A Comparative Study of Generative Adversarial Networks for Generating Car Damaged Images

Phyu Mar Kyu
King Mongkut's Institute of Technology Ladkrabang
Kuntpong Woraratpanya (kuntpong@it.kmitl.ac.th)
King Mongkut's Institute of Technology Ladkrabang

Research Article

Keywords: Cycle-Consistent Adversarial Networks (CycleGAN), Attention-Guided Generative Adversarial Networks (AttentionGAN), Quantitative GANs Metrics, vision transformer (ViT), Convolutional Neural Networks (CNNs)

Posted Date: January 27th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2509412/v1

Additional Declarations: No competing interests reported.

A Comparative Study of Generative Adversarial Networks for Generating Car Damaged Images

Phyu Mar Kyu[1] and Kuntpong Woraratpanya[†]

[1]School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand.
[*]School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand.

[*]Corresponding author(s). E-mail(s): kuntpong@it.kmitl.ac.th;
Contributing authors: 62606003@kmitl.ac.th;
[†]These authors contributed equally to this work.

Abstract

The deeper the deep learning (DL) models are, the more computational complexity in need of the vast amount of data training requires to get better performance with accurate results. Generative adversarial networks (GANs) have obtained tremendous attention from many researchers with their impressive generation of synthetic instance data via a few source data by alleviating the problems of data scarcity, insufficient data diversity, and producing only limited plausible alternative data using standard data augmentation techniques conforming to the art of low-data-driven DL training in both scratch models, and pre-trained model in a variety of image classification tasks. That is why to defeat the above-referred problems and a lack of publicly available high-quality car-damaged datasets in car damage analysis, we created a custom data with a framework including three different evaluation assessments to generate a synthesized car-damaged dataset as the comparative study of Cycle-Consistent Adversarial Networks (CycleGAN), and Attention-Guided Generative Adversarial Network (AttentionGAN) by transforming one domain to another with our custom car damaged-undamaged dataset. In addition to this, we evaluated our generated car-damaged images based on three different evaluation assessments: firstly using three quantitative GANs metrics such as Inception Score (IS), Frchet Inception

Distance (FID), and Kernel Inception Distance (KID); secondly creating a convolutional neural networks (CNNs) classifier to identify them into real or fake; and finally building a vision-transformer (ViT) classifier to analyze them into damaged or undamaged. After accomplishing our comparative analysis, we can prove that AttentionGAN is better performance than CycleGAN according to all our experimental results.

Keywords: Cycle-Consistent Adversarial Networks (CycleGAN), Attention-Guided Generative Adversarial Networks (AttentionGAN), Quantitative GANs Metrics, vision transformer (ViT), Convolutional Neural Networks (CNNs)

1 Introduction

DL techniques have become more feasible with the emergence of several state-of-the-art models of their excellent performances not only to address non-linear control problems but also to develop new scenarios through previous learning in a variety of image classification tasks [1] for helpful real-world applications of ML approaches. However, their wide perspectives of data-driven for their large complex model networks with millions of parameters require a huge amount of training data, which is directly related to their performances of data analysis and decision making.

Nowadays, many computer vision and ML researchers have still been challenged to address the low-data training for their specific DL models with their small custom datasets. Especially for car damage detection and classification tasks, the researchers have been facing the lack of data diversity and not having an openly obtainable large amount of car damage dataset [2–4]. Nonetheless, training DL models with small datasets is not enough for data-driven to get better performance with accurate results, since it requires an extremely large set of relevant datasets to determine the parameters, which can successfully help to learn and identify the correct weights of the networks by training the multiple forward and backward iterations of the model networks successfully. These demands happen together with the rise of DL approaches in almost every field of application targeting related to computer vision including image segmentation, semantic segmentation, instance segmentation, or scene understanding. Therefore, many researchers have used the common usage of various translations of standard data augmentation techniques [5] such as rotation, flipping, cropping, adding noises, etc., to solve these problems of data scarcity and insufficient data diversity by generating new data, which are nearly similar to the original one. In addition to the standard data augmentation strategies, they produce only limited plausible alternative data.

After emerging the above phenomenon, there is incredible progress in the arena of ML with deep generative models, which are made up of two components of G and D with adversarial loss, to learn any data and generate various kinds of artificial-realistic synthetic instance data from source data.

They can estimate the underlying statistical structure of highly dimensional signal/ image/ audio/ video processing and also apply in many generative tasks of image processing, signal processing, natural language processing, computer graphics, and computer vision such as unsupervised and self-supervised representation learning [6, 7], text to image synthesis [8], image-to-image translation [9], image segmentation [10], semantic segmentation [11], image super-resolution [12], image inpainting [13], saliency prediction [14], domain adaptation [15], image enhancement [16], style transfer and texture synthesis [17], 3D pose estimation [18], image/video colorization [19, 20], audio-visual emotion recognition [21], visual arts, music and text generation [22], etc. That is why GANs have become more promising and gained much attention to overthrowing these problems of data unavailability and limited plausible alternative data, but training with insufficient data also leads to overfitting the D that leaks to the G of its generated samples causing the divergence training process [23].

Among many generative tasks in GANs, we mainly focus on image-to-image translation tasks, which are the problem of mapping an image from a source domain to a target domain using paired or unpaired data to learn a parametric mapping between inputs and outputs by supervised or unsupervised methods. Isola, P. (2017) [24] proposed a supervised conditional Pix2PixGAN to break this problem with paired data to conduct precise one-side image translation. However, obtaining a large amount of paired training data and collecting them are usually unfeasible and prohibitively expensive. To overcome the unobtainable paired training data and laborers to collect them, Zhu, J.Y. (2017) [25] proposed unsupervised CycleGAN using the cycle-consistency loss to perform the unpaired image-to-image translation between two image-domain mappings based on unpaired data. Nonetheless, the effort of unpaired image-to-image translation has still been challenging in the unwanted translation parts changing affected by background samples, even using a localized loss like PatchGAN [26], as the network without explicit attention mechanism [27]. To tickle these problems with the attention mechanism, which has been widely adopted in image translation algorithms, e.g., applying DA-GAN as a deep attention encoder to discover the instance-level correspondences in [28], reducing the cross-modal heterogeneity and generating modality-invariant representations in [29], separating the instance and background by AGGAN [30], improving as a multi-instance transformation in InstaGAN [31], using the internal activation from D to guide the translation in SPA-GAN [32], optimizing over-the-distance comparisons between samples in ContrastGAN [33], training as an extra semantic information model with CycleGAN in [34, 35]. In consideration of the network capacity to defeat the aforementioned cases, Tang, H. (2021) [36] proposed unsupervised AttentionGAN with attention-guided image-to-image translation to identify the foreground of the target domain minimizing the background changes of the source domain achieving good results over GANimorph [37] and CycleGAN.

Now many numerous architectures of GANs are available to perform and the researchers have been passionate about not only a comparative study of different GANs models with metric evaluation methods but also their combinations with CNN classification models to classify the specific tasks in different application areas. As far as we know outperforms CycleGAN over other deep generative models by a comparative analysis in car domain adaption from day to night translating [38], medical image synthesis [39–41], and face-aging application [42]. Nevertheless, comparing GAN performances with the evaluation ways of their generated images for many specific tasks in different application areas has remained on the stage of the unfinished debate.

Regarding the artificial-realistic generated images as a real-world dataset is applied in many vision analysis tasks with different types of classical DL neural networks, e.g., a combination of multiple linear layers and nonlinear activations stacked of multi-layer perceptron (MLP) or the fully connected (FC) network. The Standard CNNs have been well-known with their impressive performance of convolutional layers, pooling layers, FC or dense layers with different sizes of filters, and kernels to keep the most important features in the feature maps and recognize each pixel of its represented class and feature [43], In spite of facing the case of a false positive prediction has often happened when large shape variations with small objects in the feature map grid of the standard CNNs architectures. The attention mechanism of the attention gate is first introduced to handle this problem by suppressing the activations of irrelevant background areas to avoid the excessive waste of computational resources [44]. As a potential alternative feature extractor of existing CNNs, a transformer with a self-attention mechanism can predict the pixels of the feature [45] and also achieve accurate results as CNNs accomplishment [46]. Another supervised vision transformer model is ViT [47], which directly applies a pure transformer to learn the relationships between elements of a sequence of 16 16 image patches to classify the whole input images achieving state-of-the-art performance on multiple image recognition benchmarks. Furthermore, a transformer has been overcoming a variety of other vision problems, including object detection [48], semantic segmentation [49], image processing [50], and video understanding [51]. Therefore, a transformer has gained a lot of attention by proposing transformer-based models for improving the exceptional performance of a wide range of visual tasks among researchers [52].

To the best of our knowledge, there is no comparative analysis between CycleGAN and AttentionGAN as our car-image-synthesis framework with three different evaluation assessments to create a realistic car-damaged dataset and apply it in two different types of image classification models of CNNs and ViT. Therefore, defeating the above-referred problems and not having an openly obtainable large enough car dataset for the low-data training based on DL approaches of scratch models, pre-trained models, and deep generative models, to study the comparative analysis of unpaired image-to-image translation GANs for generating new high-quality synthetic car images with different types of damaged samples, this paper focuses on three challenges: (i)

creating a custom real car damaged-undamaged dataset for both training of CycleGAN and AttentionGAN, (ii) leveraging their potential to translate car undamaged images into damaged ones by generating high-quality synthesis car images with different types of damaged samples, and (iii) evaluating their synthesized car-damaged images to become a realist car-damaged dataset for defeating the issue of a public unavailable car-damaged dataset.

To address and accomplish these challanges, we did four experimental tasks: (i) to generate realistic car-damaged images using CycleGAN and AttentionGAN, (ii) to evaluate them utilizing traditional quantitative GANs metrics such as IS, FID and KID, (iii) to classify them as a binary classification of real or fake by a CNNs model, and (iv) to identify them as damaged or undamaged via a ViT model as car image synthesis. For all tasks, we created a custom car unpaired dataset of car damaged-undamaged dataset based on 3386 car images and proposed a framework to study the comparison of CycleGAN and AttentionGAN to generate 256256 resolution of car images with different types of damaged samples while using their specific performances of Gs and Ds techniques with a significantly limited amount of our custom unpaired car training dataset, which belongs to two classes of damaged and undamaged, to assess the performance of those models based on the qualities and diversities of their abilities of producing tenable photo-realistic car-damaged images depended on three different evaluation assessments: firstly using the existing GANs metrics such as IS [53], FID[53, 54], and KID [55] since evaluating GANs models based on their generated synthetic images unlikely other DL models, the training in both G and D with a loss function together to maintain an equilibrium until convergence, learning to classify images as real or fake by D without objectively assess both progress of the training and the relative quality of the model from their losses; secondly creating our proposed CNNs classifier, which was a sequential CNNs model [43, 56] constructed of the 2D convolutional layers with ReLU, max-pooling layers with stride, flattened layer, and two dense layers with ReLU, and Sigmoid, to classify them into real or fake; and finally building a ViT classifier [47], to analyze them into damaged or undamaged with their distinct accuracy values.

After finishing a comparative analysis of those two GANs with our three different evaluation approaches of three quantitative metrics and two classifiers, all approaches of generating photo-realistic car images with damaged samples effectively address the problem of lack of car damage data leading to different data distributions of damage type, location, and severity transforming from the original car-damaged images. In addition to this, to be explained in detail information of generated car-damaged images, all our experimental results described that those two models could be generated superior quality of both minor and moderate damage levels of the scratch, car paint, dent, and even fire-burning damaged samples and effectively applied for car damage classification with our two classifiers in this paper. Finally, we organize the rest of the paper as follows: Section 2 provides a review of the related literature works; we describe the materials, methods, and evaluations in Section

3; we explain about the detailed implementations of our experiments of two GANs, and two classifiers in section 4; the experimental results are reported and discussed in section 5; we conclude the paper in our last section 6.

2 Related Work

In today's world, the deeper the DL models, the more complex computation in need of the larger amount of training data is required to get better performance with more accurate results, the more capable DL models in almost all stages of image classification tasks [1] with no doubt. Especially in the car damage detection and classification task in [2–4], the researchers presented pre-trained CNNs models based on transfer learning (TL) to classify and detect every damaged part of an input car image by using standard data augmentation techniques [5] to artificially expand and adapt their small custom car damaged datasets to improve their performance and decrease their tolerance to the overfitting issue during training. However, even using the advantage of traditional data augmentation techniques and TL in their models, they still faced the overfitting problem and reduced the ability of the model generalization because of the lack of data scarcity and insufficient data diversity with a large amount of high-quality training data.

GANs have been promised to tickle these problems and generated various kinds of artificial-realistic synthetic instance data using existing image data as another advancement for data augmentation and image enhancement in DL and effectively applied in many application areas of generative tasks in image processing, signal processing, natural language processing, computer graphics, and computer vision. Among them, the image-to-image translation technique with supervised conditional adversarial nets (Pix2PixGANs) [24] is a general-purpose solution to break the problem of translating data from one possible data representation to another with the intrinsic source data content preserved and the extrinsic target data style transferred, where data can be represented as a color image, a gradient field, an edge map, a semantic label map, etc, given sufficient training paired data. The problems of translating data will be directly related to specific application algorithms (e.g., Day Night, Semantic-Labels image, Edge-Map Photo, Grayscale Color), but the same setting focuses on learning and predicting the mapping between two different domains.

On the other hand, obtaining paired training data can be usually difficult and expensive. What is more, accessing the input-output pairs for graphics tasks like artistic stylization can be even more challenging to be gained since the desired output is extremely complicated, needing typical artistic authoring. To overcome the unobtainable paired training data, Zhu, J.Y. and Park, T. (2017)[25] proposed CycleGAN using cyclic losses, which encourage the translated domain to be faithfully reconstructed when mapped back to the original domain with the unpaired image-to-image translation, which mainly relies on shared latent space and the assumption of cycle consistency loss, by forcing the translated images to fool the Ds using a classical adversarial loss

and translating those images back to the original input images, helped to build a reasonable mapping and generated reliable high-quality results since standard procedures often lead to the mode collapse, where all input images map to the same output image are difficult to optimize making progress fails, applied in a wide range of applications: collection style transfer; object transfiguration; season transfer; day to nigh transfer; photo enhancement; and learning mapping without considering any paired input-output samples.

Typically in the unsupervised case, when images are not paired or aligned, the network must learn which image parts of the samples are intended to be translated or not, e.g. some background regions might be taken into transformation by mistake. Furthermore, the effort of unpaired image-to-image translation has still been challenging in changing the unwanted translation parts that can also be easily affected by background changes, even using a localized loss like PatchGAN [26], as the network without explicit attention mechanism [27], which can solve image-to-image translation problems as bias guidance to the allocation of available resources into informative components by allowing the modeling of global dependencies without regarding their distances between input and output based on two sub-layers: multi-head self-attention mechanism; position-wise fully connected feed-forward network, as a new encoder-decoder translation. Moreover, it can be divided into two categories like post hoc network analysis, which predominantly employs access network reasoning for the visual object recognition task, and trainable attention module with two man sub-categories: stochastic (hard) that needs reinforcement training; deterministic (hard) that can be end-to-end training [57], e.g., applying DA-GAN as a deep attention encoder to discover the instance-level correspondences performance on only object translation in [28] and adding a dual attention mechanism to reduce the cross-modal heterogeneity and generate modality-invariant representations in [29], separating the instance and background by AGGAN [30], improving as a multi-instance transformation using the attention mask from an auxiliary network in InstaGAN [31], using the internal activation from D to guide the translation in SPA-GAN [32]. What is more, ContrastGAN [33] uses the image background and object segmentation masks to optimize over-the-distance comparisons between samples as a guide to the generation by cropping the unwanted parts of the image based on the masks, but collecting the training data with object segmentation masks is hard to take back and also in [34, 35] training as an extra semantic information of attention mechanism with CycleGAN to detects the object masks and employs them for the mask-guided generation. In consideration of the network capacity to defeat the aforementioned cases, Tang, H. [36] propose unsupervised AttentionGAN with attention guided image-to-image translation to identify the foreground of the target domain and minimize the background changes of the source domain with two proposed AGs and ADs by generating foreground attention masks and a background attention mask and achieving good results over GANimorph [37] and CycleGAN.

Now the researchers have been interesting in not only a comparative study of different GANs with metric evaluation methods but also a combination of

them with CNN classification models to classify the specific tasks, especially in car generation and classification, Faster RCNN [58] and day to night transfer based on unsupervised CycleGAN were adopted as a cross domain car detection and generation in [38], in medical image synthesis: Sarv Ahrabi, S. in 2022 [39] presented the better performance results of CycleGAN by a comparative study among CycleGAN and BiGANs [59], as similarly outperforming results of CycleGAN as a comparative analysis among CycleGAN and GANs [60] in [40], and five different GANs such as CGAN [61], DCGAN [62], f-GAN[63], WGAN[64]; CycleGAN by comparison in [41], using CT images for COVID-19 detection with their classification accuracies and FID values, and in face-aging application, Sharma, N. in 2022 [42] observed that the overall performance of CycleGAN was better than AttentionGAN for face-recognizing with age progression by analyzing CycleGAN and AttentionGAN with CelebA-HQ (CelebFaces Attributes high-quality dataset) and FFHQ (Flickr Faces HQ). Nevertheless, comparing GAN performances with the evaluation ways of their generated images for many specific tasks in different application areas has remained on the stage of the unfinished debate.

To apply the generated images as a real-world dataset in a variety applications of computer vision, CNNs can be considered the fundamental component in any image analysis task of their impressive tremendous performance of convolutional layers and pooling layers with different sizes of filters, and kernels to keep the most important features in the feature maps as slightly shifting variant data process to determine the relevant feature probabilities in several FC layers, and training the last FC or dense layer to recognize each image pixel of its represented class and feature [43], but nowadays transformer with self-attention mechanism can be performed as a potential alternative feature extractor as CNNs by auto prediction of the pixels of the feature [51] and also achieve accurate results as CNNs accomplishment [52]. Although vision transformers have been accomplished as existing CNNs, they are still facing a lack of the ability to extract the local information with their potential of capturing long-rand dependencies between sequence of elements. To beat this enhancement of the locality, a supervised vision transformer of ViT [47], which directly applies a pure transformer and its pre-trained model with transfer learning and fine-tuning to learn the relationships between elements of a sequence of 16 16 image patches to classify the whole input image achieving state-of-the-art performance on multiple image recognition benchmarks.

3 Materials, Methods, and Evaluations

According to our three challenges: (i) creating a custom real car damaged-undamaged dataset for both training of CycleGAN and AttentionGAN, (ii) leveraging their potential to translate car undamaged images into damaged ones by generating high-quality synthesis car images with different types of damaged samples, and (iii) evaluating their synthesized car-damaged images to become a realist car-damaged dataset for defeating the issue of a

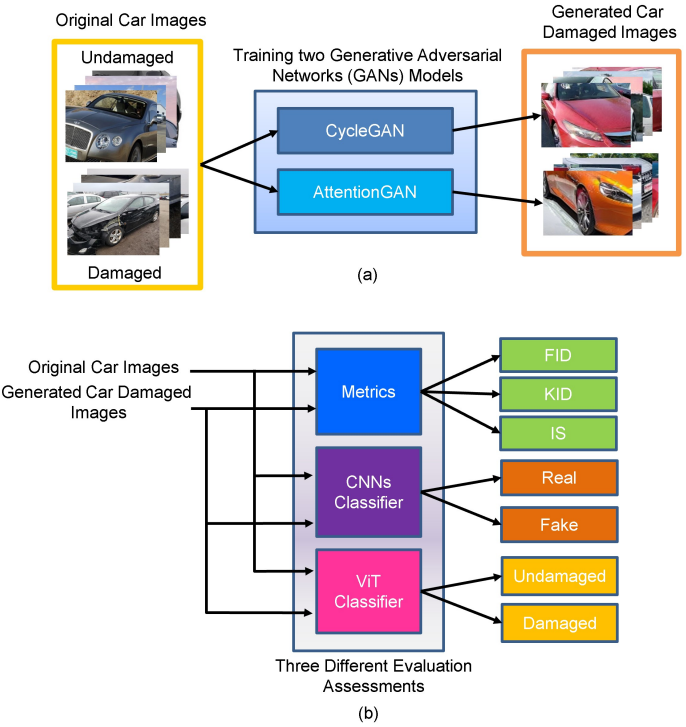


Fig. 1 Overall Framework for a Comparative Study of two Different GANs Models with two Different Datasets. (a) Training with CycleGAN and AttentionGAN for Synthesized Car Damaged Dataset (b) Synthesized Car Damaged Dataset Evaluation with three Different Assessments

public unavailable car-damaged dataset, we created a custom car damaged-undamaged dataset based on 3386 car images and a framework, which is shown in Figure 1 and operates with the following proposed materials, methods and evaluations.

3.1 Dataset

In this paper, we created our car unpaired dataset such as car undamaged and damaged datasets based on two sources: the car undamaged dataset from the Stanford cars dataset [65], which has 16185 mages with 196 classes, but we just used its 3386 car undamaged images; the car damaged dataset was congregated by the copart.com website, please visit this link (https://www.copart.com/), which is an online auto auction having many car images with their specific pieces of information, but we only applied its 3386 car-damaged images. What is more, we prepared our unpaired dataset into 512 512 pixels to be test our experiments by training and evaluating the proposed system, cropping the original dimension size of both damaged and undamaged images of 640 480 3 and 900 675 3, respectively.

More precisely, we rescaled every single image of our car unpaired datasets into 256 256 resolutions and randomly splitted them into the train and the test by 80-20% (2710-676) ratios training in both CycleGAN and Attention-GAN. Based on two datasets of original car images and generated car damaged sample of the fixed 256 256 RGB images with three different assessments, initially, we created three folders: real source, real target, and fake by 676, 2710, and 676 respectively to evaluate their generated car-damaged samples with three quantitative GANs metrics (IS, FID, and KID). finally, as a car image synthesis, we also randomly separated them into the train-test set as 80-20% of 700 images in the train set, and 176 images in the test set to train both two classifiers: CNNs classifier that model structure and its detail implementation will be described in the subsection belonging two classes such as real and fake; and ViT classifier, which will be briefly presented in the next section, to make a prediction of the generated images with two classes of damaged and undamaged.

3.2 Generative Adversarial Networks (GANs)

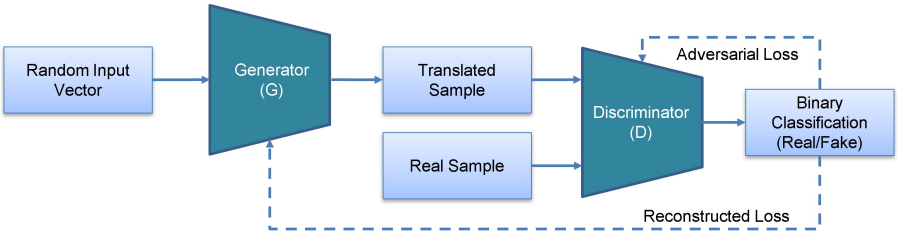


Fig. 2 Generative Adversarial Networks (GANs)

GANs have accomplished many application areas of generative tasks as one of the deep generative models. The initial main concept of GANs is a solution of zero-sum game to find a Nash equilibrium between two players or two neural networks (G and D), represented as a differentiable function to control a set of samples by each network with two losses as illustrated in Figure 2. The G trains to generate plausible fake data, and the D tries to discern them from real samples, training with the reconstructed loss and the adversarial loss that consists of generator and discriminator losses. In technical terms, the generator G receives a random input vector (z) from a prior noise distribution p_{noise} to learn G distribution over real data x and generated sample G(z), which is distinguishable the real sample p_{data}, thus making a fool to the D, which is a binary classifier to classify p_{data} as real, D(x)=1, and G(z) as fake, D(x)=0 as a pit competition against each other [9]. The objective of two players is well represented as a min-max optimization task as shown below:

$$min_G max_D \mathcal{L}_{GAN}(G,D) = \mathbb{E}_{z\sim p_{noise}(z)}[log(1-D(G(z))] + \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] \quad (1)$$

3.2.1 Image-to-Image Translation

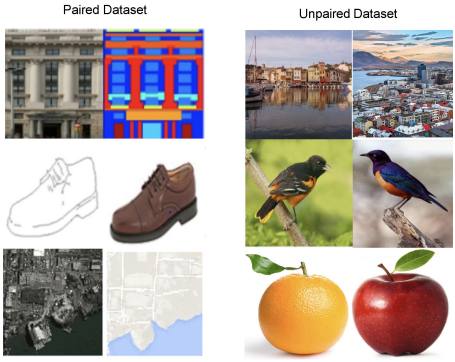


Fig. 3 Paired and Unparied Data

The image-to-image translation is adapting adversarial networks from an image generation to an image-transforming process from one domain to another domain by the intrinsic source content preserved and the extrinsic target style transferred to be applied in many applications of vision tasks as supervised, semi-supervised, and unsupervised representations with paired or unpaired data as a conditional GANs [58]. The example of paired and unpaired data is configured in Figure 3. In the image-to-image translation with paired data, the generator intends to map a source domain image $y \sim p_{source}$ into its corresponding ground truth target domain image $x \sim p_{target}$ via the mapping function of $G(z \mid y) = \hat{x} \sim p_{model}$ by generally viewed as a regression task between two domains that transfer the same underlying structures but differ in their surface appearances. In CGAN, the D classifies the concatenation of the source image y and its corresponding ground truth image x as real, $D(x, y) = 1$, while identifying y and the transformed image $\hat{x} = G(z)$ as fake, $D(\hat{x}, y) = 0$ with the random noise vector z and adversarial loss function in equation (2), however, the pixel-wise regression L_1 loss is added between the ground truth and the translated images when the output images shared unlikely the desired ground truth image of its global structure in equation (3).

$$\begin{aligned} min_G max_D \mathcal{L}_{CGAN}(G, D) = \mathbb{E}_{z \sim p_{noise}(z)}[log(1 - D(G(y, z), y))] \\ + \mathbb{E}_{x \sim p_{data}(x)}[logD(x, y)] \end{aligned} \tag{2}$$

$$\mathcal{L} = min_G max_D \mathcal{L}_{CGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G) \tag{3}$$

Moreover, two categories are divided in the supervised with paired training data as methods with single-modal and multi-modal outputs [9]. In the single-modal output, the extensive effort of collecting paired data are availabe in [66–73] using Hertzmann et al.'s image analogies on a non-parametric texture model [74], applying a conditional framework of CGAN on Pix2PixGAN and

Pix2PixHD for high-resolution image generation task[24, 75] learn a translation mapping using CNNs in most other related tasks, unfortunately, few or lack of paired training data has been still challenging to solve.

3.2.2 Unparied Image-to-Image Translation

The unpaired image-to-image translation is to overcome the challenge of paired data with an unsupervised scenario where no paired information is characterized. In the unsupervised-unpairing translation of single modal output, various methods have been presented to solve the supervised-pairing and non-identifiability problem as additional regulations including weight-coupling [76, 77], cycle-consistency with four categories: translation using a cycle-consistency constraint, translation beyond a cycle-consistency constraint, translation of fine-grained objects, and translation by combining knowledge in other fields [78, 79], forcing the generator to identity function [80], and a combination of them [81–83]. The cycle consistency loss is computed as the reconstruction error. Expect CycleGAN [25], many other GAN models have been tackling the cross-domain problem, although they can be easily affected by unwanted content and difficulty to focus on the semantic part of images while the translation takes place.

3.2.3 Attention Guided Image-to-Image Translation

A Transformer with self attention mechanisms have been introduced as depth estimation in image-to-image translation [84] to fix the mention limitation by unsupervisely focusing on the relevant input portion for the region of interest into two ways: using an extra supporting data as applying object mask annotations of ContrastGAN [33], utilizing object segmentation mask of InstaGAN [31] in the first way; training extra segmentation or attention model in [34, 35, 84]. All mention models are facing the problem of increasing the number of parameters, training time, and storage space, AttentionGAN [36] can fix these limitations and disentangle the input image into foreground and background by generating multiple attention masks and content masks with them.

3.2.4 Cycle-Consistent Adversarial Networks (CycleGAN)

A four-model composing CycleGAN, which is proposed by Zhu, J.Y. (2017) [25], two Gs (G_{XY}, G_{YX}) and two Ds (D_X, D_Y), converts input images from domain X into domain Y, and domain Y into domain X by G_{XY}, and G_{YX} respectively, without requiring a paired-image dataset to train the Gs. After finishing the training of two Gs, the remaining two Ds will be established training to determine the generated images as the process of convincing with D_X, which identifies the difference between real images from domain X given training samples $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and generated fake images from the G_{YX}, and D_Y, which is conversely the process of D_X by analyzing real images from domain Y given training samples $\{y_j\}_{j=1}^{M}$ where $y_j \in Y$ and generating

fake images from G_{XY}, similarly vice versa. To accomplish the above process by learning the two mapping functions (G, F), G: X → Y (from domain X to domain Y) and F: Y → X (from domain Y to domain X), the main two cycle-consistency losses prevent the learned mappings G and F from contradicting each other, help to get intuitively back the original real image when a translation happens from one domain to another with associated two adversarial D_X and D_Y for matching the distribution of generated images to the data distribution in the target domain by forward cycle-consistency loss: x → G(x) → F(G(x)) ≈ x, and backward cycle-consistency loss: y → F(y) → G(F(y)) ≈ y.

The adversarial loss and cycle consistency loss are mathematically shown in equation (4) and equation (5). Their full objective function of the combination of both adversarial loss and cycle consistency loss is expressed in equations (6) and (7). For the mapping functions G: X → Y and its D_Y, the equation of the adversarial loss is as follows.

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{x\sim p_{data}(x)}[log(1 - D_Y(G(x))] \\ + \mathbb{E}_{y\sim p_{data}(y)}[logD_Y(y)]\end{aligned} \tag{4}$$

Where G tries to generate images G(x) that look like samples from domain Y while D(y) translates between samples G(x) and real samples y. G tries to decrease this objective versus D goes to increase it, i.e., $min_G max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$, as the similar pattern as the mapping functions F: Y → X and its D_X as well: i.e., $min_G max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$. After expressing the main objective equation of adversarial loss, the other main objective cyclic losses of their forward and backward cycle-consistency losses are defined as:

$$\begin{aligned}\mathcal{L}_{cycle}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}[\| F(G(x)) - x \|_1] \\ + \mathbb{E}_{y\sim p_{data}(y)}[\| G(F(y)) - y \|_1]\end{aligned} \tag{5}$$

The full combination of both two cyclic losses and adversarial losses is prescribed by equations (4) and (5), where λ controls the relative significance of the two objectives.

$$\begin{aligned}\mathcal{L}_{G,F,D_X,D_Y} = \mathcal{L}_{GAN}(G, D_Y, X, Y) \\ + \mathcal{L}_{GAN}(F, D_X, Y, X) \\ + \lambda\mathcal{L}_{cycle}(G, F)\end{aligned} \tag{6}$$

$$G^*, F^* = argmin_{G,F} max_{D_x,D_Y} \mathcal{L}_{G,F,D_X,D_Y} \tag{7}$$

3.2.5 Attention-Guided Generative Adversarial Networks (AttentionGAN)

A four-model composing AttentionGAN is designed by Tang, H. (2021) [36] with two attention-guided generators (AGs) and two attention-guided discriminators (ADs) to learn two mappings between domain X given training

samples $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and domain Y given training samples $\{y_j\}_{j=1}^{M}$ where $y_j \in Y$ converting them into domain X $\rightarrow$ Y and domain Y $\rightarrow$ X without requiring a paired-image dataset, i.e., G: x $\rightarrow [A_y, C_y] \rightarrow$ G(x) and F: y $\rightarrow [A_x, C_x] \rightarrow$F(y), where A_x and A_y are the attention masks of images x and y to define each pixel intensity for the content masks C_x and C_y; G(x) and F(y) are the generated images, for matching the distribution of generated images to the data distribution in the target domain by using the cycle consistency loss: for each image x in domain X, i.e., x $\rightarrow G(x) \rightarrow F(G(x)) \approx x$; for each image y in domain Y, i.e., y $\rightarrow F(y) \rightarrow G(F(y)) \approx y$. The attention mask generator G_A targets to generate both n-1 foreground attention masks $\{A_y^f\}_{f=1}^{n-1}$ and one background attention mask $\{A_y^b\}$ to simultaneously learn the foreground and preserve the background of the input images and the feature map m extracted from the parameter-sharing encoder G_E that aims to extract both low-level and high-level deep feature representations is fed into the content mask generator G_C helping to produce the n-1 content masks $\{C_y^f\}_{f=1}^{n-1}$ followed by a Tanh(.) activation function and a channel-wise Softmax function Softmax(.) in equations (8) and (9) as follow.

$$C_y^f = Tanh(mW_C^f + b_C^f), for f = 1, ..., n-1 \tag{8}$$

$$A_y^f = Softmax(mW_A^f + b_A^f), for f = 1, ..., n \tag{9}$$

Where m feature map is fed into filter groups of $\{W_C^f, b_C^f\}_{f=1}^{n}$ and $\{W_A^f, b_A^f\}_{f=1}^{n}$ to generate the corresponding the n-1 content masks and the n attention masks respectively. Then $\{A_y^f\}_{f=1}^{n}$ splits into the n-1 foreground attention masks $\{A_y^f\}_{f=1}^{n-1}$ and one background attention mask A_y^b to flexible learn and translate the foreground content along the channel dimension. Finally, the attention masks are multiplied by the content masks to get the final target results as shown in the figure, and written mathematically as equation (10).

$$G(x) = \sum_{f=1}^{n-1}(C_y^f * A_y^f) + x * A_y^b \tag{10}$$

Where $x * A_y^b$ conserves the background of the input image x and $\sum_{f=1}^{n-1}(C_y^f * A_y^f)$ generates the foreground content for the input image. After combining them, we get the final target result G(x). In equation (11), the generator F has a similar structure as the generator G with three subnets: a parameter-sharing encoder F_E; an attention mask generator F_A, which generate the n attention masks of both foreground and background (i.e., A_x^b and $\{A_x^f\}_{f=1}^{n-1}$; a content mask generator F_C, which supports to create n-1 content masks (i.e., $\{C_x^f\}_{f=1}^{n-1}$), to generate both attention mask A_x and content mask C_x by mixing them with G(x) to reconstruct the original input image x.

$$F(G(x)) = \sum_{f=1}^{n-1}(C_x^f * A_x^f) + G(x) * A_x^b \tag{11}$$

Where the resulting image of F(G(x)) should be closer to the original image x, likewise as G(F(y)) should be near to image y (i.e., $G(F(y)) = \sum_{f=1}^{n-1}(C_y^f * A_y^f) + F(y) * A_y^b$). The min-max game between the AD D_{YA} and the G works together expressed in equation (12) as:

$$\begin{aligned}\mathcal{L}_{AGAN}(G, D_{YA}) = \mathbb{E}_{x\sim p_{data}(x)}[log(1 - D_{YA}([A_y, G(x)])] \\ + \mathbb{E}_{y\sim p_{data}(y)}[logD_{YA}([A_y, y])]\end{aligned} \tag{12}$$

Where D_{YA} is an attention-guided discriminator, plans to distinguish between the generated image pairs $[A_y, G(x)]$ and the real image pairs $[A_y, y]$. In the same way, the another discriminator D_{XA} analyzes between the fake image pairs $[A_x, F(y)]$ and the real image pairs $[A_x, x]$ with $\mathcal{L}_{AGAN}(G, D_{YA})$. The optimization objective of AttentionGAN can be expressed as equation (13).

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda_{cycle} * \mathcal{L}_{cycle} + \lambda_{id} * \mathcal{L}_{id} \tag{13}$$

Where $\mathcal{L}_{GAN}, \mathcal{L}_{cycle}$ and $\mathcal{L}_{id}$ are GAN, cycle-consistency, and identity preserving loss, respectively. λ_{cycle} and λ_{id} are parameters to control each relation.

3.3 Generative Adversarial Networks (GANs) Metrics

3.3.1 Inception Score (IS)

Inception Score (IS) is an evaluation metric for the quality of GANs using a pre-trained Inception V3, which is already trained on the ImageNet dataset, to capture the properties of the desirable generated samples of highly classifiable and diverse w.r.t class labels by measuring the divergence of average KullbackLeibler ($\mathbb{KL}$) value between the class conditional label distribution probability $p(y \mid x)$ of a generated sample, that supposes to have low entropy for easily classifiable samples as better sample quality, and the marginal distribution probability $p(y)$, which expects to have high entropy if all classes are equally in the set of samples as high diversity, obtained from all the samples, as mathematically described in the following equation.

$$exp(\mathbb{E}_x[\mathbb{KL}(p(y \mid x) \parallel p(y))]) = exp(H(y) - \mathbb{E}_x[H(y \mid x)]) \tag{14}$$

where $p(y \mid x)$ denotes the class conditional label distribution for image x and H(x) represents entropy of variable x in $p(y) \approx 1/N \sum_{n=1}^{N} p(y \mid x_n = G(z_n))$. IS measures the lowest score as 1 while the highest score depends on the number of classes of the dataset as a prediction to the domain label of the

translated images, the higher IS score is, the better-translated performance by the GAN model as well as the diverse images [52, 53].

3.3.2 Frchet Inception Distance (FID)

Frchet Inception Distance (FID) is used to assess the quality of synthetic images by computing the mean and covariance of synthetic and real images as shown in equation (1). It visualizes an embedded layer that contains a set of synthetic images in the pre-trained Inception V3 and uses it as the continuous multivariate Gaussian.

$$FID(r,g) = \| \mu_r - \mu_g \|_2^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{(1/2)}) \tag{15}$$

where r and g shows real and synthetic images while $(\mu_r, \sum_r)$ and $(\mu_g, \sum_g)$ are the mean and covariance of real and synthetic data distributions. FID score is a distance measurement between real and synthetic images in GANs and its quality depends on the features given by the pre-trained Inception V3 model. A lower FID score means a smaller distance between real and synthetic data distributions. Unlike IS score, a lower FID score means a better performance [52–54].

3.3.3 Kernel Inception Distance (KID)

Kernel Inception Distance (KID) is a similar metric evaluation as FID by applying the features returned from the pre-trained inception V3 model representation and relaxing the strict Gaussian assumption helped to improve the performance of FID. It measures the skewness mean and variance between the vector representations using a polynomial kernel to correct the distributions as two values such as KID mean and KID variance values. If its score approaches 0.0, the given two sets of images are identical. The lower the KID scores are, the better Likely as FID [54].

3.4 Convolutional Neural Networks (CNNs) Classifier

Convolutional Neural Networks (CNNs) take an input image and remove all unnecessary background information, using a kernel that output is sometimes referred to as the feature map to find the most important features for classification. Convolutional and pooling layers are responsible for the feature extraction. The convolutional layer is composed of a set of convolutional kernels where each neuron acts as a kernel but, the convolution operation becomes a correlation operation if the kernel is symmetric. The convolutional kernels divide the image into small slices known as receptive felds, helping to extract the feature motifs. Convolution operation and pooling operation can

be expressed as following equations (16) and (17):

$$f_l^k(p,q) = \sum_c \sum_{x,y} i_c(x,y).e_l^k(u,v) \tag{16}$$

Where $i_c(x,y)$ is an element of the input image tensor I_C , which is multiplied by the kth index convolutional kernel k_l of the lth as $e_l^k(u,v)$, where the output feature map of the kth convolutional operation can be expressed as $F_l^k = [f_l^k(1,1), ..., f_l^k(p,q), ..., f_l^k(P,Q)]$

$$Z_l^k = g_p(F_l^k) \tag{17}$$

Where Z_l^k denotes the pooled feature map of the lth for kth input feature map of F_l^k, $g_p(.)$ defines the type of pooling operation. There are different types of pooling operations such as Max, Average, L2, Overlapping, Spatial pyramid pooling, etc. For the Max pooling, it reports the maximum output within a rectangular neighborhood. The activation function serves as a decision function helping to learn the intricate patterns for a convolved feature map is described as an equation (18):

$$T_l^k = g_a(F_l^k) \tag{18}$$

where F_l^k defines a convolutional output assigned to activation function $g_a(.)$ adding non-lineraity and returning as a transformed output for the lth layer as T_l^k. There are different activation functions such as Sigmoid, tanh, maxout, SWISH, ReLU, etc. Among them, ReLU helps in overcoming the vanishing gradient problem and Sigmoid takes any real value as input and outputs values in the range of 0 to 1. Flattening layer is used to convert all the resultant 2-Dimensional arrays from the pooled feature maps into a single long continuous linear vector. The fully connected layer is mostly used at the last layer of the neural netowrk for classification [43, 56].

3.5 Vision Transformer (ViT) Classifier)

The vision transformer (ViT) [46] is proposed to reshape the image of 1D sequence of token embeddings $x \in \mathbb{R}^{H \times W \times C}$ into the image of 2D sequence of flattend patches $x_p \in \mathbb{R}^{N \times (P^2 C)}$ to handle 2D images, where C is the number of channels, (H,W) is the resolution of the original image, (P,P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches served as the effective input sequence length for the transformer. Flatten the patches and mapping them to D dimensions with a trainable linear equation 12 when transformer uses constant widths of latent vector size D in all layers, as the output of patch embeddings. 1D position embeddings is utilized to maintain the position information in the patch embeddings rather observing to apply more advanced 2D position embeddings. Likewise as BERT's [class] token, a learnable embedding to the sequence of embedded patches ($z_0^0 = x_{class}$) assert at the output encoder z_L^0 set out as the image representation $y = LN(z_L^0)$, where L is the last layer, the output y is the 0 index token of z wrapped in a

LayerNorm layer (LN), and z_L^0 joins a classification head, which is assigned by a Multilayer Perception (MLP) with one hidden layer at pre-training striving time and a single linear layer at fine-tuning obligation time. Both multiheaded self-attention (MSA) and MLP blocks are served as the transformer encoder described them as the technical terms of equations 20 and 21, where a LN should be imposed before every block and residual connections after every block.

$$z_o = [x_{class};\ x_p^1\mathtt{E};\ x_p^2\mathtt{E};\ ...;\ x_p^N\mathtt{E}] + \mathtt{E}_{pos} \tag{19}$$

$$z\prime_l = \mathtt{MSA}(\mathtt{LN}(z_{l-1})) + z_(l-1) \tag{20}$$

$$z_l = \mathtt{MLP}(\mathtt{LN}(z\prime_l) + z\prime_l \tag{21}$$

In Equation 19, $\mathtt{E} \in \mathbb{R}^{(P^2.C)\times D}$ and $\mathtt{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$, where E and $E_{pos}$are for the patch and position embeddings of the input image with the output of the initial patch embedding layer of z_0, a particular z layer of its prime value or an intermediary value of $z\prime_l$, and a particular layer of z_l. In equations 17 and 18, $l = 1...L$ means every layer l passing trough from 1 to L (the total number of layers), and there is also MSA and MLP wrapping LN.

4 Experiments

4.1 Experimental Setup

In this section, we explain our experimental tasks which are divided into four sections:(i) to generate realistic car-damaged images using CycleGAN and AttentionGAN, (ii) to evaluate them utilizing traditional quantitative GANs metrics such as IS, FID and KID, (iii) to classify them as a binary classification of real or fake by a CNNs model, and (iv) to identify them as damaged or undamaged via a ViT model as car image synthesis. Responding to our three challenges with four tasks, we carried out our experiments depending on our proposed framework in two different workstations, the experiment of our all implementations platforms are in the following ways: the first station of using our Lab computer with the Graphics Processing Unit (GPU) on Ubuntu to assemble our CycleGAN, AttentionGAN, and evaluation GANs metrics; the second station of utilizing Google Colab Pro platform, supporting NVIDIA System Management Interface driver version of 460.32.03, and Cuda version of 11.2 to train our CNNs Classifer and ViT Classifier on Tesla T4 GPU with 25GB RAM. To sum up, all detail model implementations and training process with parameter setting are described in following.

4.2 GANs Implementation Details

We implemented CycleGAN [25], and AttentionGAN [36] for car-damaged images generation based on the train datasets (trainX & trainY), and test datasets (testX & testY) building them in Pytorch on NVIDIA Geforce GTX 1070Ti with one GPU taking approximately three days for each model training per time. For training our car-damaged generation with CycleGAN and

AttentionGAN, we just only focus on a domain adaption of X Y (two-domain adaptions: X Y; Y X) to generate car-damaged images from car-undamaged data (X) into car-damaged data (Y). For a fair comparison, we utilized the same parameter setting in both models: a fixed learning rate of 0.0002 for 100 epochs; a batch size of 1; Gaussian distribution of weights from 0 to 0.02; image buffer of 50, Adam optimizer with the momentum $\beta_1 = 0.5$ and $\beta_2 = 0.999$, $\lambda_{cycle} = 10$, $\lambda_{gan} = 0.5$, $\lambda_{pixel} = 1$, $\lambda_{tv} = 1e-6$, $\lambda_{id} = 0.5$ by training with our train-test datasets, which have 2710 training images and 676 testing images, respectively, splitting 80%-20% ratio from 3386 images of car damaged-undamaged dataset.

4.2.1 CycleGAN for Car Damaged Images Generation

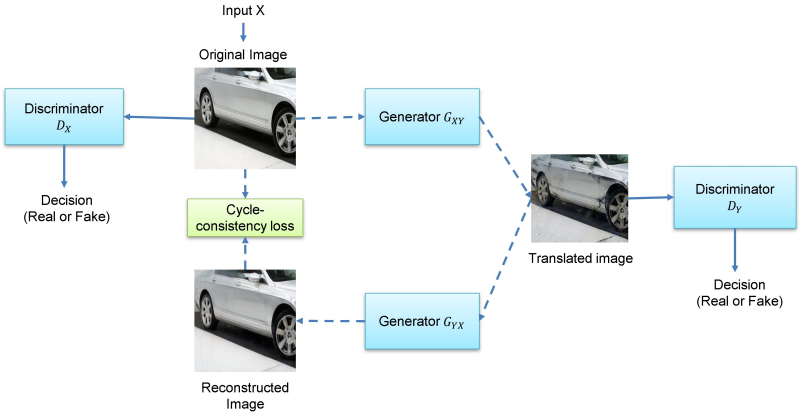


Fig. 4 CycleGAN for Car Damage Generation

The detailed architecture of generated car-damaged images of CycleGAN[25] that both Gs and Ds are the following descriptions. we adopted them from Zhu, J.Y. (2017) as one of our car-damaged image generators. In the generator models, the input of an image size of 256 256 RGB (resizing an original 512 512 3 image) as a down sample and up sample back to create generated car-damaged images by passing through a stack of 7 7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1 denoted as c7s1-k (C7S1-64 & C7S1-3), a stack of 3 3 Convolution-InstanceNorm-ReLU layer with k filters and stride 2 symbolized as C3S2-k (C3S2-128 & C3S2-256), a 9-residual-block stack of 3 3 Convolution-InstanceNorm-ReLU layer with fixed k filters indicated as RB-k (RB-256), and a stack of 3 3 fraction-stride-Convolution-InstanceNorm-ReLU layer with k filters and stride$\frac{1}{2}$ represented as C3S$\frac{1}{2} - k$ (C3S $\frac{1}{2} - 64$ & C3S$\frac{1}{2} - 128$)by their networks as C7S1-64, C3S2-128, C3S2-256, RB-256, RB-256, RB-256, RB-256, RB-256, RB-256, RB-256, RB-256, C3S$\frac{1}{2} - 128$, C3S$\frac{1}{2} - 64$, C7S1-3. The discriminator models, which aim to classify whether 70 70 overlapping area of input image patches

are real or fake requiring fewer parameters than a full-image discriminator able to work on arbitrarily sized images in a fully convolutional layer, allow input images size of 256x256 RGB into output image-tensor size of 30x30 to pass through a stack of 4 4 Convolution-InstanceNorm-LeakyReLU layer using a slope of 0.2 with k filters and stride 2 shown as C4S2-k (C4S2-64, C4S2-128, C4S2-256 & C4S2-512) structured by C7S2-64, C7S2-128, C7S2-256, C7S2-512. As an exception, there is no InstanceNorm in C4S2-64. After the last layer, a convolution layer is applied to get a 1-dimensional output. Finally, a cycle-consistency loss function forces the Gs to reduce the space between their possible mapping functions and minimizes the discrepancy between the original image and the reconstruction obtained from the translated samples of Gs as shown in Figure 4. We trained a scratch CycleGAN [25] with our train-test (2710-676) car damaged-undamaged dataset and the above-presented parameter setting. In cycleGAN, the total number of parameters of G_X and G_Y are 11.378 Million respectively; D_X and D_Y are 2.765 Million separately.

4.2.2 AttentionGAN for Car Damaged Images Generation

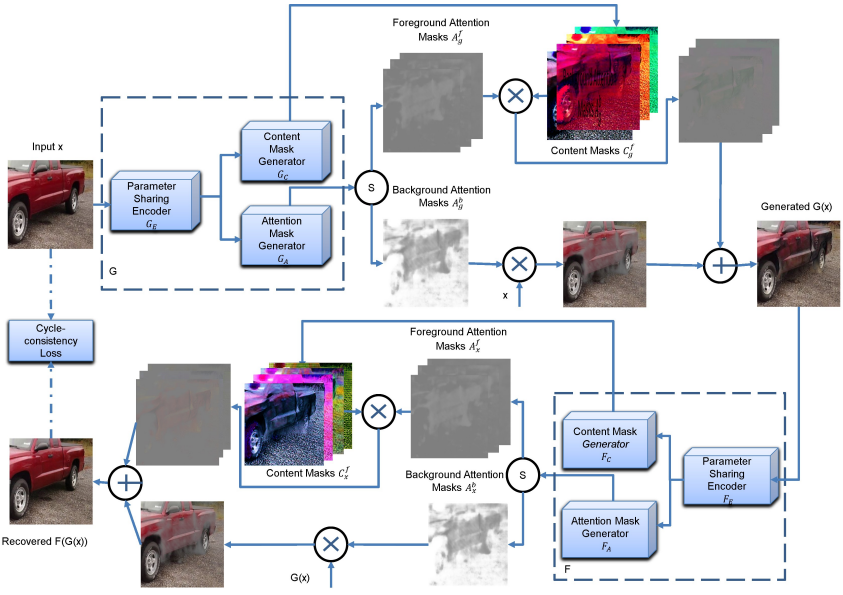


Fig. 5 AttentionGAN Framework [36] For Car Damage Generation

The detailed network architectures of car-damaged generation of AttentionGAN [36], defining the input (cropping an original 512 512 3 image into a three-channel 256 256 RGB image) and the outputs (n attention masks and n-1 content masks), the input images flow through the models of Gs, which is similar to the Gs of CycleGAN structured by C3S$\frac{1}{2}-128$, C3S$\frac{1}{2}-64$, C7S1-10,

where C3S$\frac{1}{2}-k$ (C3S$\frac{1}{2}-64$ & C3S$\frac{1}{2}-128$) is a stack of 3 3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters and stride $\frac{1}{2}$; a stack of 7 7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1 represents c7s1-k (C7S1-10), and the models of Ds, which is same as the vanilla Ds of CycleGAN taken an image as their input. In addition to the outputs of n attention masks and n-1 content masks, the first mask is used as the background attention mask and the remaining masks are utilized as the foreground attention masks, setting the n value as 10 in our experiments. Figure 5 shows the overall architecture of generated car-damaged iareass of AttentionGAN. We trained a scratch AttentionGAN [36] with our train-test (2710-676) car damaged-undamaged dataset and the above-presented parameter setting. In AttentionGAN, the total number of parameters of G_X and G_Y are 11.823 Million respectively; D_X and D_Y are 2.765 Million each in order.

4.3 CNNs Classifier Implementation Details

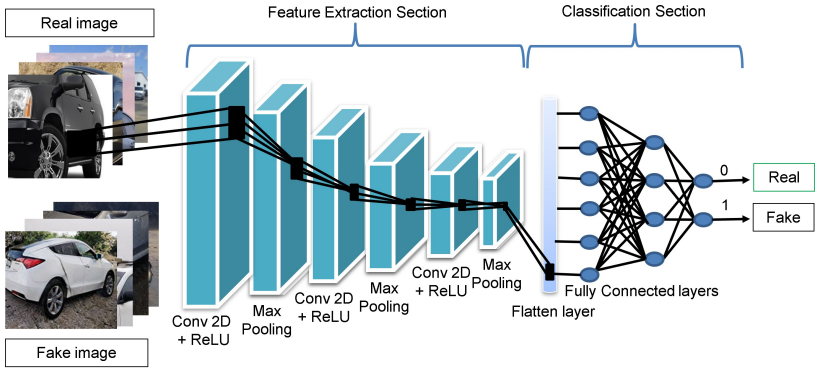


Fig. 6 Convolutional Neural Networks (CNNs) Classifier

In Figure 6, we defined the architecture of our simple sequential CNNs Classifier that it was composed of the three-2D-convolution layers with an activation function of a rectified linear unit (ReLU), using filters with very small 3 3 receptive fields, performing 1 1 stride over three Max pooling layers, a flattened layer, and two dense layers in the company of the activation functions of RelU and Sigmoid respectively, applying as a checker to classify our generated images of CycleGAN and AttentionGAN in TensorFlow and Keras. In our CNNs Classifier, we used a loss function of the binary cross-entropy, and adaptive moment estimation (Adam) as an optimizer. What is more, we allowed our input images to pass through the 2D convolution layers and the Max pooling layers by three times alternatively. After that, they were entered the flattened layer, and the last two dense layers to finish the process of classification by the training of 100 epochs with a batch size of 32. In addition to this, we expained the detail implementations of our CNNs Classifier as

follows. As a summarizing of our CNNs Checker, its layers and parameters are shown in Table 1.

- Firstly, our fixed image dimension of 256 256 3 passed through the stack of three 2D-convolution layers with ReLU, where we used the 16-32-16 feature kernel filters with very small 3 3 receptive fields. We used 1 1 stride, which is performed over three Max pooling layers of 127 127 16, 62 62 32, and 30 30 16. Dimensions of the feature maps after performing through two and three 2D-convolution layers, we reduce the input image size from 256 256 3 to 125 125 32 with the feature map of 125 125 and the depth or filter of 32, and then 60 60 16 in our last 2D convolutional layer.
- The flattened layer is added at our last convolutional layer to flatten our input images, using 14400 nodes.
- The last two dense layers are fully connected hidden layers. the first dense layer has 256 nodes with the RelU activation and the node of the last one is 1 with sigmoid activation.

Table 1 Detail Variants of CNNs Classifier

Layer (Type)	Output Shape	Parameters
Conv2D(ReLU)	(None, 254, 254, 16)	448
MaxPooling2D	(None, 127, 127, 16)	0
Conv2D(ReLU)	(None, 125, 125, 32)	4640
MaxPooling2D	(None, 62, 62, 32)	0
Conv2D(ReLU)	(None, 60, 60, 16)	4624
MaxPooling2D	(None, 30, 30, 16)	0
Flatten	(None, 14400)	0
Dense(ReLU)	(None, 256)	3686656
Dense(Sigmoid)	(None, 1)	257
Total Parameters: 3696625		

4.4 ViT Classifier Implementation Details

Table 2 Detail Model Variants between Scratch and Pre-trained ViT Models

Model	Total Para	Train Para	Non-Train Para	For/Back Size	Total Size
Scratch ViT	85,800,963	85,800,963	0	3292.20 MB	3540.67 MB
Pre-trained ViT	85,800,963	1538	85,798,656	3330.74 MB	3579.20 MB

Note: Both models' parameters size, input size, and total mult-adds are 229.20MB, 19.27 MB, and 5.52 GB, respectively.

We implemented both the scratch ViT model and the pre-trained ViT model [47] as shown in Figure 7, which is already trained on the ImageNet

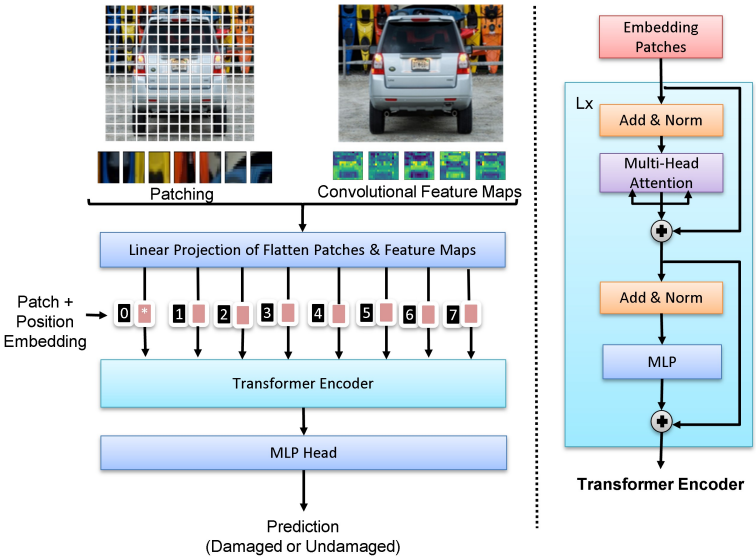


Fig. 7 Visual Transformer (ViT) Classifier

dataset and applying transfer learning with fine-tuning techniques) for identifying as a binary classification with our custom real dataset and generated fake damaged datasets from CycleGAN and AttentionGAN. We randomly separated our datasets into a train-test set with an 80-20% ratio of 700 images for training and 176 images for testing based on three datasets with two classes, which belong to damaged and undamaged. After that we resized input images into 224 224 3, and set up the parameter as Adam optimizer with the momentum $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a batch size of 32 instead of the referred batch sizes of 4096 since we made an apparatus it on Google Colab pro+ that could not handle larger batch sizes, a patch size of 16, 197 of patches, embedding dimension of 769, a high weight decay of 0.1 by training with 100 epochs. Tables 3 and 4 summarized the details of scratch and pre-trained ViT models variants as a type of layer, input shape, output shape, parameters, and training set. A comparison of both models' divergences is described in Table 2. Our custom pre-trained ViT model of training parameters is 2,307 with model size of 327 MB compared to the 85,800,963 of trained parameters of the vanilla ViT model.

5 Experimental Results and Discussion

5.1 Experimental Results of CycleGAN and AttentionGAN

Within both training of CycleGAN and AttentionGAN with car-unpaired datasets (train X, train Y, test X, and test Y) for generating synthetic car damaged samples by focusing only on a domain adaption of X Y, where X

Table 3 Detail Variants of the Scratch ViT Model

Layer (Type)	Input Shape	Output Shape	Parameters	Trainable
ViT	[32, 2, 224, 224]	[32, 2]	152064	True
PatchEnbedding	[32, 2, 224, 224]	[32, 196, 768]	–	True
Conv2D(Patcher)	[32, 2, 224, 224]	[32, 768, 14, 14]	590592	True
Flatten	[32, 768, 14, 14]	[32, 768, 196]	–	–
EmbeddingDropout	[32, 197, 768]	[32, 197, 768]	–	–
Sequential(Layers)	[32, 197, 768]	[32, 197, 768]	–	True
EncoderLayer(0)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True
EncoderLayer(1)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True
EncoderLayer(2)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True
EncoderLayer(3)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True
EncoderLayer(4)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True
EncoderLayer(5)	[32, 197, 768]	[32, 197, 768]	–	True
MSABlock	[32, 197, 768]	[32, 197, 768]	2363904	True
MLPBlock	[32, 197, 768]	[32, 197, 768]	4723968	True

Layer (Type)	Input Shape	Output Shape	Parameters	Trainable
EncoderLayer(6) & [32, 197, 768] & [32, 197, 768] & – & True				
MSABlock & [32, 197, 768] & [32, 197, 768]	2363904	True		
MLPBlock & [32, 197, 768] & [32, 197, 768] & 4723968 & True				
EncoderLayer(7) & [32, 197, 768] & [32, 197, 768] & – & True				
MSABlock & [32, 197, 768] & [32, 197, 768] & 2363904 & True				
MLPBlock & [32, 197, 768] & [32, 197, 768] & 4723968 & True				
EncoderLayer(8) & [32, 197, 768] & [32, 197, 768] & – & True				
MSABlock & [32, 197, 768] & [32, 197, 768] & 2363904 & True				
MLPBlock & [32, 197, 768] & [32, 197, 768] & 4723968 & True				
EncoderLayer(9) & [32, 197, 768] & [32, 197, 768] & – & True				
MSABlock & [32, 197, 768] & [32, 197, 768] & 2363904 & True				
MLPBlock & [32, 197, 768] & [32, 197, 768]& 4723968	True			
EncoderLayer(10) & [32, 197, 768] & [32, 197, 768] & – & True				
MSABlock & [32, 197, 768] & [32, 197, 768] & 2363904 & True				
MLPBlock & [32, 197, 768] & [32, 197, 768] & 4723968 & True				
EncoderLayer(11)	[32, 197, 768]	[32, 197, 768] & – & True		
MSABlock & [32, 197, 768] & [32, 197, 768]	2363904 & True			
MLPBlock & [32, 197, 768] & [32, 197, 768] & 4723968 & True				
Sequential(Classifier) & [32, 768] & [32,2] & – & True				
LayerNorm(ln) & [32, 768] & [32,768] & 1536 & True				
Linear(Heads) & [32, 768] & [32,2] & 1538 & True				

Note: This table is the final part of Table 2.

Table 4 Details Variants of the Pre-Trained ViT Classifier

Layer (Type)	Input Shape	Output Shape	Parameters	Trainable
VisionTransformer	[32, 2, 224, 224]	[32, 2]	768	Partial
Conv2D(Patcher)	[32, 2, 224, 224]	[32, 768, 14, 14]	590592	False
Encoder	[32, 197, 768]	[32, 197, 768]	151296	False
Dropout	[32, 197, 768]	[32, 197, 768]	–	–
Sequential(Layers)	[32, 197, 768]	[32, 197, 768]	–	False
EncoderLayer(0)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(1)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(2)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(3)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(4)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(5)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(6)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(7)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(8)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(9)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(10)	[32, 197, 768]	[32, 197, 768]	7087872	False
EncoderLayer(11)	[32, 197, 768]	[32, 197, 768]	7087872	False
LayerNorm(ln)	[32, 197, 768]	[32, 197, 768]	1536	False
Linear(Heads)	[32, 768]	[32, 2]	1538	True

and Y represent car undamaged and damaged data, the training losses of all Gs and Ds have happened over the number of epochs as shown in Figure 8. According to monitoring the behavior of training losses of both Gs and Ds, the model can generate a high-realist synthesis car damaged samples when both losses of Gs and Ds have been remaining consistent throughout the training process happening. In addition to the initial state of training with smaller epochs, the losses of Gs should be larger than the loss of Ds, and this significantly decreases the generation performance of the blurry translated images. Therefore, we observe that the larger the epochs are, the lower the training losses and the more realistic-damaged features of car images outcome.

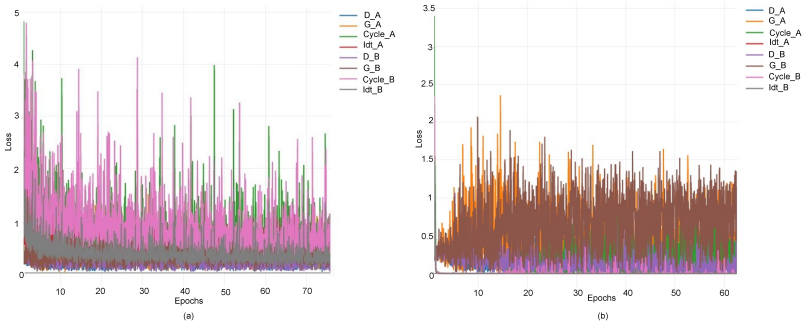


Fig. 8 Training Losses over Epochs for Synthesized Car Damaged Generation in (a) CycleGAN and (b) AttentionGAN

In our first experimental task of car damage generation with CycleGAN and AttentionGAN, both models can create synthesized realistic car images with different types of synthetic damaged samples with location and severity transforming from the original car undamaged images into damaged ones. Tables 5 and 7 depict the qualitative of some undamaged to damaged translation of the variety of damaged samples on the front, rear, and side of a car, as well as represent their damaged levels of generating minor and moderate severity in Tables 6 and 8 from CycleGAN and AttentionGAN, respectively. As we can see that there are some artifacts present in both models of the generated images, but the overall appearance of their samples looks realistic. Corresponding to the generated types of damaged samples, these models can generate not only car images with damaged samples of scratch, dent, car paint, and even fire-burning, but also samples with damaged levels of minor and moderate as high-quality car damaged samples. Nonetheless, the generated damaged samples with the dent and severe level seem to be artificial and blurry not looking like a photo-realistic of damaged samples of scratch, car paint, and fire-burning in both models of minor and moderate conditions.

As a comparative analysis between CycleGAN and Attention, the resolutions of individual outputs of the synthesized damaged features of AttentionGAN are higher than the CycleGAN outcomes shown in Table 9. Therefore,

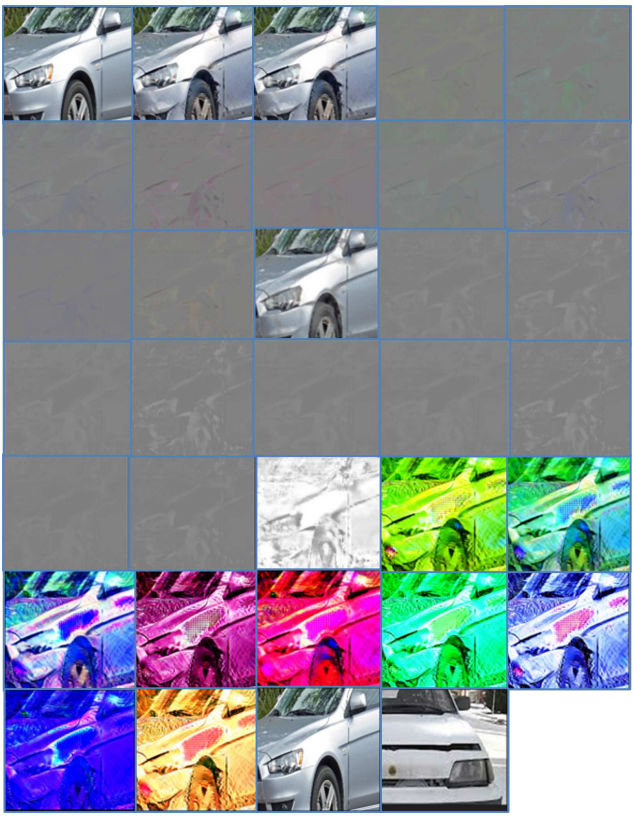


Fig. 9 Step by Step Generating Synthesized Car Damaged images with both attention masks and content masks in AttentionGAN from the leftmost of original car-undamaged images into the rightmost of the original car-damaged images. From the left to the right arrangement of images, we get all results of AttentiionGAN as real X, fake Y,.., idt Y, and real Y by adaption X $\rightarrow$ Y, where X is an original car-undamaged image and Y is an original car-damaged image.

AttentionGAN outperforms CycleGAN not only in the visual appeal of producing a sharper instance of synthetic car-damaged samples but also generating both attention and content masks, which are presented in Figure 9 by its main contribution of learning the foreground and preserving the background of input images all at once. What is more, depending on the amount of training dataset and time, the performance of both models can improve their capabilities further. On the other hand, we can utilize both models of synthesized car-damaged images not only in our second task of applying three evaluation metrics but also in our third and four tasks of the car image synthesis to defeat the problem of publicly unavailable car-damaged datasets.

Table 5 CycleGAN for Synthesized Car Damage Generation with Damaged Location

Location	Input	Generated	Input	Generated	Input	Generated
Front						
Rear						
Side						

Table 6 CycleGAN for Synthesized Car Damaged Generation with Damaged Severity

Damage Level	Input	Generated	Input	Generated
Minor				
				
Moderate				
				

Table 7 AttentionGAN for Synthesized Car Damaged Generation with Damaged Location

Location	Input	Generated	Input	Generated	Input	Generated
Front						
Rear						
Side						

Table 8 Attention for Synthesized Car Damaged Generation with Damaged Severity

Damage Level	Input	Generated	Input	Generated
Minor				
				
Moderate				
				

Table 9 Comparative Generated Synthesized Car Damaged Images of CyleGAN and AttentionGAN

Input	CycleGAN	Input	AttentionGAN

5.2 Experimental Results of the Evaluation GANs Metrics

When we evaluate both the visual quality and the diversity of generated images using the Inception score (IS), Frecht inception distance (FID), and Kernel Inception Distance (KID) recorded every ten tick intervals, which refers to the number of iterations after the training snapshot has been taken in both CycleGAN and AttentionGAN as our second experimental task, we achieve the scores of IS, FID, and KID such as 2.431073 ± 0.426682, 47.2298, and 1.625106 ± 0.103924 respectively in AttentionGAN. In CycleGAN, we accept the scores of 2.395367 ± 0.44129908, 49.3723, and 1.636226 ± 0.122677 for IS, FID, and KID, correspondingly summarized in Table 10 and displayed graphically in Figure 10. By a comparative analysis over our results, we observe that AttentionGAN gives the scores of the highest IS, the lowest FID, and KID, and then also generates better-quality synthetic car damaged samples than CycleGAN.

Table 10 Evaluation GANs Metrics Values of CycleGAN and AttentionGAN

Metrics	CycleGAN	AttentionGAN
IS	2.395367 ± 0.44129908	**2.431073 ± 0.426682**
FID	49.3723	**47.2298**
KID	1.636226 ± 0.122677	**1.625106 ± 0.103924**

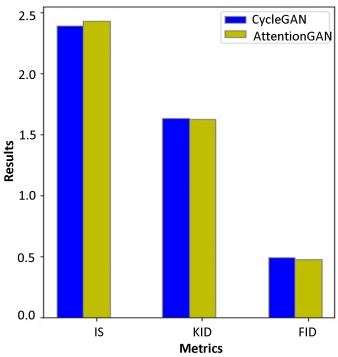


Fig. 10 Graphical Representation of Quantitative Metrics' Results with Image Quality Assessment Score over CycleGAN and AttentionGAN

5.3 Experimental Results of the CNNs Classifier

When we classify the quality of generated images from CycleGAN and AttentionGAN with a CNNs model as our third experimental task by training with two datasets with two classes including the original real car images and their

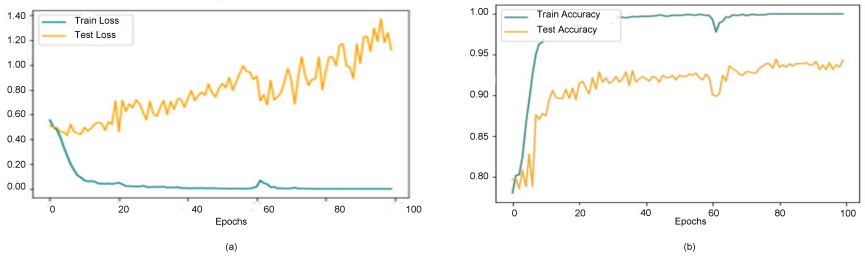
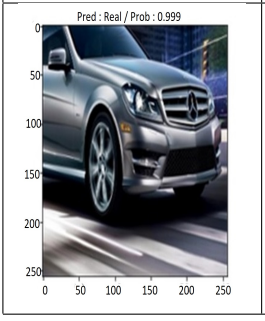
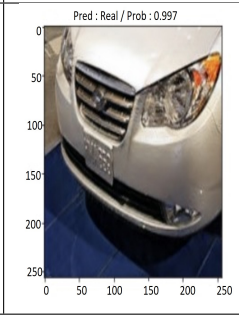
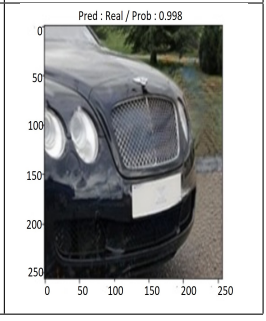


Fig. 11 The Plots of CNNs Classifier (a) Train-Test Loss over Epochs (b) Train-Test Accuracy over Epochs

generated fake car-damaged images, we receive the CNNs Classifier of its values of precision, recall and accuracy such as 93.9%, 95.7%, and 91.9%, respectively. Figure 11 shows the loss and accuracy plots for both train-test results of the CNNs Classifier. According to Table 11, we observe that the prediction accuracy of both real and fake from two GANs are slightly better than one another, however, we can make the CNNs Classifier fool with some instances of fake images of CycleGAN, unlike the wrong prediction of fake images of AttentionGAN into real often. To sum up, the more we can make fools of the CNNs model, the better synthetic images with good generative models are. Regarding the performance of making fools of a CNNs model, AttentionGAN is better than CycleGAN. In addition to their prediction accuracy, the quality of generated fake images is as similar to the original real images. This depicts that they can be applied as a real-world car-damaged dataset in car image synthesis tasks as our aim.

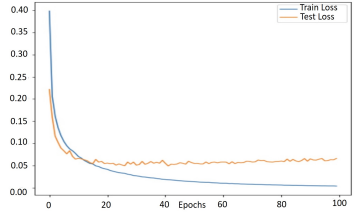
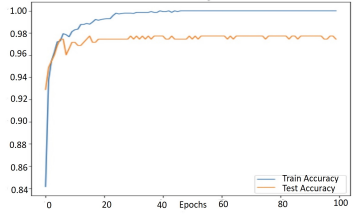
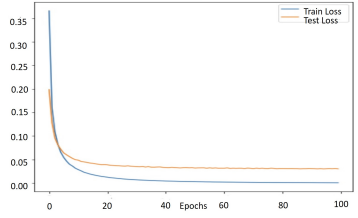
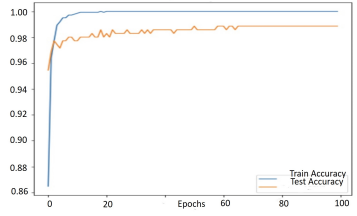
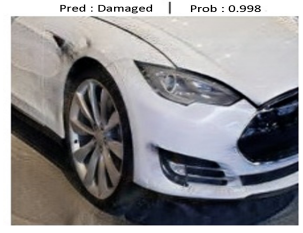
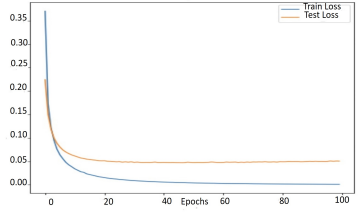
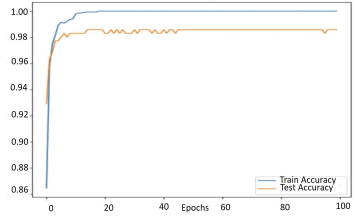
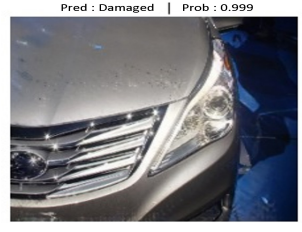
Table 11 A CNNs Classifier Predication for Real Images and Synthesized Images of both CycleGAN and AttentionGAN

Real Image	CycleGAN	AttentionGAN
		

5.4 Experimental Results of the ViT Classifier

When we built and trained a scratch ViT model, we faced the severly underfitting problem and its train and test accuracy were 49.7% and 50%. Nonetheless,

Table 12 A ViT Classifier Prediction for Real Dataset and Fake Datasets of both CycleGAN and AttentionGAN

Dataset	Train-Test Loss Plot	Train-Test Accuracy Plot	Prediction
Real			
CycleGAN			
AttentionGAN			

when we built and trained a pre-trained ViT model with transfer learning and fine-tuning, we achieved a training accuracy of 100% and a testing accuracy of 97.4%. Therefore, we utilized a pre-trained ViT as a binary classifier in our final task. Table 12 describes the ViT classifier of the loss-accuracy plots of both train and test with the prediction of three datasets with two classes such as damaged and undamaged as our final experimental task to assess the quality of generated images of CycleGAN and AttentionGAN comparing with original real images. Similar to the performance of the CNNs classifier, all predictions of the accuracy are not quite different. To evaluate between AttentionGAN and CycleGAN, the accuracy of AttentionGAN is slightly larger than the CycleGAN as a good performance.

6 Conclusion

In this work, we studied mainly focusing on how to defeat the problems of data scarcity, insufficient data diversity, and producing limited plausible alternative data by using the standard data augmentation techniques for low data-driven training of DL models. To overcome these problems and also the lack of a publicly available car damage dataset in car damage detection and classification tasks, we took the advantage of unpaired image-to-image translation GANs of CycleGAN and AttentionGAN to generate synthetic car damage images and the superfluity of different evaluation assessments: GAN metrics; a standard DL classification model of CNNs; and a fewer computational resources than the CNNs of ViT model to evaluate the translated unnatural realist car damaged images by approaching both quantitative, and qualitative assessments in our experiments.

First and foremost, we created a car unpaired dataset based on 3386 car damaged-undamaged images randomly splitting them for the training set (80%) and the test set (20%) for unpaired image-to-image translation GANs and started to present a useful framework for the comparative study of CycleGAN and AttentionGAN to generated car damaged images and consider both their quantitative and qualitative results with three different evaluation assessments in our experiments. After analyzing them with our framework, they can generate superior quality of synthesized car-damaged images including their features of damage type, location, and damage level via translating from original undamaged to damaged car images by CycleGAN and AttentionGAN. Moreover, they can also be produced the features of scratch, car paint, dent, and fire-burning damaged samples with the severity of minor and moderate by high-quality synthetic car images, but generating damaged samples with a severe condition does not seem to be realistic and looks blurry. In addition to the generation of the damaged samples, they can pretty much effort to generate scratch and car paint-damaged samples into three levels of severities, although they are difficult to generate dent samples of car images in the conditions of both damaged level of moderate and severe. However, we can effectively apply their synthesized car damaged data with our original real data in our evaluation assessments.

In our three quantitative assessments of GANs metrics, AttentionGAN gives higher values of IS, lower values of FID and KID when compared with all results of CycleGAN. According to the results of GANs metrics, we observe that AttentionGAN is better-performance than CycleGAN. In our evaluation assessments of two classifiers: CNNs to predict real or fake; and ViT to analyze damaged or undamaged via the original car damaged-undamaged images, and their generated fake damaged datasets, both models can generate realistic and significant car images with different types of damaged samples, but also their predictions of accuracy are just slightly better than one another. Anyhow, we can make fool often to our CNNs classifier into a wrong prediction as a reality when we test with generated fake damaged samples of AttentionGAN than CycleGAN. Conforming with overall experimental results, and performances, we can prove that AttentionGAN is better than CycleGAN. Finally we can confirm their high-quality synthesized car-damaged images as a reliable real-world car-damaged dataset in our experiments by overcoming our three facing challengens with four experimental tasks.

Therefore, this work will motivate other computer vision and ML researchers and can be further upgraded as a continuous research process to leverage the potential of CycleGAN, AttentionGAN and other deep generative models with different datasets and another evaluation methods or approaches for various applications of many applied domains of image classification by creating new synthetic data of any image generation to defeat the publicly unavailable high-quality data for data-driven DL models.

References

[1] Schmarje, L., Santarossa, M., Schrder, S.M., Koch, R.: A survey on semi-, self-and unsupervised learning for image classification. IEEE Access **9**, 82146–82168 (2021)

[2] Kyu, P.M., Woraratpanya, K.: Car damage detection and classification. Paper presented at the 11th international conference on advances in information technology, 1–6 July (2020)

[3] Kyu, P.M., Woraratpanya, K.: Car Damage Assessment Based on VGG Models. Paper presented at the 8th Joint Symposium on Computational Intelligence (JSCI8), 1–4 May (2020)

[4] Dwivedi, M., Malik, H.S., Omkar, S.N., Monis, E.B., Khanna, B., Samal, S.R., Tiwari, A., Rathi, A.: Deep learning-based car damage classification and detection. Paper presented at Advances in Artificial Intelligence and Data Engineering, Springer, 207–221 (2020)

[5] Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. Paper presented at the AAAI conference on artificial intelligence, 13001-13008, April (2020)

[6] Botteghi, N., Poel, M., Brune, C.: Unsupervised representation learning in deep reinforcement learning: A review. Preprint at https://arxiv.org/abs/2208.14226 (2022)

[7] Ericsson, L., Gouk, H., Loy, C.C., Hospedales, T.M.: Self-supervised representation learning: Introduction, advances, and challenges. IEEE Signal Processing Magazine **39(3)**, 42–62 (2022)

[8] Tan, Y.X., Lee, C.P., Neo, M., Lim, K.M.: Text-to-image synthesis with self-supervised learning. Pattern Recognition Letters **157**, 119–126 (2022)

[9] Pang, J. Y.and Lin, Qin, T., Chen, Z.: Image-to-image translation: Methods and applications. IEEE Transactions on Multimedia **24** (2021)

[10] Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(7) (2021)

[11] Mo, Y., Wu, Y., Yang, X., Liu, F., Liao, Y.: Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing **493**, 626–646 (2022)

[12] Singla, K., Pandey, R., Ghanekar, U.: A review on single image super resolution techniques using generative adversarial network. Optik **266** (2022)

[13] Zeng, Y., Fu, J., Chao, H., Guo, B.: Aggregated contextual transformations for high-resolution image inpainting. IEEE Transactions on Visualization and Computer Graphics (2022)

[14] Umer, A., Termritthikun, T. C.and Qiu, Leong, P.H., Lee, I.: On-device saliency prediction based on pseudoknowledge distillation. IEEE Transactions on Industrial Informatics **18**(9), 6317–6325 (2022)

[15] Liu, X., Yoo, C., Xing, F., Oh, H., El Fakhri, G., Kang, J.W., Woo, J.: Deep unsupervised domain adaptation: a review of recent advances and perspectives. APSIPA Transactions on Signal and Information Processing **11**(1) (2022)

[16] Qi, Y., Yang, Z., Sun, W., Lou, M., Lian, J., Zhao, W., Deng, X., Ma, Y.: A comprehensive overview of image enhancement techniques. Archives of Computational Methods in Engineering **29**(1), 583–607 (2022)

[17] Jin, D., Jin, Z., Hu, Z., Vechtomova, O., Mihalcea, R.: Deep learning for text style transfer: A survey. Computational Linguistics **48**(1), 155–205 (2022)

[18] Farahanipad, F., Rezaei, M., Nasr, M.S., Kamangar, F., Athitsos, V.: A survey on gan-based data augmentation for hand pose estimation problem. Technologies **10**(2) (2022)

[19] Treneska, S., Zdravevski, E., Pires, I.M., Lameski, P., Gievska, S.: Gan-based image colorization for self-supervised visual feature learning. Sensors **22**(4) (2022)

[20] Jampour, M., Zare, M., Javidi, M.: Advanced multi-gans towards near to real image and video colorization. Journal of Ambient Intelligence and Humanized Computing, 1–18 (2022)

[21] Ma, F., Li, Y., Ni, S., Huang, S.L., Zhang, L.: Data augmentation for audio-visual emotion recognition with an efficient multimodal conditional gan. Applied Sciences **12**(1) (2022)

[22] Shahriar, S.: Gan computers generate arts? a survey on visual arts, music, and literary text generation using generative adversarial network. Displays (2022)

[23] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. Advances in Neural Information Processing Systems. Paper presented at Advances in Neural Information Processing Systems, 12104-12114 (2020)

[24] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. Paper presented at the IEEE conference on computer vision and pattern recognition, 1125-1134 (2017)

[25] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. Paper presented at the IEEE international conference on computer vision, 2223-2232 (2017)

[26] Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. Paper presented in European conference on computer vision, 702-716 (2016)

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, ., Polosukhin, I.: Attention is all you need. Paper presented at Advances in neural information processing systems (2017)

[28] Ma, S., Fu, J., Chen, C.W., Mei, T.: Da-gan: Instance-level image translation by deep attention generative adversarial networks. Paper presented in the IEEE conference on computer vision and pattern recognition, 5657-5666 (2018)

[29] Cai, L., Zhu, L., Zhang, H., Zhu, X.: Da-gan: Dual attention generative adversarial network for cross-modal retrieval. Future Internet **14**(2) (2022)

[30] Alami Mejjati, Y., Richardt, C., Tompkin, J., Cosker, D., Kim, K.I.: Unsupervised attention-guided image-to-image translation. Paper presented at Advances in neural information processing systems (2018)

[31] Mo, S., Cho, M., Shin, J.: Instagan: Instance-aware image-to-image translation. Preprint at https://arxiv.org/abs/1812.10889 (2018)

[32] Emami, H., Aliabadi, M.M., Dong, M., Chinnam, R.B.: Spa-gan: Spatial attention gan for image-to-image translation. IEEE Transactions on Multimedia **23**, 391–401 (2020)

[33] Liang, X., Zhang, H., Lin, L., Xing, E.: Generative semantic manipulation with mask-contrasting gan. Paper presented at the European Conference on Computer Vision (ECCV), 558-573 (2018)

[34] Chen, X., Xu, C., Yang, X., Tao, D.: Attention-gan for object transfiguration in wild images. Paper presented at the European Conference on Computer Vision (ECCV), 164-180 (2018)

[35] Kastaniotis, D., Ntinou, I., Tsourounis, D., Economou, G., Fotopoulos, S.: Attention-aware generative adversarial networks (ATA-GANs). Paper presented In 2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 1-5 (2018)

[36] Tang, H., Liu, H., Xu, D., Torr, P.H., Sebe, N.: Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. IEEE Transactions on Neural Networks and Learning Systems (2021)

[37] Gokaslan, A., Ramanujan, V., Ritchie, D., Kim, K.I., Tompkin, J.: Attention-aware generative adversarial networks (ATA-GANs). Paper presented in the European Conference on Computer Vision (ECCV), 649-665 (2018)

[38] Arruda, V.F., Paixo, T.M., Berriel, R.F., De Souza, A.F., Badue, C., Sebe, N., Oliveira-Santos, T.: Cross-domain car detection using unsupervised image-to-image translation: From day to night. Paper presented in 2019 International Joint Conference on Neural Networks (IJCNN), 1-8 (2019)

[39] Sarv Ahrabi, S., Momenzadeh, A., Baccarelli, E., Scarpiniti, M., Piazzo, L.: How much bigan and cyclegan-learned hidden features are effective for covid-19 detection from ct images? a comparative study. The Journal of Supercomputing, 1–32 (2022)

[40] Lee, K.W., Chin, R.K.Y.: A Comparative Study of COVID-19 CT Image Synthesis using GAN and CycleGAN. Paper presented in 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 1-6 (2022)

[41] Ubale Kiru, M., Belaton, B., Chew, X., Almotairi, K.H., Hussein, A.M., Aminu, M.: Comparative analysis of some selected generative adversarial network models for image augmentation: a case study of covid-19 x-ray and ct images. Journal of Intelligent & Fuzzy Systems, 1–20 (2022)

[42] Sharma, N., Sharma, R., Jindal, N.: Comparative analysis of cyclegan and attentiongan on face aging application. aSdhan **47**(1), 1–20 (2022)

[43] Haar, L.V., Elvira, T., Ochoa, O.: An analysis of explainability methods for convolutional neural networks. Engineering Applications of Artificial Intelligence **117** (2023)

[44] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., Glocker, B.: Attention u-net: Learning where to look for the pancreas. Preprint at https://arxiv.org/abs/1804.03999 (2018)

[45] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. Paper presented in International conference on machine learning, 1691-1703 (2020)

[46] Rosso, M.M., Marasco, G., Aiello, S., Aloisio, A., Chiaia, B., Marano, G.C.: Convolutional networks and transformers for intelligent road tunnel investigations. Computers & Structures **275** (2023)

[47] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J.: An image is worth 16x16 words: Transformers for image recognition at scale. Preprint at https://arxiv.org/abs/2010.11929 (2020)

[48] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. Paper presented in European conference on computer vision, 213-229 (2020)

[49] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. Paper presented in the IEEE/CVF conference on computer vision and pattern recognition, 6881-6890 (2021)

[50] hen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C.,

Xu, C., Gao, W.: Pre-trained image processing transformer. Paper presented in the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12299-12310 (2021)

[51] Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. Paper presented in the the IEEE conference on computer vision and pattern recognition, 8739-8748 (2018)

[52] Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM computing surveys (CSUR) **54**(10s), 1–41 (2022)

[53] Borji, A.: Pros and cons of gan evaluation measures. Computer Vision and Image Understanding **179**, 41–65 (2019)

[54] Borji, A.: Pros and cons of gan evaluation measures: New developments. Computer Vision and Image Understanding **215** (2022)

[55] Betzalel, E., Penso, C., Navon, A., Fetaya, E.: A Study on the Evaluation of Generative Models. Preprint at https://arxiv.org/abs/2206.10935 (2022)

[56] Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review **53**(8), 5455–5516 (2020)

[57] Lin, Y., Wang, Y., Li, Y., Gao, Y., Wang, Z., Khan, L.: Attention-based spatial guidance for image-to-image translation. Paper presented in the IEEE/CVF Winter Conference on Applications of Computer Vision , 816-825 (2021)

[58] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Paper presented in the Advances in neural information processing systems (2015)

[59] Donahue, J., Krhenbhl, P., Darrell, T.: Adversarial feature learning. Preprint at https://arxiv.org/abs/1605.09782 (2016)

[60] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2020). https://doi.org/10.1145/3422622

[61] Mirza, M., Osindero, S.: Conditional generative adversarial nets. Preprint at https://arxiv.org/abs/1411.1784 (2014)

[62] Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint at https://arxiv.org/abs/1511.06434 (2015)

[63] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Paper presented in the Advances in neural information processing systems (2016)

[64] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. Paper presented in International conference on machine learning, 214-223 (2017)

[65] Krause, M. J.and Stark, Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. Paper presented in athe IEEE international conference on computer vision workshops, 554-561 (2013)

[66] Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Paper presented in the IEEE international conference on computer vision, 2650-2658 (2015)

[67] Isola, P., Zhu, J.Y., T., Z., Efros, A.A.: Image-to-image translation with conditional adversarial networks. Paper presented in the IEEE conference on computer vision and pattern recognition, 1125-1134 (2017)

[68] Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. Paper presented in the European conference on computer vision, 694-711 (2016)

[69] Laffont, P.Y., Ren, Z., Tao, X., Qian, C., Hays, J.: Transient attributes for high-level understanding and editing of outdoor scenes. ACM Transactions on graphics (TOG) **33**(4), 1–11 (2014)

[70] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. Paper presented in the IEEE conference on computer vision and pattern recognition, 3431-3440 (2015)

[71] Shih, Y., Paris, S., Durand, F., Freeman, W.T.: Data-driven hallucination of different times of day from a single outdoor photo. ACM Transactions on Graphics (TOG) **32**(6), 1–11 (2013)

[72] Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. Paper presented in the European conference on computer vision, 318-335 (2016)

[73] Xie, S., Tu, Z.: Holistically-nested edge detection. Paper presented in the

IEEE international conference on computer vision, 1395-1403 (2015)

[74] Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. Paper presented in the seventh IEEE international conference on computer vision, 1033-1038 (1999)

[75] Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. Paper presented in the IEEE conference on computer vision and pattern recognition, 8798-8807 (2018)

[76] Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. Paper presented in Advances in neural information processing systems (2016)

[77] Aytar, Y., Castrejon, L., Vondrick, C., Pirsiavash, H., Torralba, A.: Cross-modal scene networks. IEEE transactions on pattern analysis and machine intelligence **40**(10), 2303–2314 (2017)

[78] Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. Paper presented in Advances in neural information processing systems, (2017)

[79] Eikema, B., Aziz, W.: Auto-encoding variational neural machine translation. Preprint at https://arxiv.org/abs/1807.10564 (2018)

[80] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)

[81] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. Paper presented in the IEEE conference on computer vision and pattern recognition, 2107-2116 (2017)

[82] Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. Preprint at https://arxiv.org/abs/1611.02200 (2016)

[83] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. Paper presented in the IEEE conference on computer vision and pattern recognition, 3722-3731 (2017)

[84] Yang, C., Kim, T., Wang, R., Peng, H., Kuo, C.-C.J.: Show, Attend and Translate: Unsupervised Image Translation with Self-Regularization and Attention (2019)