2.16 | 32.29 | 24.50 | 37.57 | **87.64** | **49.43** |

TABLE 2. RESULT OF EXPERIMENTS ON INCEPTIONV3 MODEL

| No | *Text description* | $N_{in}$ | 64 × 64 | | 128 × 128 | | 256 × 256 | |
|---|---|---|---|---|---|---|---|---|
| | | | $P_s$ | $R_{avg}$ | $P_s$ | $R_{avg}$ | $P_s$ | $R_{avg}$ |
| 1 | small bird with yellow body feathers, black wings and a small orange beak | 1090 | 85.96 | 38.13 | 98.9 | 41.85 | 100 | 47.78 |
| 2 | this bird has a grey breast and abdomen with darker brown on the back and head and those feathers are lightly speckled | 1100 | 66.18 | 40.34 | 96 | 41.24 | 100 | 30.26 |
| 3 | this bird has a white and tan belly and a blue crown and dark blue grey secondaries and primaries | 1100 | 77.09 | 40.67 | 98.91 | 63.93 | 99.91 | 49.78 |
| 4 | this is a slender bird with a yellow belly, a hint of orange in the tail feathers and rectrices, and brown wings with light wing bars | 1100 | 76.64 | 52.90 | 96 | 41.62 | 99.64 | 45.12 |
| 5 | the bird has a red overall body color aside from a black patch at the base of it's bill | 1100 | 68 | 26.46 | 86.91 | 39.18 | 96.09 | 44.53 |
| 6 | this small bird has and open beak with black and white striped wing bars and a brown breast | 1090 | 60.09 | 47.07 | 93.76 | 46.41 | 100 | 62.91 |
| 7 | small bird with yellow body feathers, black wings and a small orange beak | 1000 | 72.7 | 43.16 | 99.4 | 68.08 | 100 | 89.97 |
| **Average** | | $\sum = 7580$ | 72.38 | 41.25 | 95.70 | 48.90 | **99.37** | **52.90** |

TABLE 3. SOME IMAGES ARE GENERATED BY THE PROPOSED MODEL ACCORDING TO TEXT DESCRIPTION

| *Size of image* / *Text description* | 64 × 64 | 128 × 128 | 256 × 256 |
|---|---|---|---|
| small bird with yellow body feathers, black wings and a small orange beak |  |  |  |
| this bird has a grey breast and abdomen with darker brown on the back and head and those feathers are lightly speckled |  |  |  |
| this bird has a white and tan belly and a blue crown and dark blue grey secondaries and primaries |  |  |  |
| small bird with yellow body feathers, black wings and a small orange beak |  |  |  |

The results in Table 1 and Table 2 show that images created in the Later Stage have a higher recognition rate than the First Stage and the successful rate of fooling the target model is also more effective than images rough is created in the First Stage. Table 3 shows some images generated based on the features described from the sentence in three stages with three different sizes. Comparison between the results on YOLOv7 model and INCEPTIONv3 model also shows that the success rate for the INCEPTIONv3 model is higher than that of YOLOv7 model because INCEPTIONv3 uses CUB to train the model. This also proves that the images generated by our proposed model achieve good performance. We also conduct statistical benchmarks to evaluate the entire generated data set including values $T_s$ is the total time to generate $N_{in}$ images, $T_{avg}$ is the average of time in which the proposed model successfully creates an adversarial image. In addition, $PSNR$ in Equation (12) is also investigated to evaluate the fidelity of adversarial images (show in Table 3). A larger $PSNR$ value indicates that the adversarial image more closely resembles its original image, meaning that the adversarial image has better imperceptibility. If the value of PSNR is large, it indicates that the noise or distortion due to the adversary is very small. Minimum PSNR of 40-50 dB is advised by Chen and Ramabadran [32]:

$$PSNR = \frac{20log_{10}\max(x)}{\sqrt{\frac{1}{n}\sum_{n=1}^{N}(x-x')^2}}, \qquad (12)$$

where $x$ is original image, $x'$ is the transformed image.

Compared to GAN models such as DCGAN [33], WGAN [34], CGAN [22-24], the models only have one pair of $D$ and $G$. Therefore, it will be difficult to control what it will generate at the same time unsupervised learn the necessary characteristics. Training the GAN with multiple stages will improve the shortcomings of simple GAN models compared with previous works. According to the original GAN, the input of $D$ is not only $G(z)$ but also has a label $y$, the label $y$ usually has two Fake and Real. When training $D$, it will first assign the real data to Real and assign the data generated from $G$ to be Fake, and then try to trick $D$ by assigning the data generated from $G(z)$ to be Real. That is, there are two calls $G(z)$ one time to train $D$ and another time to fool $D$, to find loss for backpropagation to improve weight. Our model has further improved the input of $D$ will be $G\big(f(z)\big)$ and $\overline{w}$ is a matrix of vector match, it will be the label for training. We label 0 and 1, it will not know what features the model is learning in the data set. As for the label $\overline{w}$ we will control on a space that similar description sentences will be close to each other.

TABLE 4. STATISTICS FOR EVALUATION BENCHMARKS OF THE PROPOSED MODEL

| ***No of Text*** | $N_{in}$ | $T_s$ | $T_{avg}$ | ***PSNR*** |
|---|---|---|---|---|
| 1 | 1090 | 576.54 | 0.17 | 29.53 |
| 2 | 1100 | 513.27 | 0.15 | 31.01 |
| 3 | 1100 | 522.13 | 0.15 | 20.02 |
| 4 | 1100 | 505.24 | 0.15 | 29.51 |
| 5 | 1100 | 537.13 | 0.16 | 28.75 |
| 6 | 1090 | 539.81 | 0.16 | 29.25 |
| 7 | 1000 | 540.72 | 0.18 | 29.41 |
| **Average** | | **533.55** | **0.16** | **29.65** |

Our model will repeat this process many times to improve its predictability also known as backpropagation. By utilizing backpropagation, it is possible to calculate the derivative of the loss function concerning each weight in the neural network. When this process is repeated over and over, the derivative value becomes very small at the first layer of neurons, making it impossible to update the network weights. This phenomenon is also known as "Vanishing Gradient". Theoretically not using $F_0$, $F_1$, $F_2$ will also produce an image but only to a certain extent. $F_0$, $F_1$, $F_2$ are born to reduce the corresponding work load for $G_0$, $G_1$, $G_2$, it is a network Long short-term memory. The Long short-term memory overcomes many of the previous limitations of derivation suppression.

## IV. CONCLUSION

In this paper, the authors have successfully built a GAN model to generate adversarial images with the purpose of fooling the target

model. The paper also pointed out some problems that other studies have not addressed. From there, a method of image generation is proposed in two stages: rough generation and refinement. With this method, the authors found that the images after randomization carry the basic characteristics and components of the target object classification. The generated image can deceive the target model according to the classification that the authors desire.

However, the research only stopped testing on the CUB dataset to fool the target model with the bird classes. In the future, the authors hope to develop the model at a higher level with many stages greater than two stages. The authors want the proposed model to be trained on many different datasets with a large number of images, and experiment with the proposed model to generate random images with multiple subclasses on other target models.

## ACKNOWLEDGMENTS

Truong Phi Ho was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.TS.037.

## REFERENCES

[1] TR. S. S. Kumar, M. Nystrom, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, "Adversarial machine learningindustry perspectives," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 69–75.

[2] Luận, L. C., & Thiên, T. H. (2023). Tăng cường độ chính xác trong việc nhận diện đối tượng trên các thiết bị cạnh thông minh. *Journal of Science and Technology on Information Security*, 2(19), 29-38. https://doi.org/10.54654/isj.v2i19.948.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[6] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[7] C. Cortes, N. Lawarence, D. Lee, M. Sugiyama, and R. Garnett, "Advances in neural information processing systems 28," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems*, 2015.

[8] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[9] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[10] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019.

[11] Pham, V. H., To, T. N., & Duy, P. T. (2023). A method of generating mutated Windows malware to evade ensemble learning. *Journal of Science and Technology on Information Security*, 1(18), 47-60. https://doi.org/10.54654/isj.v1i18.906.

[12] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[13] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied Soft Computing*, vol. 93, p. 106384, 2020.

[14] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.

[15] Olaye, Iredia M.; Seixas, Azizi A. The Gap Between AI and Bedside: Participatory Workshop on the Barriers to the Integration, Translation, and Adoption of Digital Health Care and AI Startup Technology Into Clinical Practice. *Journal of Medical Internet Research*,

2023, 25: e32962.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[17] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *Advances in neural information processing systems*, vol. 33, pp. 1633–1645, 2020.

[18] T. Bai, J. Zhao, J. Zhu, S. Han, J. Chen, B. Li, and A. Kot, "Aigan: Attack-inspired generation of adversarial examples," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2543–2547.

[19] W. Zhang, "Generating adversarial examples in one shot with imageto-image translation gan," *IEEE Access*, vol. 7, pp. 151 103–151 119, 2019.

[20] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[21] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[22] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5706–5714, 2017.

[23] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[24] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," In *Advances in Neural Information*

[25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Neural image caption generation with visual attention," in *Proc. ICML*, 2015, pp. 2048–2057.

[26] R. Yu, F. Jin, Z. Qiao, Y. Yuan, and G. Wang, "Multi-scale image-text matching network for scene and spatio-temporal images," *Future Generation Computer Systems*, vol. 142, pp. 292–300, 2023.

[27] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[28] C. Fjellstrom, "Long short-term memory neural network for financial time series," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 3496–3504.

[29] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on adam," *Applied Sciences*, vol. 10, no. 3, p. 1073, 2020.

[30] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[31] Rahman, M. M., Biswas, A. A., Rajbongshi, A., and Majumder, A, "Recognition of local birds of Bangladesh using MobileNet and Inception-v3," in *International Journal of Advanced Computer Science and Applications*, 2020, 11.8.

[32] Sheng Zhong, Qing-yun Shi, and Min-Teh Cheng. 1994, "Adaptive hierarchical vector quantization for image coding," *Pattern recognition letters 15*, 1994, 1171–1175.

[33] Radford, Alec; Metz, Luke; Chintala, Soumith, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[34] Weng and Lilian, "From gan to wgan," *arXiv preprint arXiv:1904.08994*, 2019.

ABOUT THE AUTHORS

**Truong Phi Ho**

Workplace: Vietnam Academy of Cryptography Techniques.

Email: hotp.gvm@actvn.edu.vn

Education: Bachelor's degree in Information Technology at Military Technology Academy in 2017; Master's degree in Information Technology at Nha Trang University in 2020; PhD student in Information Security at Vietnam Academy of Cryptography Techniques in 2022.

Recent research interests: Adversarial attack and defense; Generative adversarial networks; Robustness of deep learning models.

Cơ quan làm việc: Học viện Kỹ thuật mật mã.

Email: hotp.gvm@actvn.edu.vn

Quá trình đào tạo: Cử nhân Công nghệ thông tin tại Học viện Kỹ thuật quân sự vào năm 2017; Thạc sĩ Công nghệ thông tin tại Đại học Nha Trang vào năm 2020; Hiện đang là Nghiên cứu sinh An toàn thông tin tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: Tấn công và phòng thủ đối nghịch; Mạng đối nghịch tạo sinh; Về mô hình học sâu.

**Pham Duy Trung**

Workplace: Vietnam Academy of Cryptography Techniques.

Email: trungpd@actvn.edu.vn

Education: Bachelor's degree in Information Technology at Hanoi University of Science and Technology in 2005; Master's degree in Information Technology and Systems at University of Canberra - Australia in 2012; PhD degree in Information Sciences and Engineering at University of Canberra - Australia in 2018.

Recent research interests: Privacy; Machine learning; Safe Machine learning.

Cơ quan làm việc: Học viện Kỹ thuật mật mã.

Email: trungpd@actvn.edu.vn

Quá trình đào tạo: Nhận bằng Cử nhân Công nghệ thông tin Đại học Bách Khoa Hà Nội vào năm 2005; Thạc sĩ Công nghệ thông tin và Hệ thống tại Đại học Canberra - Úc vào năm 2012; Tiến sĩ Khoa học và Kỹ thuật thông tin tại Đại học Canberra - Úc vào năm 2018.

Hướng nghiên cứu hiện nay: Sự riêng tư; Học máy; Học máy an toàn.

**Bui Thu Lam**

Workplace: Vietnam Academy of Cryptography Techniques.

Email: lam.bui@mta.edu.vn

Education: Bachelor's degree in Information Technology at Military Technology Academy - Vietnam in 1998; Master's degree in Information Technology at The University of New South Wales - Australia in 2002; PhD degree in Computer Science at The University of New South Wales - Australia in 2007, Assoc.Prof in Information Technology in 2012.

Recent research interests: Artificial intelligence; Machine learning; Evolutionary computation.

Cơ quan làm việc: Học viện Kỹ thuật mật mã

Email: lam.bui@mta.edu.vn

Quá trình đào tạo: Cử nhân Công nghệ thông tin tại Học viện Kỹ thuật quân sự vào năm 1998; Thạc sĩ Công nghệ thông tin tại Đại học New South Wales - Úc vào năm 2002; Tiến sĩ Khoa học máy tính tại Đại học New South Wales – Úc vào năm 2007; PGS. TS Công nghệ thông tin vào năm 2012.