A survey of synthetic data augmentation methods in computer vision

Alhassan Mumuni[1]*, Fuseini Mumuni[2] and Nana Kobina Gerrar[1]

Abstract—The standard approach to tackling computer vision problems is to train deep convolutional neural network (CNN) models using large-scale image datasets which are representative of the target task. However, in many scenarios, it is often challenging to obtain sufficient image data for the target task. Data augmentation is a way to mitigate this challenge. A common practice is to explicitly transform existing images in desired ways so as to create the required volume and variability of training data necessary to achieve good generalization performance. In situations where data for the target domain is not accessible, a viable workaround is to synthesize training data from scratch—i.e., synthetic data augmentation. This paper presents an extensive review of synthetic data augmentation techniques. It covers data synthesis approaches based on realistic 3D graphics modeling, neural style transfer (NST), differential neural rendering, and generative artificial intelligence (AI) techniques such as generative adversarial networks (GANs) and variational autoencoders (VAEs). For each of these classes of methods, we focus on the important data generation and augmentation techniques, general scope of application and specific use-cases, as well as existing limitations and possible workarounds. Additionally, we provide a summary of common synthetic datasets for training computer vision models, highlighting the main features, application domains and supported tasks. Finally, we discuss the effectiveness of synthetic data augmentation methods. Since this is the first paper to explore synthetic data augmentation methods in great detail, we are hoping to equip readers with the necessary background information and in-depth knowledge of existing methods and their attendant issues.

Index Terms—Data augmentation, generative AI, neural rendering, data synthesis, synthetic data, neural style transfer.

1 Introduction

1.1 Background

Currently, deep learning is the most important technique for solving many complex machine vision problems. State-of-the-art deep learning models typically contain very large number of parameters that need to be learned in order to characterize a wide range of visual phenomena. Moreover, as a result of the enormous appearance variations of real-world objects and scenes, there is often the need to introduce various variations of the available data during training. Consequently, training deep learning models require very huge amount of annotated data to guarantee good generalization performance and avoid overfitting. However, data collection and annotation are often time-consuming and costly exercises. Instead of attempting to collect very large quantities of annotated data, it is often more practical to create new samples artificially. Data augmentation is the process of creating new data to artificially extend training set. Typically, the process of augmentation involves performing transformations on the original data so as to alter them in a particular way. The transformation operations usually change the visual characteristics of the data but preserve their labels. Data augmentation (DA) can, thus, be seen as a means for simulating real-world behavior such as the visual appearance of objects and scenes under different view angles, pose variations, object deformations, lens distortions and other camera artifacts. In practice, there are several situations in which the training of machine learning (ML) models may require data augmentation. The commonest scenarios include the following:

- Training a deep learning models but the quantity of training data is small
- Adequate training data exists but is of perceptually poor quality (e.g., low resolution, hazy or blurry)
- Available training data is not representative of the target data (e.g., does not have adequate appearance variations)
- The proportion of various classes is skewed (imbalanced data)
- Data is only available for one condition (e.g., bright day) but there is the need to train models to perform inference under different set of conditions (e.g., night, rainy or foggy weather)
- There is no practical way to access data for training (e.g., excessive cost or restriction)

The first four problems can be adequately solved by manipulating the existing data to produce additional data that enhances the overall performance of the trained model. In the case of the last two problems, however, the only viable solution is to create new training data.

1.2 Significance of synthetic data augmentationn

As discussed earlier, the commonest approach to data augmentation is to transform training data in various ways. However, in application scenarios where no training data exists naturally, or where its collection is too costly, it often becomes impractical to create additional training data using

[1]*Cape Coast Technical University, Cape Coast, Ghana.*
**Corresponding author, E-mail: alhassan.mumuni@cctu.edu.gh*
[2]*University of Mines and Technology, UMaT, Tarkwa, Ghana*

the aforementioned methods. Moreover, many computer vision tasks are often use-case sensitive, requiring task-specific data formats and annotation schemes. This makes it difficult for broadly-annotated, publicly-available large-scale datasets to meet the specific requirements of these tasks. In these cases, the only viable approach is to generate training data from scratch. Modern image synthesis methods can simulate different kinds of task-specific, real-world variability in the synthesized data. They are particularly useful in applications such as autonomous driving and navigation [1], [2], pose estimation [3], [4], affordance learning [5], [6], object grasping [7], [8] and manipulation [9], [10], where obtaining camera-based images is time-consuming and expensive. Moreover, in some applications,bitmap pixel images may simply be unsuitable. Data synthesis methods can readily support non-standard image modalities such as point clouds and voxels. Approaches based on 3D modeling also provide more scalable resolutions as well as flexible content and labeling schemes that is adapted for the specific use-case.

1.3 Motivation for this survey

Data augmentation approaches based on data synthesis are becoming increasingly important in the wake of severe data scarcity in many machine learning domains. In addition, the requirements of emerging machine vision applications such as autonomous driving, robotics and virtual reality are increasingly becoming difficult to be met using traditional data transformation-based augmentation. For this reason, data synthesis has become an important means to provide quality training data for machine learning applications. Unfortunately, however, while many surveys on data augmentation approaches exist, very few works deal with synthetic data augmentation methods. This work is motivated by the lack of adequate discussion on this important class of techniques in the scientific literature. Consequently, we aim to provide an in-depth treatment of synthetic data augmentation methods to enriched the current literature on data augmentation. We discuss the various issues on data synthesis in detail, including concise information about the main principles, use-cases and limitations of the various approaches.

1.4 Outline of work

In this work, we first provide a broad overview of data augmentation in Section 2 and provide a concise taxonomy of synthetic data augmentation approaches in Section 3. Further, in Section 4 through 7, we explore in detail the various techniques for synthesizing data for machine vision tasks. Here we discuss the important principles, approaches, use-cases and limitations of each of the main classes of methods. The approaches surveyed in this work are generative modeling, computer graphics modeling, neural rendering, and neural style transfer (NST). We present a detailed discussion of each of these approaches in the following sections. We also compare the advantages and disadvantages of these classes of data synthesis methods. We summarize the main features of common synthetic datasets in Section 8. In Section 9 we discuss the effectiveness of synthetic data augmentation in machine vision domains. We present a summary of the main issues in Section 10 and outline promising directions for future research in Section 11. Finally, conclude in Section 12. A detailed outline of this survey is presented in Figure 1.

2 Overview of data augmentation methods

Geometric data augmentation methods such as affine transformations [11], projective transformation [12] and nonlinear deformation [13] are aimed at creating various transformations of the original images to encode invariance to spatial variations resulting from, for example, changes in object size, orientation or view angles. Common geometric transformations include rotation, shearing, scaling or resizing, nonlinear deformation, cropping and flipping. On the other hand, photometric techniques – for example, color jittering [14], lighting perturbation [15], [16] and image denoising [17] – manipulate the qualitative properties – for example, image contrast, brightness, color, hue, saturation and noise levels – and thereby render the resulting deep learning models invariant to changes in these properties. In general, to ensure good generalization performance in different scenarios, it is often necessary to apply many of these procedures simultaneously.

Recently, more advanced data augmentation methods have become common. One of the most important class of techniques [18], [19], [20], [21] is based on transforming different image regions discretely instead of uniformly manipulating the entire input space. This type of augmentation methods have been shown to be effective in simulating complex visual effects such as non-uniform noise, non-uniform illumination, partial occlusion and out-of-plane rotations.

The second main direction of data augmentation exploits feature space transformation as a means of introducing variability of training data. These regularization approaches manipulate learned feature representations within deep CNN layers to transform the visual appearance of the underlying images. Examples of feature-level transformation approaches include feature mixing [18], [22] , feature interpolation [23], feature dropping [24] and selective augmentation of useful features [25]. These methods do not often lead to semantically meaningful alterations. Nonetheless, they have proven very useful in enhancing the performance of deep learning models. The third direction is associated with the automation of the augmentation process. To achieve this, typically, different transformation operations based on traditional image processing techniques are applied to manually generate various primitive augmentations. Optimization algorithms are then used to automatically find the best model hyperparameters, as well as the augmentation types and their magnitudes for the given task.

The approaches described above are realizable only when training data exists, and the goal of augmentation is to transform the available data to obtain desirable features. This work focuses on approaches that seek to generate novel training data even in cases where data for the target task is inaccessible.

Several survey works (e.g., [26], [27], [28], [28], [29], [30]) have explored data augmentation in great detail. Shorten et al. [27], in particular, present a broad discussion of important data augmentation methods. However, like most

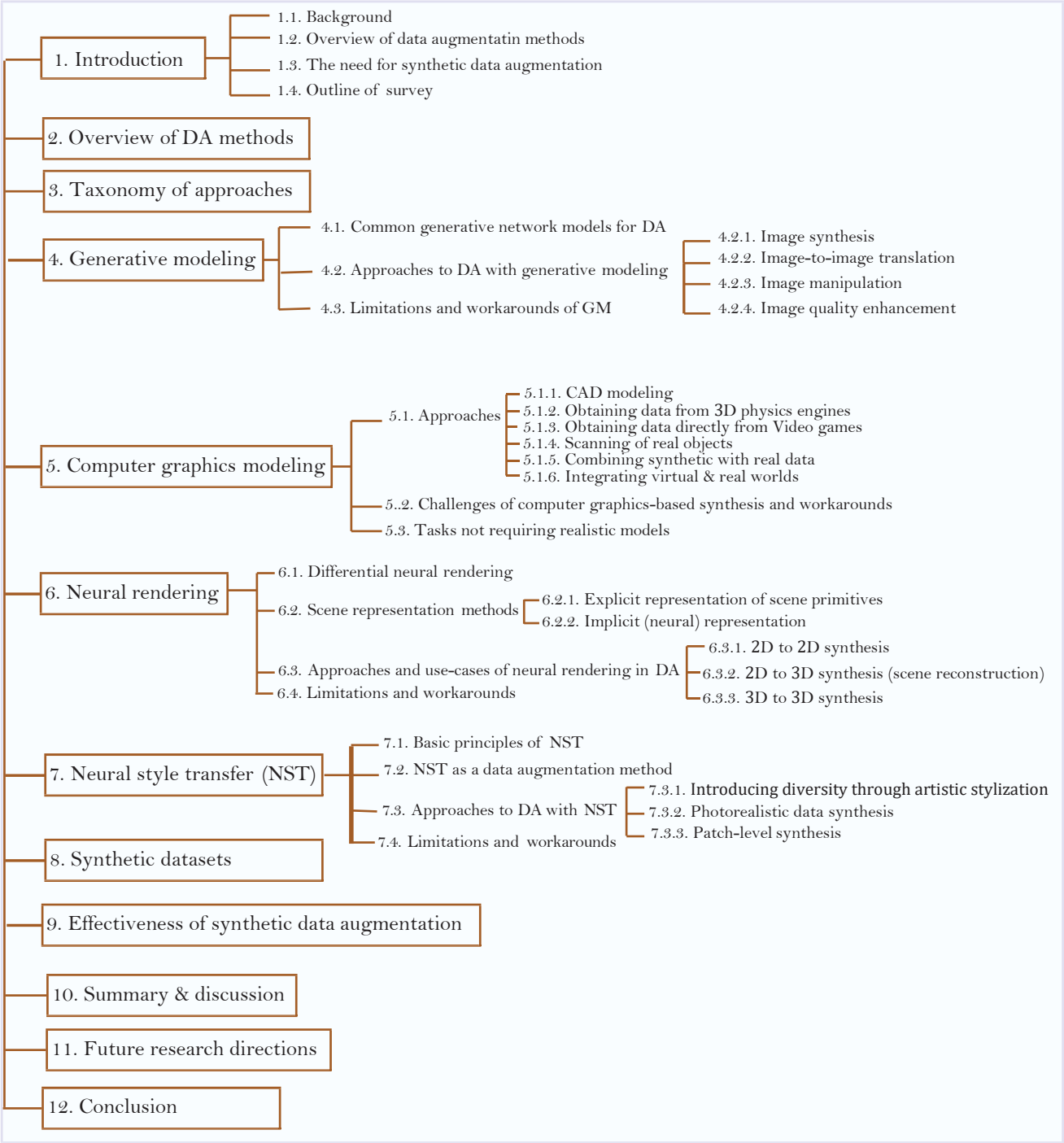

Figure 1. Detailed outline of work

previous surveys, their coverage of data synthesis methods is rather limited.

To address this gap, in this surveys, we focus mainly on data augmentation techniques that generate synthetic data for training machine learning models in computer vision domains. data augmentation methods. The main approaches covered here are methods based on generative AI, procedural data generation using 3D CAD tools and game engines, differential neural rendering and neural style transfer. We consider that such a narrow scope will enable us to provide a much detailed treatment of topic and its important issues while at the same time maintain a relatively concise volume.

3 Taxonomy of synthetic data augmentation methods

In practice, four main classes of synthetic data generation techniques are commonly used:

- generative modeling
- computer graphics modeling
- neural rendering
- neural style transfer

Generative modeling methods rely on learning the inherent statistical distribution of input data in order to (automatically) generate new data. The second class of

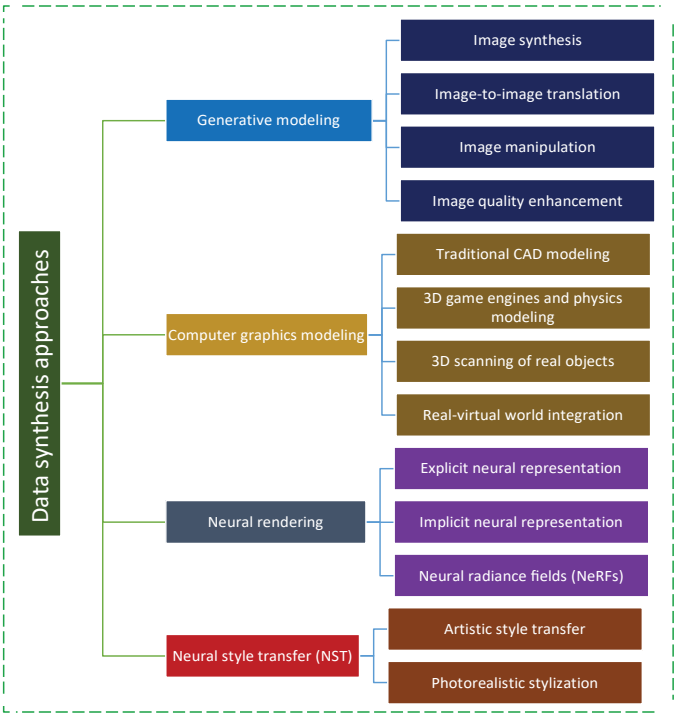

Figure 2. Taxonomy of synthetic data augmentation approaches

approaches, computer graphics modeling, are based on the elaborate process of manually constructing 3D models of objects and scenes with the aid of computer graphics tools. Neural rendering approaches are based on special methods for constructing novel data using conventional feedforward neural network architectures are described. They allow deep neural networks to generate completely new images by learning intermediate 3D representations. Where needed, the generated 3D data can be used for training. The fourth class of methods are known as neural style transfer. These approaches combine features of different semantic levels extracted from different images to create a new set of images. A general classification of synthetic data augmentation approaches is depicted in Figure 2.

4 GENERATIVE MODELING

Generative AI techniques present the most promising prospect for generating synthetic datasets for complex computer vision tasks. Generative modeling methods are a class of deep learning techniques that utilize special deep neural network architectures to learn holistic representation of the underlying categories in order to generate useful synthetic data for training deep learning models. Generally, they work by learning possible statistical distributions of the target data using noise or examples of target data as input. This knowledge about the distribution of training data, thus, can enable them to generate complex representations. Examples of generative models include Boltzmann machines (BMs) [31] and restricted Boltzmann machines (RBMs) [32], generative adversarial networks (GANs) [33], variational autoencoders (VAEs) [34], autoregressive models [35] and deep belief networks (DBNs) [36]. Currently, GANs and VAEs and their various variants such as [37], [38], [39] are the most popular neural network architectures for generative modeling.

4.1 Common generative AI models for data generation

a. Generative adversarial network (GAN)

The general structure of the GAN is shown in Figure 3a. For image generation, the basic working principle of the GAN is as follows. A generator samples multi-dimensional noise from a random distribution and converts this noise into a representation similar to real images. A discriminator then tries to distinguish between real images and the artificially generated samples and provides feedback about its predictions to the generator. The generator, aiming to produce samples that are indistinguishable from real ones, iteratively refines its prediction based on the feedback error computed by the loss function. In a similar way, the discriminator uses its own loss to improve subsequent predictions. This process eventually leads to the model generating high-quality images. The vanilla GAN uses fully connected multi-layer deep neural network architecture for the implementation of both Generator and Discriminator sub-models.

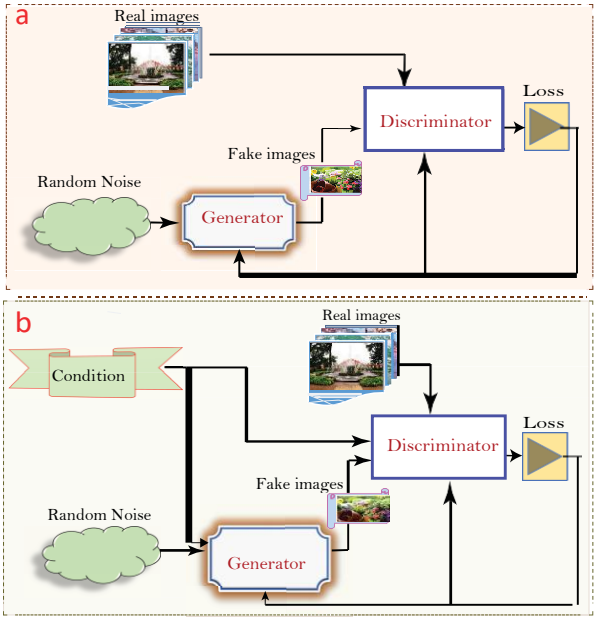

Figure 3. Functional block diagram of a basic GAN (a) and Conditional GAN (b) models.

In the Conditional GAN (cGAN) [40], the generation process is conditioned on control input to the generator and discriminator (Figure 3 (b)). This provides additional information that helps the network to reproduce desired characteristics in the target class. From a simple, fully connected architecture in [33], [40], many new architectural innovations have been introduced to improve the GAN's ability to model data in image domains. Notable among these include the Deep Convolutional GAN (DCGAN), which employs convolutional layers for the generator and transpose-convolutional layers for the discriminator instead of fully connected layers throughout; the Laplace Pyramid GAN (LAPGAN) [41], which uses multiple generator-discriminator pairs in a multi-scale pyramidal structure; Information Maximizing Generative Adversarial Network

(InfoGAN) [42], which proposes a non-conventional representation of the multi-dimensional noise, dividing it into an input noise vector and a latent variable, in order to help learn complex factors that control image appearance; Super Resolution GAN (SRGAN) [43], utilizes deep CNN together with adversarial networks to significantly increase the resolution training images; Pix2pix [40] propose paired image-based image-to-image translation approach using a cGAN model, DiscoGAN [44] and CycleGAN [45] both utilize a pair of generators and discriminators to perform unsupervised image-to-image translation (i.e., they convert images from one style to the another without using paired images); and Self-Attention GAN (SAGAN) [46], which incorporates attention mechanism into the GAN structure to model more global visual features and long-term dependencies. These and many other improvements have enabled GANs to synthesize high-detailed, photorealistic images with natural texture and lighting for training deep models. Figure 5 provides a visual illustration of how GANs have advanced over the years. Figure 6 shows how photorealistic images can be generated by modern generative adversarial networks (in this case BigGAN)can generate. In Figure 8, we show the multi-view 3D image synthesis capabilities of several state-of-the-art GAN models: GRAF [47], GIRAFFE [48], pi-GAN [49] and MVCGAN [50].

Some GAN models [51], [52], [53], [54] incorporate autoencoders into their structures to learn a latent space in which different images attributes (e.g., object texture, pose, or facial expression) can be easily manipulated. An autoencoder is basically made up of an encoder and decoder sub-models. The encoder transforms input data into a random, lower-dimensional representation in latent space. The decoder reconstructs the original data by mapping the low-dimensional data in the high-dimensional output space. The overall goal is to guarantee that the reconstructed data is as similar as possible to the original data. Models that employ autoencoders in their structure are able to learn a joint distribution of data and latent variables. Learning this joint space simplifies the process of creating desirable visual characteristics in output space as this can be accomplished indirectly by manipulating latent variables.

b. Variational autoencoders

Another class of generative AI techniques that has shown enormous promise is the variational autoencoder (VAE). Like the generative adversarial network, the variational autoencoder can be trained to generate novel data from a given domain. In addition to data generation, VAEs, like GANs, can perform tasks such as anomaly detection and correction, noise elimination, and data refinement. As we will see in Subsection 4.2, all these use-cases are important data augmentation functions in computer vision. Variational autoencoders are commonly used jointly with GANs and other generative models to achieve better results. They have been used to synthesize realistic images, videos as well as text for computer vision tasks.

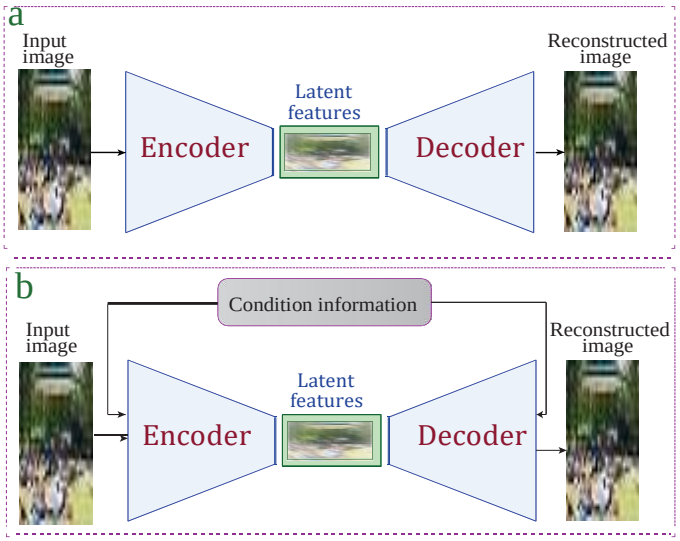

Figure 4. Functional block diagram of a basic VAE (a) and Conditional VAE (b) models.

Figure 5. The rapidly improving performance of generative adversarial networks. Shown here is the quality of generated images by state-of-the-art models through the period 2014 to 2017. Images courtesy Brundage et al. [55].

Figure 6. High quality photorealistic images generated by BigGAN [56].

The VAE (Figure 4a) was first proposed by Kingma and Welling in [34]. Since then, VAEs have been widely used to generate data for training deep learning models. Similar to GANs, VAEs learn the probability distribution of input data to help generate new data with similar characteristics. VAE first maps each input point to a normal distribution in a latent space using the encoder sub-model. The decoder then samples from this normal distribution to generate new samples, ensuring that the distribution of the real and generated data is as close as possible. The conditional variational autoencoder (cVAE), like the conditional GAN, uses additional metadata to generate outputs based on desirable characteristics defined by conditional input (Figure 4b). Since by their nature VAEs are not susceptible to the mode collapse problem that GANs suffer from, and GANs generate much perceptually-better images as compared to VAEs, many recent works (e.g., [37], [38], [39], [57] have

proposed more complex generative modeling frameworks that leverage the advantages of both types of models.

4.2 Approaches to image data augmentation with generative AI models

In the context of data augmentation, generative models can be applied in several different ways. These include to generate new images, to transfer specific image characteristics from source to target images, and to enhance the perceptual quality or diversity of training data. data augmentation approaches based on these different principles have been used in diverse computer vision tasks, including medical image classification [58], [59], object detection [60], pose estimation [61] and visual tracking [62]. Common approaches to solving data augmentation problems based on generative modeling techniques are presented in subsections 4.2.1 to 4.2.4. The key aspects and application scenarios of these methods are summarized in Table 1.

4.2.1 Image synthesis

In application settings where it is difficult or impossible to obtain sufficient labelled data, the main goal of generative modeling is to generate synthetic data [63], [64], [65], [66], [67] to be used in place of, or in combination with real data. Models used in this situation are aimed at synthesizing specific categories of image data to aid training. The primary objective of the generative modeling, then, is to generate samples that cover the distribution of the underlying categories. This type of data can be achieved with conventional CNN-based GAN architectures without utilizing conditional information as in the case of conditional GANs or conditional VAEs. For example, Kaplan et al. [68] demonstrated the ability of GAN and VAE to generate photorealistic retinal images without using conditional information. In [64], Bowles et al. employ a PGGAN model to synthesize images for Computed Tomography (CT) and Magnetic Resonance (MR) image segmentation tasks. The authors showed that using GANs to generate synthetic images improves segmentation performance on the two different tasks, irrespective of the original data size and the proportion of synthetic samples added. To improve the perceptual quality of generated images, several GANs may be used, each tuned for creating specific category (e.g., in [63]). In [66], Souly et al. proposed generative adversarial network to generate large amount of labeled images in a semi-supervised manner using unlabeled GAN-synthesized image data to help in semantic segmentation tasks.

4.2.2 Image-to-image translation

Image-to-image translation [40] is a technique used to transform an image by transferring its content into the visual style of another image. In its basic form, the approach involves learning a mapping from source to target domain.The approaches rely on principles of conditional generation using models such as cGANs and cVAEs. Generative modeling methods based on image-to-image translation can be used to convert images from one color space to another. In particular, approaches for converting among infrared, grayscale and RGB color images are common [73], [74]. The techniques can also enable different visual effects and specific features such as contrast, texture, illumination and other complex photometric transformations which would otherwise be challenging for traditional augmentation approaches. The approach, as a data augmentation method, has a wide scope of applications in computer vision. In medical imaging applications, for example, approaches based on image-to-image translation can be used to transfer images from one modality to another (e.g., from CT to MRI or X-ray image format). Figure 7 shows different translations using StyleGAN model [45].

Another important application of image-to-image translation is to synthesize view-consistent scenes, where 3D information and the overall spatial structure of the scene is preserved in the process of translation [49], [75]. This is extremely useful in semantic scene understanding tasks. Image-to-image translation approaches have also been widely used to generate images of novel views from particular views such as frontal facial views from angular views (e.g., [76], [77]) or to produce different human poses from a single pose (e.g., [78]). The effectiveness of image-to-image translation as a viable data augmentation strategy has also been demonstrated in challenging computer vision tasks such as visual tracking [79], [80], [81], person re-identification [82], [83], [84], object detection under severe occlusion [85] and strong lighting conditions [86]. Figure 8 depicts multi-view images generated by different GAN models.

While traditional image-to-image translation approaches [40] are generally based on cGANs and cVAEs architectures that utilize paired images, new techniques are based on the concept of cyclic consistency. For example, CycleGAN [45] DualGAN [87] and DiscoGAN [44] can translate images from one style to another without paired images. These methods learn a mapping function between source and target image domains by means of unsupervised learning—that is, images in the target domain do not have corresponding examples in the source. The approach is useful in many practical application since it is often challenging to obtain paired images in real-world scenarios for training machine learning models. For instance, for a specific environment, images for autonomous driving tasks may only be available for a limited set of weather conditions, but it may be required to improve robustness by training on a wide range of possible conditions. In such a situation, with unpaired image-to-image translation methods, the available content image can be transferred to all desired visual appearances without requiring additional content images.

The integration of large language models (LLMs) [88] and vision-language model (VLMs) [89] with GANs, VAEs and other generative models allows the process of image-to-image translation to be automated using textual prompts. LLMs and VLMs can also enable textual descriptions of visual scenes to be automatically generated as supplementary input for training computer vision models.

4.2.3 Image manipulation

Another common generative modeling approach to data augmentation is to qualitatively transform training data in desired ways by performing specific photometric (e.g., [91], [92]) and geometric ([71], [93], [94], [95]) image manipulations. Photometric image operations such as binarization

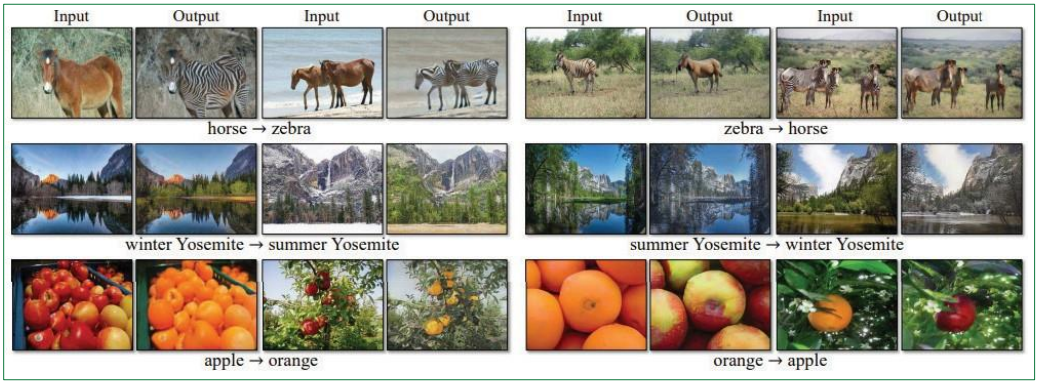

Figure 7. Examples of image-to-image translation by CycleGAN [45].

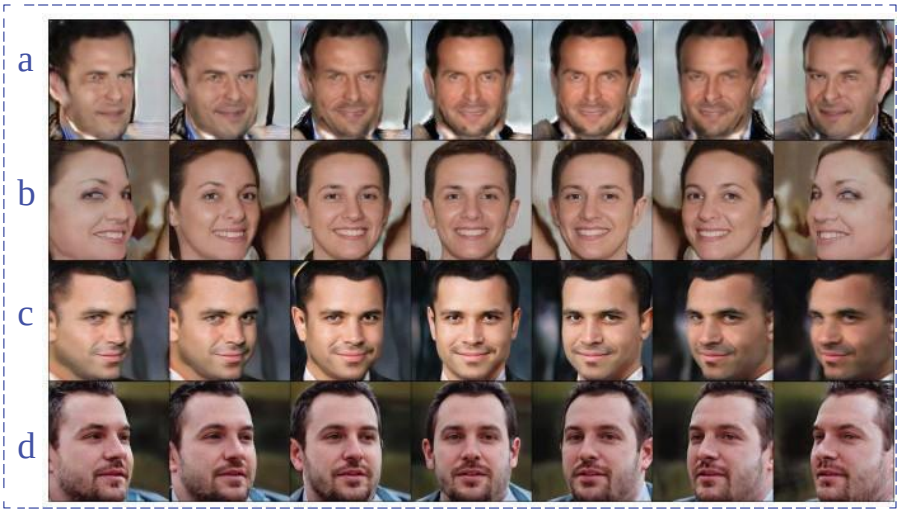

Figure 8. Photorealistic, multi-view images generated by 3D-aware image synthesis approaches: (a) GRAF [47], (b) GIRAFFE [48], (c) pi-GAN [49] and (d) MVCGAN [50].

Table 1
Common approaches to data augmentation by using generative modeling

Approach	Main function of generative model	Classic models	Application scenario
Image synthesis	Generate new samples per given categories	Deep Convolutional GAN (DCGAN) (e.g., in [69])	Synthesize new samples where no training data exist
Image-to-image translation	Reproduce specific visual characteristics	Conditional GAN (CGAN) and variants such as DiscoGAN [44] and CycleGAN [45]	Transfer desired visual appearance to training samples
Image manipulation	Perform specific transformations on input images	Spatial Transformer GAN (ST-GAN) [70]; MOST-GAN [71]	Produce diversity in training data to enhance generalization
Image Enhancement	Denoise images or improve quality or photorealism of training data	Denoising GAN (DN-GAN) [72]; Super Resolution GAN (SRGAN) [43]	Improve the perceptual quality of training samples

[91], colorization (i.e., conversion from grey-scale to color images) [96] and dehazing [92] are common tasks that can be accomplished with generative modeling.

4.2.4 *Image quality enhancement*

In some computer vision tasks, images available for training deep learning models are often of low quality. One way to improve performance is to enhance the quality of the training data. For example, generative modeling is commonly used to clean noisy images (e.g., in [72]). Also, low resolution images, can be qualitatively improved by using super-resolution GANs such as Pix2Pix [40], SRGAN [43], ESRGAN [97] or their derivatives. In a recent work [98], Wang et al. used a modified Pix2Pix model as a super-resolution GAN to increase the resolution of low-resolution, microscopic images for training deep neural networks. They first generated additional data using CycleGAN before employing the super-resolution GAN to improve the quality of the training dataset. In many studies, generative models have been used to generate large, clean images from noisy (e.g., [99], [100]) data, low resolution images (e.g., [101]), corrupted labels (e.g., [102]) or images taken in

adverse weather conditions such as rainy weather [90]. Figure 9 shows the effectiveness of image enhancement techniques such as de-raining in improving the perceptual input data. Generative modeling approaches that apply geometric transforms on training samples have also been reported in [103], [104], [105]. Some recent approaches are aimed at enhancing the perceptual quality of CAD-generated models. For example, RenderGAN [105] and DA-GAN [106] seek to improve performance by refining simple, synthetically generated 3D models so as to endow them with photorealistic appearance and desirable visual properties.

Figure 9. Examples of using generative modeling approaches to augment data by enhancing perceptual quality, in this case – by de-raining. The images by Zhang et al. [90] shows improvement in object detection performance after de-raining input samples.

4.3 Common limitations of generative modeling techniques and possible workarounds

One of the main problems with generative modeling methods is that they require very large training data for good performance [107]. GANs are also susceptible to overfitting – a situation where the discriminator memorizes all the training inputs and no longer offer useful feedback for the generator to improve performance. To address these problems of GAN performance, a number of works [108], [109], [110] have considered augmenting the data on which the generative model is trained. These approaches have been shown to be effective in alleviating small data and overfitting problems. However, employing augmentation strategies can lead to a situation where the generator reproduces samples from the distribution of the augmented data which may not be truly representative of the target task. Consistency regularization [111], [112] is a recent approach that has been proposed to prevent augmented data from being strictly reproduced by the generator. More advanced methods to improve GAN generalization include techniques based on perturbed convolutions [113] and Extreme Value Theory [114]. For instance, to enable GANS to handle rare samples, Bhatia et al. [115] proposed a probabilistic approach based on Extreme Value Theory [114] that allows to generate realistic as well as out-of-distribution or extreme training samples from a given distribution. In this context, extreme samples are training examples that deviate significantly from those present in the dataset. The approach also provides a way to set degree of deviation and the likelihood of the occurrence or proportion of these deviations in the generated data. Liu et al [116] demonstrated that GAN may in some situations fail to generate the task-required data (as a result of optimizing for a different task altogether) because GANs may be optimizing for a different objective. Specifically, in [116] a GAN designed for object detection tasks was shown to (have) rather optimize for realism of generated images.

Like with all data synthesis methods, it is currently not possible to directly compare different sets of synthetic data, or even to determine the suitability of synthesized data for a particular task without carrying out exhaustive tests. The fundamental problem is the general lack of quality metrics that can objectively evaluate the fitness of data for a given task. While techniques like log-likelihood provides a means to evaluate and assess the quality of VAEs, it is currently difficult to extend these to objectively compare the quality of GAN models. Some workarounds exist that allow to roughly estimate the quality of generated samples based on their similarity with the target data. This involves comparing the statistics of the generated data to that of the target data. The simplest metrics involve using more traditional similarity measures such as nearest neighbors, log-likelihoods [57], Minimum Mean Discrepancy (MMD) [117] and Multi-scale Structural Similarity Index Measure (MS-SSIM) [118]. Since these techniques merely estimate pixel distribution, high scores on the metrics do not strictly indicate high image quality. More advanced metrics allow to quantitatively estimate data diversity (i.e., the degree to which the synthetic data approximates the distribution of the target data), quality (i.e., overall photorealism), and other characteristics. The most important of these metrics include Inception Score (IS) [119] and Fréchet Inception Distance (FID) [120], as well as their new variants such as Spatial FID [121], Unbiased FID [122], Memorization-informed FID [123] and Class-aware FID [124]. These metrics allow to evaluate not only the general quality but also important aspects such as bias and fairness of generative models. Manual evaluation by visual inspection is another common way to determine the quality of data [125], [126] synthesized using generative modeling techniques. The approach relies on the developer's domain knowledge to make good judgment about the appropriateness of the training data. In some cases, this may offer the best guarantee for success. However, the approach is very subjective and prone to biases of the human assessor. Moreover, because of the limited capacity of human experts, the method cannot be applied in settings that involve large-scale dataset.

A common class of problems with generative modeling techniques relates to training challenges. In particular, generative models based on GANs suffer from unstable training. One of the main causes of this issue is the so-called mode collapse problem [127]. This phenomenon occurs when the generator fails to learn the variety in input data and is, thus, able to generate only a particular type of data that consistently beats the discriminator but is inferior

in terms of diversity. Common solutions to this problem include weight Normalization [128] and other regularization techniques [129] as well as architecture innovation [130]. Another serious problem with training generative models is the non-convergence problem [131]. Researchers have attempted to address this difficulty by employing techniques such as adaptive learning rates [132], [133], restart learning [134] and evolutionary optimization of model parameters [135]. These approaches alleviate the problem to an extent but do not completely eliminate it.

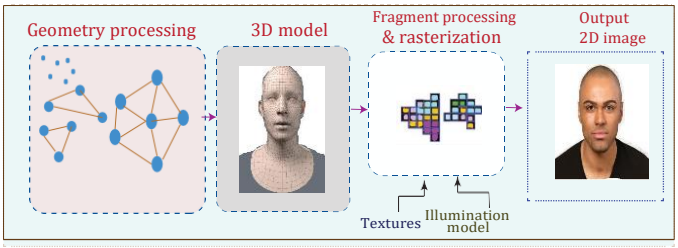

Figure 10. Simplified pipeline of the graphics modeling process.

5 Computer Graphics Modeling

An increasingly promising line of works [136], [137], [138] that aims to address the data scarcity problem exploit computer graphics tools to synthesize training data. Computer graphics tools are capable of creating 2D and 3D objects as well as whole complex scenes. The procedure for synthesizing data using computer-aided design (CAD) techniques involves complex processes such modelling, rigging, texturing, and animating of the generated 3D objects. Game engines provide more advance modeling capabilities that can be used to create large, interactive scenes and virtual environments that span whole cities.

5.1 Approaches to data synthesis based on computer graphics modeling

In this subsection, we discuss the various graphics modeling methods commonly used to generate synthetic to address augmentation problems.

5.1.1 CAD modeling

The methods discussed in Section 4 typically tackle the data augmentation problem as a 2D mapping of a particular image domain to itself by using various transformations to re- create variations of the original data in 2D space. These 2D-based transformation approaches lack semantic 3D grounding, and are, thus, highly superficial in nature and may not adequately represent the actual variations of real-world scenes. Graphics modeling approaches address this limitation by approaching data augmentation as a 3D to 2D mapping (i.e., a mapping from 3D physical world to 2D pixel representation). One important area where computer graphics models are increasingly used is in the area of 3D perception. By modeling the underlying 3D processes of image formation, simulation-based methods produce qualitatively better augmentations compared to pure 2D manipulation methods (for 3D vision). In particular, performance on tasks such as pose understanding, gesture and action recognition are immensely aided by 3D supervision. Techniques based on CAD modeling can also simulate nonstandard visual data such as point clouds (e.g., [139]), voxels (e.g., [140]), thermal images (e.g., [141]), or a combination of two or more of these modalities (e.g., clouds [142]). State-of- the-art computer graphics tools are able to produce fairly realistic visual data for training machine learning models. Three-dimensional game engines are particularly promising in this regard, as they can simulate complex natural processes and generate near-realistic environments under different conditions using real physics models. This capability provides an opportunity to train machine learning models on complex real-world (natural) scenes. Examples of simple 3D objects from the Amazon Berkeley Objects (ABO) dataset modeled using CAD tools are shown in Figure 11. Figure 12 shows realistic indoor scenes from the Hypersim [143] dataset generated, also by CAD tools.

Figure 11. Sample objects from the Amazon Berkeley Objects (ABO) dataset [144]. It is an example of dataset constructed using 3D CAD models. The dataset consists of a large collection of artistically created 3D models of common household objects, and includes the necessary metadata and physically-grounded properties.

CAD modeling methods represent basic scene information using geometric primitives such as triangles and polygons along with their location information and camera properties, global scene information. A renderer translates this representation into a complete 3D scene. Rendering is the process by which images are generated using these basic geometric primitives and additional scene parameters and textures. A typical rendering pipeline is a multi-stage process that composes the primitive geometric entities and scene parameters into more complex objects and scenes (Figure 10). The process involves sequentially creating higher-level representations at every processing stage assembling basic entities into more complex objects until whole scenes are created. The basic elements are specified using analytical formulas. A rendering engine translates the mathematical formulas into their corresponding graphical representations and compute various characteristics such as lighting, colors, and shadows for display. In an OpenGL rendering pipeline, for example, the rendering task consists of vertex shading and assembly—processes that define coordinates posisitions of objects and their attributes and their composition into higher geometric shapes like polygons; geometry shading and rasterization—i.e., converting the geometric informa-

Figure 12. Examples of photorealistic synthetic indoor scenes from the Hypersim [143] dataset.

tion into pixel form; fragment shading – for processing color and texture information. Many advanced integrated development environments (IDEs) have been developed to facilitate 3D modeling. They provide advanced, intuitive graphics user interface (GUI) and easy-to-use toolsets for rendering and editing 3D models. Common 3D modeling tools include simulation and 3D animation software tools such as Cinema 4d, Blender, Maya and 3DMax. These tools provide a means to obtain task-specific data in situations where available data does not meed the requirements of the target task. For instance, Hattori et al. [145] employ 3DMax to synthesize data for human detection tasks in video surveillance applications where task-specific data may not readily be available. The approach allows the generated data to be customized according to the specific requirements of a scene (e.g., scene geometry and object behavior) and surveillance system (i.e., camera parameters). The tools have also provided a means to create large-scale datasets for generic applications. Examples of large-scale synthetic datasets obtained from 3D CAD models include ShapeNet [146], ModelNet [147] and SOMASet [148] datasets. Some of the most important datasets created using 3D modeling tools are described in Section 8.

5.1.2 Synthetic data from 3D physics (game) engines

While CAD tools are primarily used for creating 3D assets, game engines provide tools to manipulate the generated 3D objects and scenes in nuanced ways within virtual environments. They typically come with built-in rendering engines like Corona renderer, V-ray and mental ray. Advanced game engines such as Unity3D, Unreal Engine, and Cry Engine can simulate real-world phenomena such as realistic weather conditions; fluid and particle behavior, effects including diffuse lighting, shadows and reflections; object appearance variations resulting from the prevailing phenomena. By randomizing parameters associated with these phenomena, sufficient data diversity can be achieved. Besides visual perception, simulated environments based on 3D game engines can serve for a broad range of applications. They are particularly suitable for training models in domains like planning, autonomous navigation, simultaneous localization and mapping (SLAM), and control tasks. Figures 13 and 15 show sample scenes from Carla [1] and AirSim [149], [150], respectively. Both tools are created from Unreal Engine. Figure 14 shows the different sensing modalities that can be obtained from Carla.

Because of the advanced manipulation capabilities of modern game engines, recent synthetic data generation approaches [151] favor the use of 3D game engines, which are capable of generating complete virtual worlds for not only training neural network models, but also enabling interactive training of elements of worlds using deep reinforcement learning frameworks. For instance, ML-agents introduced in the Unity 3D game engine, provides a framework for training intelligent agents in both 2D and 3D worlds using a variety of machine learning techniques, including imitation learning, evolutionary algorithms and reinforcement learning.

Figure 13. Varying appearance of a sample synthetic scene from CARLA simulator [1] in different weather conditions.

Varol et al. [3] synthesized realistic datasets using Unreal Engine. Their synthetic dataset, synthetic humans for real tasks (SURREAL) has been provided as open-source dataset for training deep learning models on different computer vision tasks. The authors in [3] showed that for depth estimation and semantic segmentation tasks, deep learning models trained using synthetic data generated by 3D game engines can generalize well to real datasets. Jaipuria et al. [152] used Unreal Engine to enhance the appearance of artificial data for lane detection and monocular depth estimation in autonomous vehicle navigation scenarios. As well as generating scenes with photorealistic, real-world objects, they also simulated diverse variability in the generated data: viewpoints, cloudiness, shadow effects, ground marker defects and other irregularities. This diversity has been shown to improve performance under a wider range of real-world conditions. Bongini et al. [141] rendered synthetic thermal objects using U3D's thermal shader and superimposed them in a scene captured using real thermal image sensors. They additionally employ a GAN model to refine the visual

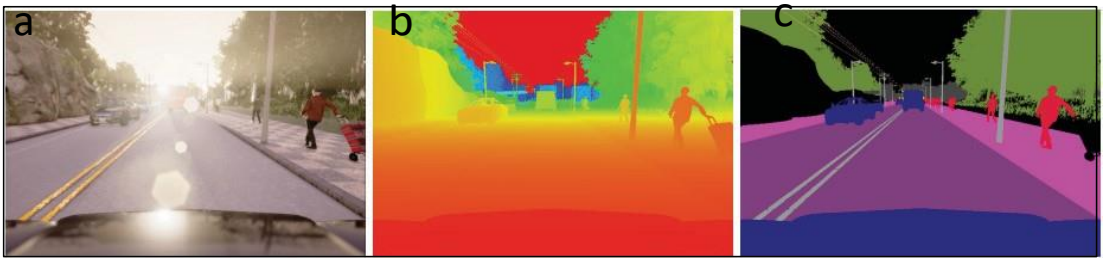

Figure 14. CARLA provides three sensing modalities: traditional vision (a), depth (b), and semantic segmentation (c).

appearance of the rendered images so that they look more like natural thermal images.

Additional plugins have been developed to facilitate the ease of generation of image data and corresponding labels from game engines [153] or from virtual environments [153], [154] developed on the basis of 3D engines. For instance, Borkman et al. [153] introduced a Unity engine extension known as Unity Perception that can be used to generate artificial data and the corresponding annotations for different computer vision tasks, including pose estimation, semantic segmentation, object detection and image classification. The extension has been designed to synthesize data for both 2D and 3D tasks. Hart et al. [155] implemented a custom OpenCV tool in Robot Operating System (ROS) environment to extract frames from simulated scenes in Gazebo platform and generate their corresponding labels. Similarly, Jang et al. [154] introduced CarFree, an open-source tool to automate the process of generating synthetic data from Carla. The utility is able to generate both 2D and 3D bounding boxes for object detection tasks. It is also capable of pixel-level annotations suitable for scene segmentation applications. Carla [1] provides a python-based (API) for researchers and developers to interact with and control scene elements. Mueller and Jutzi [156] utilized Gazebo simulator [157] to synthesize training images for pose regression task. Kerim et al. [158] introduced the Silver framework, a Unity game engine extension that provides highly flexible approach to generating complex virtual environments. It utilizes the built-in High Definition Render Pipeline (HDRP) to enable control of camera parameters, randomization of scene elements, as well as control of weather and time effects.

5.1.3 Obtaining data directly from video games

Some recent works, for instance [159], [160], [161]), have focused on directly extracting synthetic video frames from scenes of video commercial games as image data for use in training computer vision models.. To achieve this, appropriate algorithms are used to extract and label random frames of video sequences by sampling RGB images at a given frequency during game play. Since modern video games are already photorealistic, the qualitative characteristics of image data obtained this way is adequate for many computer vision tasks. Shafaei et al. [159] showed that, for semantic segmentation tasks, models trained on synthetic image data obtained directly through game play can achieve comparable generalization accuracies as those trained on real images. Further refinements by means of domain adaptation techniques to bridge the inherent semantic gap between real and synthetic images results in better performance than models trained on real image datasets.

Since it is generally more challenging to obtain relevant data for video object detection and tracking tasks than for other computer vision applications, generating data from video games has emerge as a promising workaround to alleviate the challenge (e.g., in [161], [162], [163], [164]). The main advantage of this approach is the possibility of utilizing off-the-shelf video games without strict requirements for the resolution of the captured images. Its main disadvantage is that video games contain general environments that are not tailored for specific computer vision applications. Also, the visual characteristics of images from game scenes may not be optimized for computer vision tasks. Moreover, there is generally a lack of flexibility in the data generation process as the user can exercise very little control over the scene appearance and cannot change scene behavior as needed; since scenes in the video sequences are fixed, users are not able to introduce factors of variability (e.g., arbitrary backgrounds, objects and appearance effects) into the scene. In contrast, approaches such as [158], [163], [165], [166] that synthesize scenes from scratch using 3D physics engines bypass these limitations, but require enormously long time and are labor-intensive.

5.1.4 Obtaining synthetic 3D data through scanning of real objects

Another technique [138] to alleviate the laborious work required in synthesizing dense 3D data from scratch using graphics modeling approaches is to leverage special tools such as Microsoft Kinect to capture relevant details of the target objects. This is accomplished by scanning different views of the relevant objects at different resolutions and constructing mesh models from these scans. With this approach, the desired low-level geometric representation can be obtained without explicitly modeling the target objects. In the simplest case, the 3D representation can be obtained with depth cameras to capture multiple views of the object. Suitable algorithms such as singular value decomposition (SVD) [168], random sample consensus (RANSAC) [169] and particle filtering [168] are then used to combine these multiple images into a composite 3D model. The approach proposed in [169], [170], [171] utilize RGBD cameras as 3D scanners for extracting relevant appearance information from object. Figure 16 shows sample frames from the OpenRooms dataset [167], a dataset created from 3D-scanned indoor scenes—ScanNet [172].

A common practice is to construct basic articulated 3D models from the 3D scans which are further manipulated

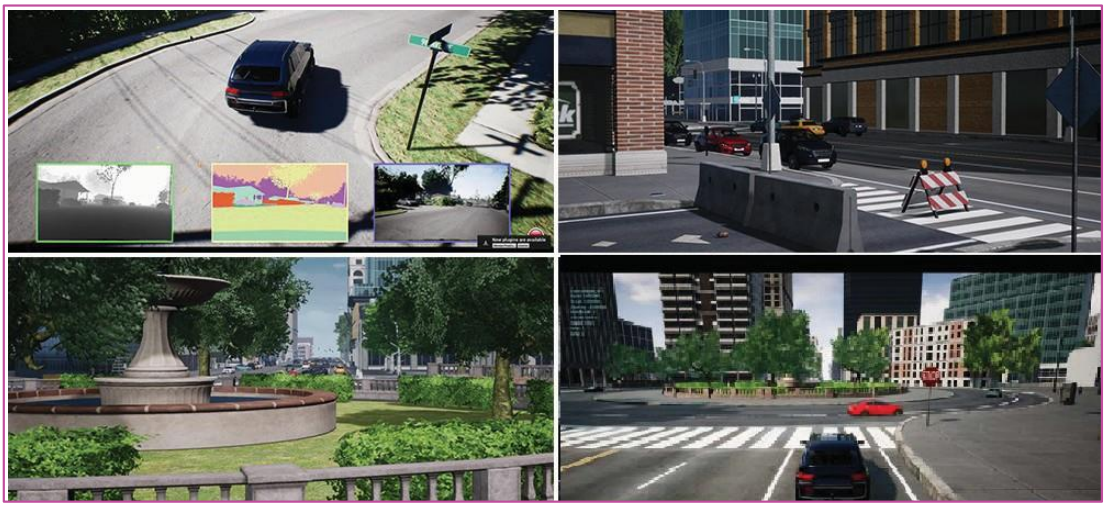

Figure 15. Sample scenes from Microsoft AirSim [149], a synthetic virtual world for training unmanned aerial vehicles (UAVs) and autonomous ground vehicles. The views show typical urban environments for autonomous driving.

within 3D modeling environments to create variability. The approach allows to incorporate only the relevant objects into already modeled 3D scenes. The use of real 3D scans of objects also helps to achieve physically-plausible representations of the relevant objects. Furthermore, visual effects such as lighting, reflections and shadows cannot easily be manipulated by conventional 2D image transformation methods. Therefore, these techniques are vital in situations where it is necessary to train deep learning models to be invariant with respect to these visual phenomena. For instance, Chogovadze et al. [173], specifically employ Blender-based light probes to generate different illumination patterns to train deep learning models robust to illumination variations. Vyas et al. [169] employed a RGBD sensor to obtain 3D point clouds and then used a RANSAC-based 3D registration algorithm to construct the geometric representation from the point cloud data. The authors obtained an accuracy of 91.2% on pose estimation tasks when trained using the synthetic dataset, albeit with some domain adaptation applied. It must, however, be noted that this method can only be used in situations where access to the target domain data is possible.

5.1.5 Combining synthetic with real data

Because of the complex interaction of many physical variables which are difficult to capture using computer graphics methods, some researchers suggest using synthetic data simultaneously with real data. A few works suggest using synthetic data only as a means for defining useful visual attributes to guide the augmentation process. In [174], for instance, Sevastopoulos et al. propose an approach where synthetic data from a Unity-based simulated environment is used as the first stage of data acquisition process to identify useful visual attributes that can be exploited to maximize performance in a given task before collecting real data. The idea is to leverage synthetic data to provide initial direction for further exploration so as to lower the cost of excessive trial and error experimentation on real data. In Section 6, we present quantitative results on the effectiveness of data augmentation approaches that combine real and synthetic data.

5.1.6 Integrating real and virtual worlds

As discussed earlier, the basic idea of the 3D object scanning approach is to obtain information about target objects as stand-alone data by extracting 3D information of real objects and then utilizing graphics processing pipelines to perform 3D transformations on the skeletal models obtained by scanning. However, in some cases (e.g., [175], [176]), 3D geometry models of objects obtained by scanning the real-world objects are incorporated into more complex, task-relevant 3D scenes created using graphics tools. Integrating scanned objects into such virtual worlds provide effective means to manipulate and randomize different factors so as to simulate complex real-world behavior useful for model generalization and robustness. Synthetic 3D models obtained in this manner can be used to train models on complex 3D visual recognition tasks such as object manipulation and grapping. They can also be rendered as 2D pixel images to augment image data. In some other cases (e.g., [177]), synthetic objects are immersed into real worlds in augmented reality fashion. These real worlds are obtained from camera images or videos. In [177], this approach was shown to outperform techniques based on purely real or synthetic data. The most important data augmentation approaches based of computer graphics modeling are summarized in Table 2.

5.2 Challenges of computer graphics-based synthesis and workarounds

Despite the advanced rendering capabilities of modern 3D modeling tools, the use of graphics modeling tools to synthesize training data has a number of limitations:

- The synthesis of realistic data with high level of detail and natural visual properties including realistic lighting, color and textures is a complex and time-consuming process. This may limit the scope of applications of 3D graphics modeling techniques in data augmentation applications to only simple or moderately complex settings.
- Currently, in many situations, there is no objective means of assessing the quality of artificially gener-

Figure 16. Scanned indoor scenes from ScanNet are used to generate new synthetic 3D data based on the approach in [167]. The reconstructed synthetic scenes (a) can then be rendered with different illumination levels and patterns (b), different materials (c), or different views (d).

Table 2
Summary of data augmentation approaches based on computer graphics modeling techniques

Approach	Description	Sample works
CAD modeling	Simulates 3D objects using geometric primitives	[146]
3D physics engines	Allows to generate and manipulate physically-plausible 3D objects in virtual environments	[3]
Data from gameplay	Generates pseudo-video data directly from video games	[163]
Data from scanners	Produces synthetic 3D data through scanning of real objects	[138]
Real-virtual world integration	integrates scanned 3D objects with CAD or physics engines	[175]

ated data. While some methods have been devised to evaluate synthetic data, these are only applicable in situations where reference images are available for comparison. However, in many of the settings that necessitate the use of graphics models, samples of the target images are generally not available at the model development and testing stages.

- It is often difficult to know beforehand the desirable factors and visual features that are good for performance. Moreover, synthetic data that appear photorealistic to the human observer may not actually suitable for a deep learning model. For instance, while high-fidelity synthetic samples have failed to provide satisfactory performance in some situations, some researchers (e.g., in [160], [178], [179] have obtained good performance using low-fidelity synthetic images.
- The generation process is usually accomplished by a careful modeling process, and not from any natural processes or sensor data obtained from real-world variables. Even with the methods that generate synthetic data by scanning real objects, 3D alignment techniques used for registration also introduce additional imperfections into the scene representation. All these problems exacerbate domain gap problem between real and synthetic data.

As a result of the above limitations, it is often challenging to produce semantically meaningful synthetic environments comparable to natural settings, especially when dealing with complex scenes. Since simulated data obtained by 3D modeling tools are often not perfect, the process of augmentation does not only consist in modeling visual scenes, but also in correcting various imperfections and refining the appearance of the artificially generated data to mitigate the domain gap between synthetic and real data. Indeed, a large number of so-called sim-to-real (sim2real) techniques [180], [181], [182], [183] have been proposed for fine-tuning and transferring synthetically generated graphics-based data to real-world domains. The use of generative modeling techniques (e.g, [105], [184], [185]) as a means to endow simple 3D models with photorealistic appearance and desirable visual characteristics has recently gained attention. These approaches provide a more practical and cheaper means for generating extra training data by leveraging unlabeled image data and GANs to introduce hard-to-model, real-world visual effects to simple computer graphics images. In [105] Sixt et al. proposed to learn augmentation parameters to enhance the photorealism of synthetic 3D image data using a large set of unlabeled, real-world image samples. Dual-agent generative adversarial network (DAGAN) has been proposed to enhance the photorealism of synthetic, rudimentary facial data generated by 3D models [106]. Atapour and Breckon in [186] employ a CycleGAN model to refine the appearance of synthetic data which resulted in improved performance. Instead of employing GAN to perform visual style transfer in order to refine synthetically

generated data, Huang and Ramanan in [187] propose to produce a rather large volume of synthetic pedestrian data and then select the most realistic poses through adversarial training.

Rich geometric features inherent in synthetic data have also been used to refine simple 2D images. An important class of refinement methods relies on image-to-image translation models to transfer novel appearance and geometry information (e.g., new poses of objects, occlusion patterns, etc.) from data sources where these views are prevalent to new datasets. For instance, Liu et al. [82] utilize a GAN model to increase the diversity of human poses in a new dataset by transferring pose information from the motion analysis and re-identification set (MARS) dataset [188] where rich pose variation is present. In contrast, some approaches [189], [190] propose to explicitly perform the necessary transformation analytically. Ma et al . [190] employ a deep neural network architecture that combines a GAN and a U-Net sub-models to generate new training samples conditioned on person images and the corresponding poses. The U-Net model reproduces new images of the reference image in the desired poses by explicitly utilizing the specified pose information. A GAN model is then used to correct any artifacts that may result and to refine the general appearance of the person in the desired pose. Similarly, Chen et al. [189] employ explicit warping operations to synthesize new shapes and poses for pedestrian datasets. The concept of transferring realistic views to more rudimentary visual data is extremely useful in person or pedestrian detection [187], [191] and tracking [80], [81], as well as person re-identification tasks [82], [83], [84].

5.3 Augmentation for tasks not requiring realistic data

As noted in the preceding paragraphs, many recent image synthesis approaches focus on generating realistic data for training machine learning models. This process is often tedius and sometimes requires expensive third-party tolls to acomplish. However, in some cases –for example, depth perception [192], optical flow [179] and ego-motion estimation [193], disparity learning for binocular (stereo) vision [194], [195] – the perceptual quality of images is not important for visual understanding. Since the problem of modeling 3D data with realistic geometry and appearance is often challenging, models designed to solve these tasks can easily leverage easy-to-model, non-photorealistic 3D data synthesized with computer graphics tools. Shotton in [192] trained a machine learning model on non-photorealistic 3D data to generate depth maps for human posture recognition. The model produced good results when trained on perceptually poor quality synthetic data alone. Fischer [179] trained a deep learning model (FlowNet) on the Flying Chairs dataset, a collection of unrealistic 3D models of chairs, for optical flow. Ilg et al., observed in [196] that the less realistic flying chairs dataset produces better results in optical flow estimation than [194], which contains more photorealistic 3D training data. The not-so-realistic FlyingThings3D dataset [194] has also proven effective in training deep learning models for optical flow and scene flow tasks [193], [195].

6 Neural rendering

Another common way to synthesize new training data for visual recognition tasks is by neural rendering. The aim of neural rendering is to realize the scene rendering process using deep learning models. Unlike traditional scene rendering based on 3D graphical modeling, the neural rendering process can be accomplished in both forward and backward directions. In the forward direction, 2D images are generated from 3D scenes and additional scene parameters. In the backward direction, the pixel image is translated into a realistic 3D scene. The pipeline for this process is depicted in Figure 17.

6.1 Differential neural rendering

Because the rendering process is inherently non-differentiable, its incorporation in deep neural networks is severely constrained. Differentiable rendering is an approach to overcome this challenge by formulating the scene modeling process as a differential problem that can be seamlessly incorporated into deep neural network pipelines and trained end-to-end. To achieve this, numerical techniques have been proposed to find approximate derivatives of rendering operations that can be used in gradient-based algorithms such as backpropagation to optimize scene parameters. Some earlier works such as [198] and [199] embed real 3D graphics components as renderers for the forward rendering, and accomplish reverse rendering by modeling the relationship between the output 2D image and the input scene parameters using approximate analytical functions. An important task in rendering is rasterization – i.e., converting scene representation from continuous raster form into discrete pixel values. Since the rasterization process is inherently non-differentiable, many of the current approaches are focused on developing approximating methods that can handle this process in a differentiable way. Indeed, a large number of differential neural rendering approaches usually restrict the differentiation to rasterization process only. Typically, the process involves estimating the gradients of the rasterization process and using it in a back propagation algorithm to optimize parameters for the rendering task.

Recent works [200], [201], [202] have focused on learning many of the stages of the forward rendering process in an end-to-end manner using deep learning techniques.The use of deep neural network models in the rendering process can help to generate more nuanced scene attributes (objects, environments, realistic textures and noise) or refine existing scene elements to improve the performance of computer vision models when trained on synthetic data.

6.2 Scene representation methods

An important part of the neural rendering process is the representation of scene elements: geometric priors and scene parameters. We discuss the common approaches for scene representation. The main strengths and weaknesses of these approaches are summarized in Figure 18.

6.2.1 Explicit representation of scene primitives

Traditional computer graphics methods use explicit representations that leverage analytical functions to characterize

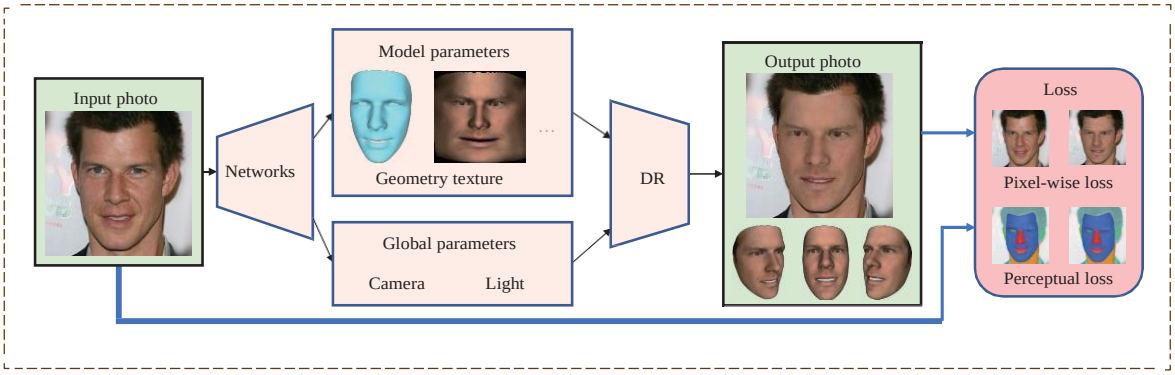

Figure 17. Simplified pipeline of the neural rendering process [197]. Reverse rendering allows intermediate 3D scenes and associated scene parameters to be generated from 2D images.

Representation	Main strengths	Weaknesses
Point cloud	Low memory requirements	Low accuracy of scene topology information
voxel	More accurate with less processing, simplicity	High memory footprint
mesh	Provides more grounding (i.e., physics-aware scene representation)	High computational cost; Difficulty in describing complex shapes
multimodal	High resolution, more robust to visual artifacts	More complex, high computational demand
Implicit (NN)	Naturally differentiable, low memory requirements	Lack of grounding
Hybrid (Explicit + NN)	Robustness, flexibility and multipurpose	Complex architecture

Figure 18. A summary of the main advantages and disadvantages of geometric prior representation approaches used in differential neural rendering pipelines. Here, the designation NN stands for neural network.

various visual attributes of scenes. In general, most neural rendering methods use mesh representation to describe geometric scene properties. In addition to these methods, special techniques have also been developed to handle other image modalities using voxels [203], [204], mesh [198] and point cloud [205] data. Baek et al. [206] used the 3D mesh renderer developed in [198] to synthesize 3D hand shapes and poses from RGB images.

While approaches based on explicitly modeled geometric priors can leverage readily available 3D CAD models, they are typically limited by imperfections in the underlying models. Additional processing is usually needed to meet the requirements of specific tasks. Thus, rendering methods based on explicit representation pipelines can be adversely impacted by scene capture process and modeling time. Besides, these approaches often require the use of proprietary software tools at some stages of the development process, making accurate scene representation difficult and costly to obtain, and thereby limiting their scope of application. To achieve competitive performance, highly detailed geometric primitives and accurate scene parameters are required, along with sophisticated rendering methods. The process of modeling these elements is therefore extremely tedious. Because of these challenges, some works propose to deeply learned to compose scene primitives rather than explicitly modeling all elements from scratch

6.2.2 Implicit (neural) Representation

Unlike real 3D primitives whose construction is manual and laborious, their pure neural counterparts are generated automatically and can be constructed using less human effort, albeit with long training schedules. However, their ability to model 3D scene structure is directly dependent on the representation power and capacity of the underlying neural network.

To improve the effectiveness of approaches designed using scene prior representation some works [204], [207], [208] utilize deeply learned priors as auxiliary elements to refine the accuracy of explicitly modeled priors. Thies et al. [204], for instance, proposes to alleviate the burden of rigorous modeling of textures by incorporating so-called neural textures, a set of learned 2D convolutional feature maps that are obtained from intermediate layers of deep neural networks in the process of learning the scene capture (process). The learned textures are then superimposed on the geometric priors (i.e., 3D mesh) used for rendering. The approach allows course and imperfect 3D models without detailed texture information to leverage artificially generated textures to generate high-quality images. In [208], point-based neural augmentation method is proposed to enrich point cloud representations by leveraging learnable neural representations. Similarly, Liu et al. in [204] propose a hybrid geometry and appearance representation approach based on so-called Neural Sparse Voxel Fields (NSVF). The

method combines explicit voxel representation with learned voxel-bounded implicit fields to encode scene geometry and appearance. While the above studies [204], [207], [208] employ learnable elements to refine scene primitives explicitly constructed from scratch, some recent works [200], [201], [202], [209], [210] have proposed to learn scene models – including geometry and appearance – entirely by using learnable elements. These representations are commonly learned using 2D image supervision (as depicted in the reverse rendering process of Figure 17). Most recent implicit neural representations commonly use Neural Radiance Fields or NeRFs [201], [209], [211], [211], an approach that employs neural networks to learn a 3D function on a limited set of 2D images to synthesize high-quality images of unobserved viewpoints and different scene conditions based on ray tracing techniques.

Currently, the photorealism of scenes generated using approaches based solely on learned representations have not matched those generated using explicit and hybrid representations. Duggal et al. [212] suggest that the problem is as a result of a lack of robust geometric priors in neural representation. They then propose an encoder sub-model to initialize shape priors in latent space. In order to guarantee that the synthesized shapes retain high-level characteristics of their real-world counterparts, high-dimensional shape priors realized with the aid of a discriminator sub-model. This serves as a regularizer of the shape optimization process. More recently, implicit neural rendering techniques have been used within generative modeling frameworks based on VAEs [213], [214] and (GANs) [48], [49], [215] to enable 3D-aware visual style transfer. These have improved the results of data synthesis achievable with either neural rendering or generative modeling techniques alone. For neural rendered scenes, the use of generative modeling helps to easily adapt visual contexts as needed. On the other hand, while generative modeling methods alone have been used to successfully model 3D scene, they often lack true 3D interpretation sufficient for complex 3D visual reasoning tasks [216].

6.3 Approaches and use-cases of neural rendering in data augmentation

For the purpose of data augmentation, there are three broad scenarios in which neural rendering is commonly applied:

- 2D to 2D synthesis – situations where one synthesizes additional pixel (2D) data using 2D supervision
- Scene reconstruction (2D to 3D)–3D visual understanding tasks where 3D dataset is inaccessible, and it is required to synthesize 3D objects or scenes using available 2D image data
- 3D to 2D synthesis–tasks where 2D images are generated from 3D assets

6.3.1 2D to 2D synthesis

A straight-forward application of the data synthesis architecture presented in Figure 17 is to generate new 2D image data with desired visual attributes from a given 2D input image. In this case, the task of the neural rendering process is to apply appropriate transformations in the intermediate 3D representation before the final rendering stage to generate desired 2D output. The basic idea is to disentangle pertinent factors of image variation such as pose, texture, color and shape. These can then be manipulated in an intermediate 3D space before mapping into a 2D space during the rendering process. Such an approach facilitates a more semantically meaningful manipulation of various scene elements and visual attributes. In data augmentation, this may be necessary when synthesizing different poses of objects in a scene [222] or when generating novel views from a single image sample (e.g., in [223]) or when introducing lighting effects to global scene appearance in order to encode invariance to these variations. These transformations are usually difficult to realize accurately with 2D operations that lack the 3D processing stage.

6.3.2 2D to 3D synthesis (scene reconstruction)

Another important application of differentiable rendering related to data augmentation is in scene reconstruction, i.e., the conversion from 2D image to 3D scene. Scene reconstruction techniques typically rely on inverse graphics principles. In this case, a 2D image is used to recover the underlying 3D scene. Important scene parameters such as scene geometry, lighting, camera parameters, as well as object properties such as position, shape, texture and materials are also estimated in the reconstruction process. This is useful in applications that require 3D visual understanding capabilities. Examples of such applications include for high-level cognitive machine vision tasks such as semantic scene understanding, dexterous control and autonomous navigation in unstructured environments. Tancik et al. [202] recently proposed Block-NeRF, a technique to enable the synthesis of large-scale environments (see Figure 21) . They addressed current limitations of NeRF-based models by dividing the scene representation into distinct blocks that can be rendered independently in parallel and combined to form a holistic contiguous virtual environment for training machine learning models on navigation tasks. These appearance modifications can also be applied in a blockwise manner, where smaller regions corresponding to individual NeRFs are updated separately.

Scene reconstruction methods have also been widely used to improve the quality of medical image capture in application like MIR (e.g., [224], [225] and CT (e.g., [226], [227], [228], [229], [230]). For instance, the works in [226], [227], [228] propose using implicit representations to increase the resolution of otherwise sparse CT images. Gupta et al. [229] employ NIR in an image reconstruction model, known as NeuralCT, to compensate for motion artifacts in CT images. Their approach does not require an explicit motion model to handle discrepancies resulting from patient motion during the capture process.

6.3.3 3D to 2D synthesis

Pixel image synthesis from 3D scene is a reverse process of scene reconstruction. It is aimed at learning the 3D scene and mapping it to a 2D space with the help of neural rendering techniques. The idea is to utilize 3D models designed by traditional graphics tools or game engines to generate 2D pixel images. The process is quite simple: given a set of primitive 3D geometric priors and corresponding scene

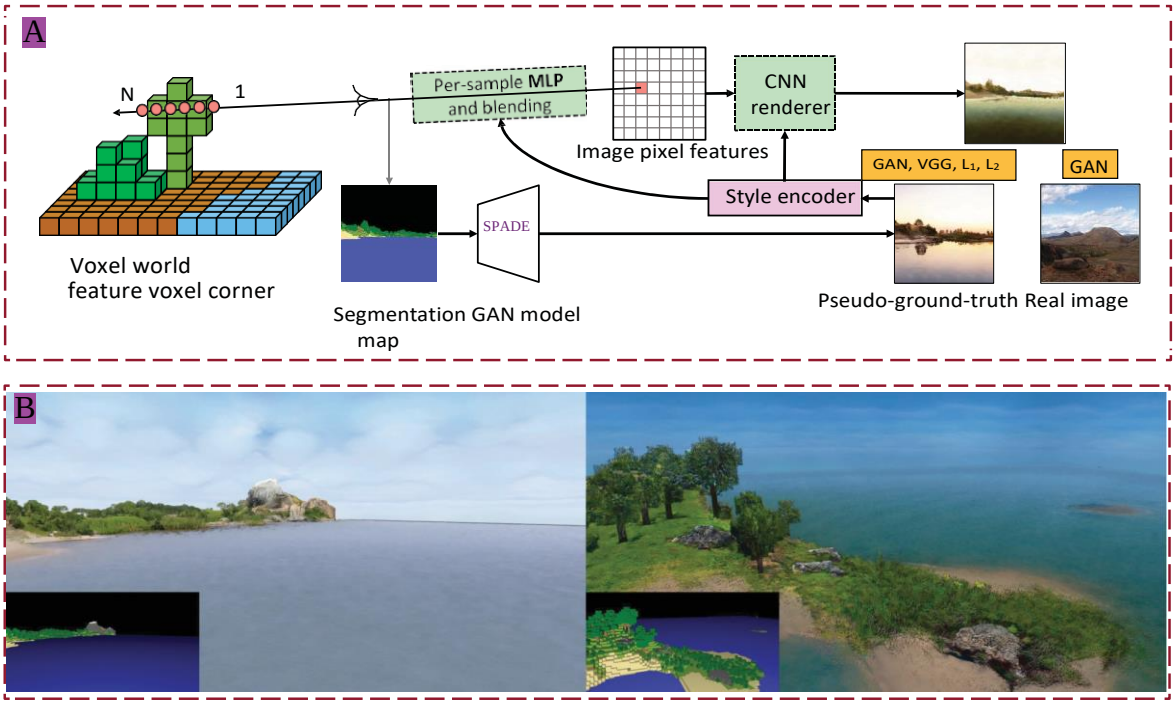

Figure 19. A: General architecture of GANcraft [217],a NeRF framework for generating realistic 3D scenes from semantically-labeled Minecraft block worlds without ground truth images. It employs SPADE, a conditional GAN framework proposed by Park et al. in [218] to synthesize pseudo-ground truth images for training the proposed NeRF model. B: Sample Minecraft block inputs (insets) and corresponding photorealistic scenes generated by the model. .

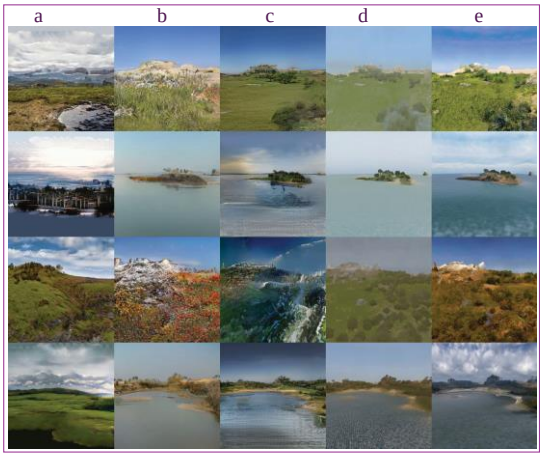

Figure 20. Visual comparison of scenes generated by different GANs and NeRF models. Each of the rows represents a unique scene, while the scenes in different columns correspond to different style-conditioning images. The specific models used here are: (a) MUNIT [219], (b) SPADE [218], (c) wc-vid2vid [220], (d) NSVF-W [221], and (e) GANcraft [217].Images are taken from [217]. .

parameters, the task is to obtain a 2D pixel image. The 3D to 2D subtask can be seen within the differential diagram in Figure 17. Unlike the 2D to 2D synthesis methods discussed earlier that require pixel images to synthesize training data, 3D to 2D synthesis methods directly generate 2D images from 3D assets.

A common application of differential rendering is to employ deep neural network models to generate 2D images and to provide flexibility in applying various 3D-like transformations in synthetically generated 2D image data. For instance, differential renderers can utilize scene elements such as 3D geometry (e.g., vertices of volumetric objects), color, lighting, materials, camera properties and motion to faithfully synthesize unobserved pixels images. Variability in the synthesized images is achieved by manipulating individual factors that contribute to the scene structure (geometry) and appearance (photometry). Thus, the process provides a way to control specific objects in the scene (e.g., modify object scale, pose or appearance) as well as general scene properties. This can help to incorporate more physically-plausible semantic features into synthesize than with traditional image synthesis approaches that lack 3D grounding.

6.4 Limitations of neural rendering methods and possible workarounds

Neural rendering approaches allow to bypass the tedious and laborious process of manually constructing computer graphics models of real world objects. They provide automated way for modeling complex physical processes and scenes. State-of-the-art differential neural rendering techniques make it possible to predict the spatial structure of a complex scene pixel images. They can be used to restore 3D information for a particular 2D image. Modern approaches based on NeRF models such as PixelNeRF [211], DS-NeRF [231] and IBRNet [232] can even generate diverse scenes in various lighting conditions and poses from just a few or a single RGB image. However, training such models requires enormously large quantities of task-relevant data. These approaches generalize poorly to unseen data. Since, in many practical settings, labeled datasets for these tasks may be extremely challenging to produce, many works [207], [208], [233], [234] incorporate traditional graphics pipelines into end-to-end deep neural networks to view-consistent

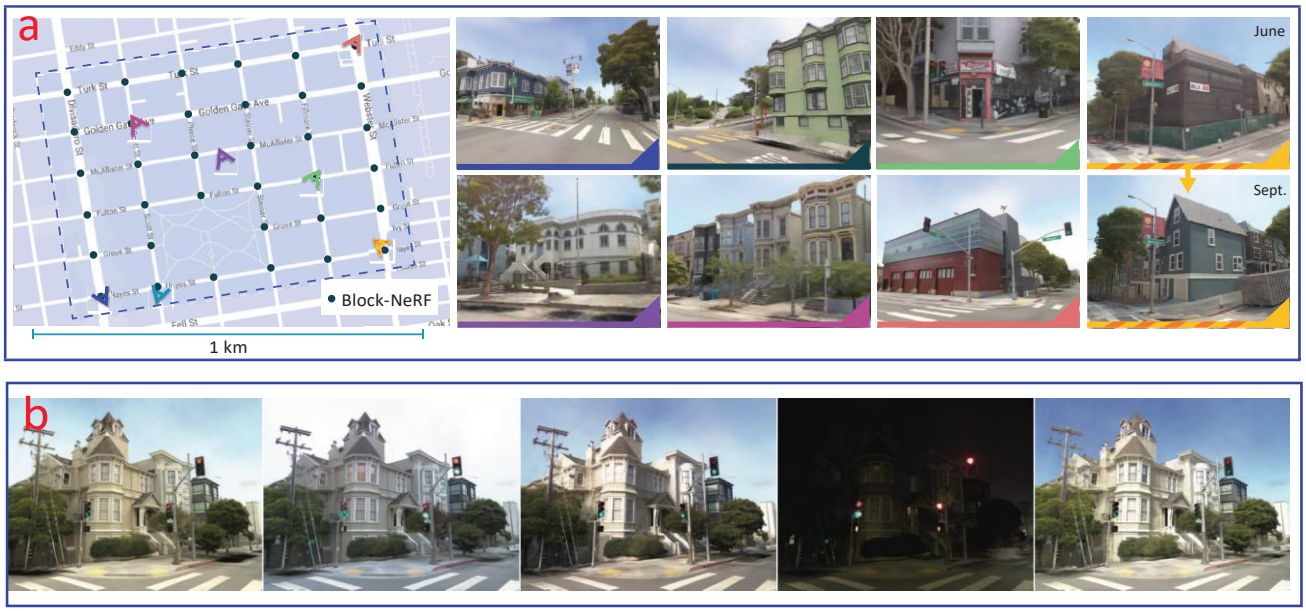

Figure 21. State-of-the-art neural rendering models such as Block-NeRF [202] allow large 3D environments to be generated from sparse 2D views. In multiple neural radiance field models combine to encode large. Shown here is an approximately one square kilometer environment rendered by Block-NeRF (a). The authors also provide a way to alter the appearance of the rendered scene corresponding different environmental conditions such as weather and illumination changes (b). .

scenes. Generative modeling techniques [47], [215], [217] have also been suggested as a viable means for providing the necessary supervision in the absence of ground-truth images. Hao et al. in [217], for instance, propose a neural rendering model that allows to generate realistic scenes from simple 3D LEGO-like block worlds like Minecraft. Since there are no ground-truth images for these types of inputs that can be used to supervise the training, the authors utilize a pretrained GAN-generated images as pseudo-ground truth instead of real images. Their approach, like [202], also allows flexible user control over both scene appearance and semantic content using appearance codes. This capability is useful for applications like long term visual tracking [235], where scene dynamics impacts severely on performance; the ability to control scene appearance is extremely important to simulate all possible view conditions. The basic structure of GANcraft and sample outputs are shown in Figure 19. In Figure 20 we present a visual comparison of scenes generated by different GANs (MUNIT [219], (SPADE [218] and wc-vid2vid [220]), NeRF (NSVF-W [221]) and a combination of GAN and NeRF (GANcraft [217]).

An important aspect of data synthesis based on neural rendering approaches is the ability to work with different data representations – for example, voxels, raw point clouds, pixels, or implicitly defined data forms based on learned functional description of the physical properties of objects. However, this flexibility also presents difficulties: representations can be very different for the same data and the different forms of representations are not necessarily compatible with one another. Recent techniques [236], [237], [238], [239] allow to fuse or convert between these diverse data representations. However, these methods often lead to the loss of vital information about the scene elements. As a result of these difficulties, training neural rendering models to generate data in these modalities is still a challenging issue. Moreover, relatively small number of datasets exist for training differential neural rendering models on many of the possible representation modalities. Given the rapid growth of interests in neural rendering methods, this problem will be solved in the short term.

Implicit neural representation can also support non-standard imaging modalities such as synthetic aperture sonar (SAS) images [240], [241], computed tomography (CT) [230], [242], [243] and intensity diffraction tomography (IDT) [244]. In addition, these approaches are capable of modeling non-visual information like audio signals [245]. Approaches have also been proposed to leverage multiple visual modalities together with other physical signals to provide complementary information about the physical properties of objects [245], [246], [247]. This can support multi-sensory intelligence in applications such as robotics, virtual, augmented, extended and mixed reality, and human-computer interaction. However, adding these additional elements means that data requirements grow even more exponentially, making it challenging to accomplish realistic results based on only learnt models. Many works (e.g., [248], [249], [250]) propose integrating task-specific priors to improve the ability of neural rendering methods that model complex scenes and interactions to generalize better to unseen contexts. Incorporating strong priors often requires special mathematical formulations within neural network models for encoding spatial features and natural behavior of objects in the 3D world, leading to increased computational complexity and cost. For this reasons, state-of-the-art neural rendering frameworks such as [248], [250], [251]) are inherently complex, limiting their ability to represent large-scale scenes. Block-NeRF [202] proposes a modular framework that allows separate NeRF modules to represent individual regions of a scene (shown in Figure 21). This allows large-scale 3D scenes to be represented and efficiently manipulated using sparse 2D images.

7 NEURAL STYLE TRANSFER

Neural style transfer (NST) is a method for synthesizing novel images similar to GAN-based style transfer. However, in contrast with generative modeling approaches, neural style transfer exploit conventional feed forward convolutional neural networks for the synthesis.

7.1 Principles of neural style transfer

Neural style transfer involves first learning representations for the content and structure of the original images, and the style of a reference samples. These representations are then combined to generate new representations in the style of the reference images while at the same time maintaining the content and structure of the original image. The method leverages the hierarchical representation mechanism of deep convolutional neural networks (DCNNs) to flexibly generate novel images with various appearance artifacts and styles. An illustration of the basic principles of the concept is shown in Figure 22. Since shallower layers in CNNs encode low-level visual features such as object texture, lines and edges [252], while deeper layers learn high-level semantic attributes, different augmentation schemes can be realized by manipulating the two semantic levels separately and combining them in different ways. In NST, typically, a DCNN model without the fully connected layers are used to extract image features at different levels. Low level features encoded by shallower layers are then extracted and combined with high-level features extracted from a second image. As the second image contributes high-level features, essentially, its semantic content is transferred to the artificially generated image while the first image's visual style is transferred (see Figures 22 and 23).

The original technique for neural style transfer) was first proposed by Gatys et al. in [254] as a way to artificially create different artistic styles in images. Specifically, they altered landscape images taken by digital camera to look like images produced by artworks while still maintaining their semantic meaning. To accomplish this, Gatys et al. [254] remove the fully connected layers of a VGG model [255] and use the convolutional layers to extract visual features at different semantic levels. They then introduce an optimization function that consists of two different loss components – the style and content loss functions – which are controlled separately by different sets of weight parameters. Based on these principles, the authors showed that image content and style information can be transferred independently to different visual contexts by separating the filter responses of shallower and deeper layers of the CNN model. To create a new image with the content of a source image I_c and the style of a target image I_s, they propose to combine the latter processing stages of I_c with the earlier stages of I_s (Figure 22). Neural style transfer has also been extended to video [?], [256], [257], [258] and 3D computer vision [259] applications. Ruder et al. [259] propose an video style transfer technique that applies a reference style from a single image to an entire set of frames in a video sequence. In [260], a neural style transfer approach is combined with a GAN model to generate diverse 3D images from 2D views. The generated image data is then used to improve performance 3D recognition tasks.

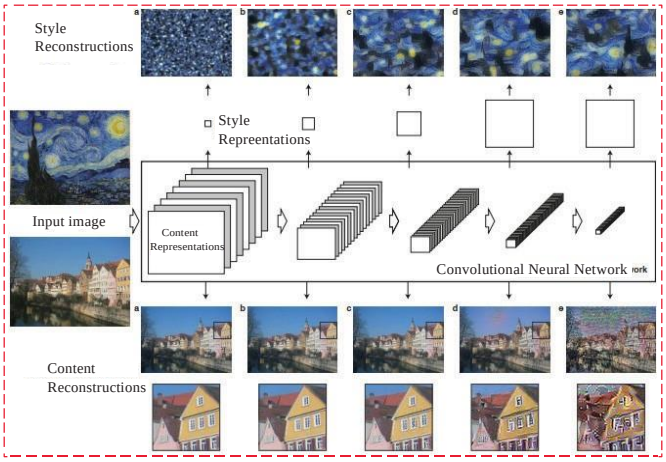

Figure 22. Neural style transfer relies on the hierarchical representation of features in convolutional neural networks to render high-level semantic content of images in different visual styles. In the original work [253] Gatys propose to extract style and content information from different processing stages of convolutional neural networks. The extracted style and content information are then manipulated separately and combined to form new images.

7.2 Neural style transfer as a data augmentation technique

Following the original work [254], many subsequent studies (e.g., [260], [261], [262], [263], [264], [265], [266], [267], [268]) have employed neural style transfer technique as a form of data augmentation strategy to synthesize novel images to extend training data. In many of the works, including the original work that introduced the approach, style transfer is typically applied to synthesize non-photorealistic images. It is reasonable to assume that adding images that are stylized in a non-photorealistic way to the training dataset could still help to reduce overfitting and improve generalization performance. Indeed, a number works have shown this to be the case for many tasks. Jackson et al. [269] demonstrated that augmenting training data nonphotorealistic images (e.g., images synthesized in artistic styles)can significantly improve performance in several different computer vision tasks. Using neural style transfer, Jackson et al. in [269] achieved an improvement in the range of 11.8 to 41.4% over a baseline model (InceptionV3 without augmentation) and improvement between 5.1 and 16.2% over color jitter (alone) on cross-domain image classification tasks. In addition, their method yields a performance increase of at least 1.4% over a combination of seven different classical augmentation types. Instead of transferring a single style per image, the authors in [269] aim to create more random styles suitable for multi-domain classification tasks by randomizing image features (texture, color and contrast). They used a so-called style transfer network to obtain random style attributes by sampling from multivariate distribution of low-level style embeddings.

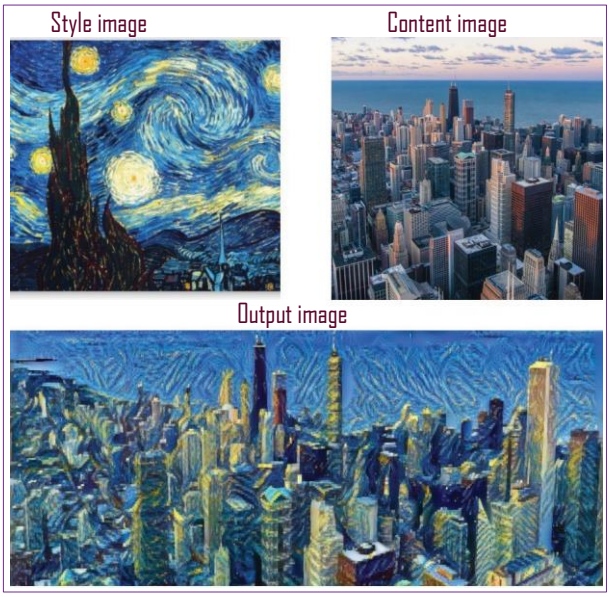

Figure 23. An example of artistic style transfer by neural style transfer (NST) technique (image courtesy [270]). The NST model reproduces the content of the content image in the style of the style image.

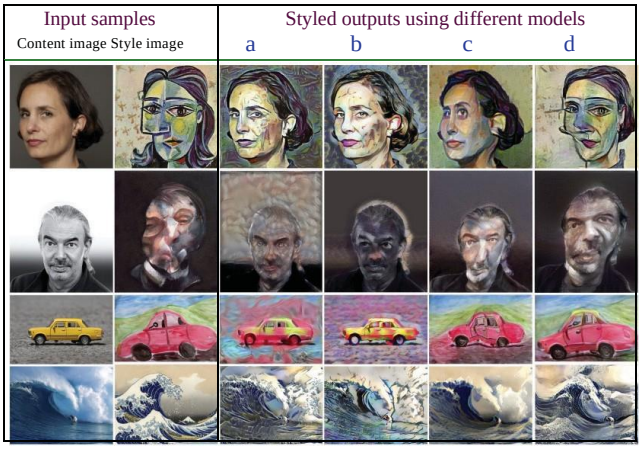

Figure 24. Comparison of results of different artistic style transfer approaches: Gatys et al. [253] (a), Huang and Belongie [271] (b), Kim et al. [272] (c) and Liu et al. [273] (d).

7.3 Approaches to data augmentation using NST

In the literature, three main approaches to data augmentation based on neural style transfer can be distinguished. These are (1) approaches aimed at increasing data variability in order to improve generalization performance, (2) approaches aimed at synthesizing photorealistic image data, and (3) approaches that seek to selectively modify only the most important (in a given context) image parts. We briefly discuss these three neural style transfer implementations in subsections 7.3.1 to 7.3.3. In Table 3, we present a summary of the main features of these approaches.

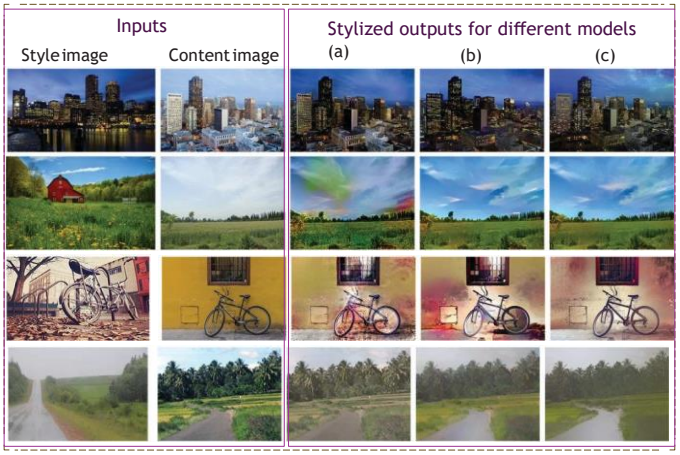

Figure 25. Visual comparison of different NST-based photorealistic stylization methods. We show results for Li et al. [274] (column a), Luan et al. [275] (column b) and Yang [276] (column c).

7.3.1 *Inproving data diversity with artistic stylization*

Since higher data diversity generally favor better generalization accuracy, many works [281], [285], [286], [287] have recognized the importance of increasing diversity of styled images and have introduced different techniques to achieve this. Wang et al. [281] propose an approach that relies on the perturbation of deep convolutional features by means of random noise injection. Another class of approaches [263], [280], [288], [289] aim to address this problem by developing methods that allow many different styles to be transferred simultaneously. For instance, in [280], sets of convolution feature maps in intermediate CNN layers are grouped into filter banks, with each group uniquely encoding one specific style (see Figure 27). Dumoulin et al. [289] propose a simple normalization approach – named conditional instance normalization – that leverages the mean-variance statistics of convolutional features to enable as many as 32 different styles to be transferred to a specified content image. Instead of explicitly extracting style information from reference images, Georgievski [263] proposes an approach in which different styles can be independently specified as input style embeddings. First, a large set of generic style embeddings are first obtained from a pre-trained neural network (Inception-ResNet-V2) that is trained on large-scale image dataset (in this case, ImageNet). The learned styles are then used to construct a multivariate normal distribution of styles that can later be applied to content images. The application of styles is accomplished through an encoder-decoder subnetwork, which the author termed style transformer. Huang and Belongie [271] extend the concept of [289] by modifying the instance normalization operation in such a way that it is able to encode arbitrary style, as opposed to transferring a pre-defined set of style as in [289]. They introduced a new layer, adaptive instance normalisation (AdaIN), in place the conditional instance normalization layer in [CIN]) whose function is to align the mean and variance of convolutional features of style and content images. Gu et al. [288] propose to reshuffle deep feature maps of the style image so as to synthesize images with arbitrary styles. Data augmentation schemes based on such stylizations would result in more ro-

Table 3
The main ways to perform data augmentation using neural style transfer technique

Approach	Rationale for data augmentation	Sample works
Artistic style transfer	Simulates noise to provide variability in training data so as to decouple models from being tied to specific visual features. It can also help encode invariance to image textures.	[261], [263], [269]
Photorealistic style transfer	Simulates desired real-world visual effects such as illumination and weather effects in training data.	[277], [278], [279]
Multi-style transfer	Increase the diversity of training images to improve generalization performance	[280], [281]
Patch-level stylization	Provides a means to manipulate specific image content (e.g., objects of interest in object detection task)	[267], [282], [283]

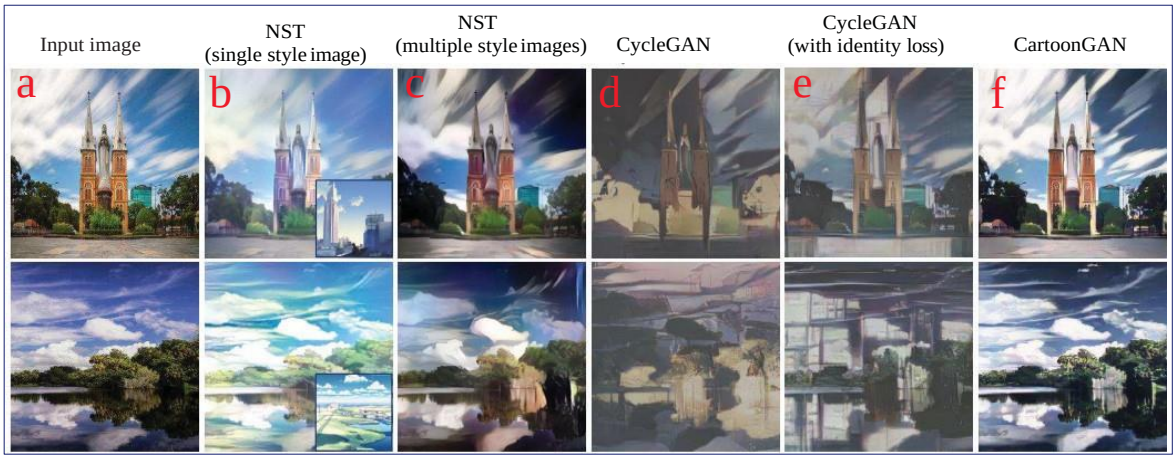

Figure 26. Qualitative comparison of neural style transfer (NST) and GAN models (results courtesy [284]). The different models aim to perform artistic stylization of the original images while maintaining all content. The styles were obtained from images extracted from cartoon videos. The NST model in column b has been trained on style image that has similar content as the input image (shown inset). The rest of the columns (c, d, e and f) show results of styling with an aggregate of 4,573 cartoon images. Note that the quality of stylization is defined by how well the semantic content is preserved, including the clarity of edges. The specific models used are: NST by Gatys et al. [253] (column b and c), CycleGAN [45] (column d), CycleGAN with identity loss (column e), and CartoonGAN [284] (column f).

bust representations akin to style randomization in rendered images.

In addition to these approaches that aim to improve image synthesis by introducing new architectural methods of encoding styles within CNN layers, a number of works [290], [291] are primarily focused on developing new loss functions that allows to better transfer more diverse and fine-grained features than earlier approaches based on style and content loss functions [254]. Li et al. in [285] introduced a special loss function, known as diversity loss, to encourage diversity. Wang et al. Luan et al. [275] propose a loss function based on smoothness estimation.

7.3.2 Photorealistic data syntheis with NST

Recently, several researchers [275], [292], [292] have introduced methods to enhance the photorealism of images synthesized using NST methods. Photorealistic styles are stylization schemes that render the resulting image as visually close as possible to real images captured by real image sensors. Photorealism is aimed at simulating high-quality data in terms of image details and textures, taking into consideration a range of real-world factors, which may include blurring effects resulting from camera motion, random noise, distortions and varying lighting conditions. Figure 24 compares the results of different artistic style transfer approaches. In Figure 25, the outputs of various models that aim to achieve photorealistic stylization are compared. We also compare the artistic style transfer ability of common GAN models with NST methods in Figure 26.

There is also a family of NST approaches that aim to transfer specific visual attributes such as image color [286], [293], [294], illumination settings [295], material properties [296] or texture [297], [298]. Rodriguez-Pardo and Garces in [296] propose a NST-based data augmentation scheme based on transferring properties of images under varying conditions of illumination and geometric image distortions. Gatys et al. in [299] propose to decompose style into different perceptual factors such as geometry, color and illumination which can then be combined in different permutations to synthesize image with specific, desirable attributes. In order to avoid geometric distortions and preserve statistical properties of convolutional features extracted from style images, Yoo et al. [292] introduced a wavelet corrected transfer mechanism that replaces standard pooling operations with wavelet pooling units.

Neural style transfer techniques have also been successfully employed to transfer specific semantic visual contexts to synthetic data. For instance, Li et al. [277] transferred images taken in clear weather to snowy conditions. In [253], images taken in bright day are converted to night images using NST stylization techniques. More advanced neural stylization approaches [278], [279] have been designed to learn the semantic appearance as well as physically-plausible behavior of objects of interest. Kim et al. [278], for instance, proposed a physics-based Neural Style Transfer method to simulate particle and fluid behavior in synthetic images. Their approach allows realistic visual appearance and semantic content of different particular matter to be

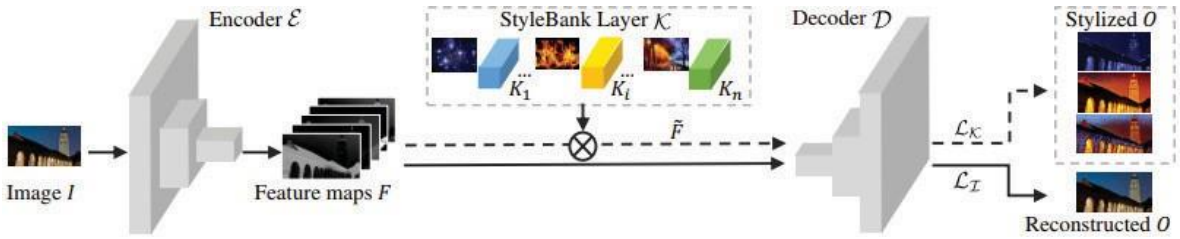

Figure 27. StyleBank [280] is an example of multi-style transfer technique based on an encoder-decoder architecture. It encodes different styles in groups of convolutional filters in intermediate CNN layers that can be selectively applied to specific image content.

learned and transferred from 2D to 3D settings.

7.3.3 Patch-level synthesis

Researchers have also proposed photorealistic synthesis methods [296] based on patch-wise stylization that allow specific image parts or object within images to be selectively altered. Such methods allow a more granular control of the output images. They may enable, for instance, the manipulation of specific objects in a complex scene. This family of style transfer techniques can be useful in applications such as semantic segmentation (e.g., in [282]. Some works have suggested using patch-level styling to achieve photorealism while at the same time reducing the computational demands of conventional (holistic image) methods. Cygert and Czyżewski [267] argue that styling the whole input image could be detrimental to generalization performance. They propose to address this limitation by utilizaing a stylization method that transforms only small patches of input images. To achieve this, they perform a hyperparameter search to find the best patch size for stylization. In [283], Chen et al. proposed a patch-based stylization technique as a means of reducing complexity and improving memory efficiency in the stylization of high resolution images.

7.4 Limitations of NST methods and possible workarounds

The main advantage of neural style transfer is the fact that the approach leverages the natural representation of visual features in CNN layers to synthesize new image data. Consequently, the amount of data and the time required for training models is relatively small. Also, model architectures for neural style transfer are generally simpler compared to other synthesis methods such as generative modeling and neural rendering. Moreover, since NST training is usually more sample-efficient than other synthesis methods, memory requirement is greatly reduced.

One major weakness of datasets created by neural style transfer approaches is that the data is generated by feature-level manipulations and not by intuitive transformation of real data. As a result, it is extremely difficult to account for semantic information about the entities in a scene when applying stylization. This can also lead to noise and unwanted artifacts which may seriously affect the quality of the generated data. Another challenge with non-intuitive, feature level perturbations is the difficulty in maintaining consistency between local and global features [275] when applying styles. These issues require additional (costly) processing of semantic information so as to ensure that realistic data is produced. For instance, recent works have proposed to optionally perform semantic segmentation on the target scene so as to enable context-specific style transfer (e.g., in [275], [300]). This additional processing step may lead to increased model complexity and computational cost. In addition, it can potentially introduce errors, e.g., a result of incorrect segmentation masks, which may harm the accuracy of the stylization.

Also, the fact that the generated dataset is not derived from intuitive, physically grounded manipulation of input data at the sample level may limit the degree to which desired visual attributes can be simulated. Unlike neural rendering and generative modeling techniques that can naturally learn desired visual contexts such as color, contrast and illumination levels, neural style transfer methods require more complex formulations to handle these attributes successfully.

An even more serious limitation of neural style transfer methods is the difficulty of simulating spatial transformations. For this reason, the approach is mainly used for photometric data augmentation tasks. Furthermore, geometry-consistent photometric stylizations (e.g., brightness levels based on viewpoints; generating and placing shadows at the right positions in a scene; and fine-grained style-content consistency under different poses) are difficult to achieve. This can lead to distorted style patterns. The difficulty lies in the inherent principle of the approach: different hierarchical levels of feature maps separately capture style and geometry information and are combined to generate new samples. The decoupling of style from geometry makes it challenging to apply learned styles in a view-dependent manner. Some works (e.g., [273], [301]) have proposed to incorporate explicit spatial deformation models within NST architectures to handle geometric transformations. In [302] Jing et al. utilize graph neural network model to learn fine-grained style-content correspondences that minimizes local style distortions under geometric transformations.

8 Synthetic Datasets

Presently, a wide range of large-scale synthetic datasets are publicly available for training and evaluating machine vision models. We summarized the details of some of the most important synthetic datasets in Table 4. These datasets support a wide range of visual recognition tasks. In addition, they cover many of the synthesis methods explored in this work. In addition, they include the common representation methods.

Table 4
A summary of publicly-available synthetic datasets

Dataset	Synthesis method	Domain	Supported tasks	Dataset size
RTMV [303]	Neural rendering	General scene understanding	View Synthesis Scene reconstruction Pose estimation	300,000
MPI-Sintel [161]	3D animation (video)	Spatio-temporal scene understanding	Optical flow	1628
Objectron [304]	3D data from augmented reality library	Multi-view object recognition	3D object detection	4M
Synthia [164]	3D game engine	Autonomous driving	Depth estimation Scene segmentation Ego-motion	200,000
LCrowd [166]	3D game engine	Crowd analysis	Crowd counting Person detection (in crowd)	20M
Semantic3D [305]	Point cloud data from laser scanner	urban scene understanding	Semantic segmentation	
Synscapes [165]	Neural rendering	Autonomous driving	Pose estimation Depth estimation Semantic and instance segmentation	25,000
ShapeNet [146]	3D CAD	General recognition	3D Perception	51,300
ScanNet [172]	RGB-D capture	Indoor scene understanding	Semantic and instance segmentation	2.5M
Virtual Kitti [162]	3D game engine	Urban scene understanding	Depth estimation Optical flow Object detection and tracking Scene and instance segmentation	21,260
Pix3D [306]	3D scanning and web sources	Object shape retrieval viewpoint estimation	2D-to-3D reconstruction	395
GTA-V [160]	Video game play	Autonomous driving	Sematic Segmentation	25,000
Hypersim [143]	3D CAD	Indoor scene understanding	Semantic and instance segmentation	77,400
PreSIL [307]	Video game play	3D scene understanding	2D and 3D object detection Depth perception Semantic segmentation	50,000
SOMASet [148]	3D CAD	Person re-identification	Object recognition	100,000
SceneNet-RGBD [308]	RGB-D rendered objects from CAD models	Indoor scene understanding	Object detection Pose estimation Depth estimation segmentation	5M

9 Effectiveness of Synthetic Data Augmentation Methods

Many works have demonstrated the effectiveness of synthetic data augmentation techniques. In some cases (e.g., [3], [4], [184], [309]) data generated synthetically leads to better generalization performance than real data. Wang et al. [309], for example, reported several instances where models trained on synthetic data achieve better results on face recognition tasks. Similarly, Rogez and Schmid [4] consistently obtained higher performance with synthetic data than with real data on pose estimation tasks. Applications scenarios where synthetic images have particularly performed better than real data have been settings that do not require high level of photorealism (e.g., depth perception [3]) and pose estimation [4], [184]. This is due to the fact that synthetic images are often "cleaner" (i.e., the do not contain spurious details and artefacts which may be irrelevant to the target task). While some studies have obtained impressive results with training exclusively on synthetic data, many other works show that synthetic data, when used alone, would not always yield the desired performance. For instance, results obtained by Richter et al. in [160] demonstrates that while synthetic data can drastically reduce the amount of real training data needed to achieve optimum performance, by themselves synthetic images do not guarantee good performance. In their study [160], they experimented first with real data and obtained an mean IoU of 65.0%. When the training set was augmented with synthetic data, they were able to improve the mean IoU score by 3.9% (from 65.0 to 68.9%). Indeed, using the augmented data, they achieved comparable performance (65.2%) as with real data (65.0%) using only about one-third of the real data. However, the results were poor and unsatisfactory when only synthetic data was used (43.6%). Rajpura and Bojinov [310] compared the performance of deep learning-based object detectors trained on synthetic (3D-grahics models), real (RGB images), and hybrid (synthetic and real) data. The results showed lower performance on synthetic data (24 mAP) compared to real (28 mAP). However, the addition of the real and synthetic images improves the performance by up to 12% (36 mAP). Similarly, in [177] Alhajia et al. observed that training with augmented reality environment that integrates real and synthetically generated objects into a single environment achieves a significantly higher performance than with either separately. In [311], Zhang et al. observed that expanding the training set by increasing the proportion of synthetic data does not lead to a linear increase in model performance. In fact, for some tasks, the performance flattens out at about

25% composition of synthetic data for the specific cases investigated.

10 Summary and discussion

In the literature, many data synthesis approaches have been proposed. Despite the large variety of techniques, four main classes of approaches can be distinguished: generative modeling, data synthesis by means of computer graphics tools, neural rendering approaches that utilize deep learning models to simulate 3D modeling process, and neural style transfer that relies on combining different hierarchical levels of convolutional features to synthesizer new image data. The first group of approaches – generative modeling – is mainly based on the generative adversarial networks and variational autoencoders. Generative modeling methods allow realistic image data to be synthesized using only random noise as input. Models can also generate outputs conditioned on specific input characteristics that define the desired appearance of target data. The second class of methods, computer graphics approaches explicitly construct 3D models based on manually modeled, primitive geometric elements. Neural rendering methods use deep neural networks to learn the representation of 3D objects and then optionally render these as 2D images. Neural style transfer combines different semantic information contained in different layers of neural networks to synthesis various visual styles. The main strengths and weaknesses of each of these classes of approaches are summarized in Figure 28.

The different data synthesis approaches have unique characteristics that define their scope of application. Neural style transfer method, for example, is highly flexible since the stylization process can be controlled by setting appropriate weight parameters corresponding to different style intensities. This allows low-level transformations to be easily accomplished for a given task. However, despite its simplicity and relative efficiency, the method is generally limited to 2D synthesis. It is also challenging to generate large-scale photorealistic data. Its most important prospect is in enhancing robustness to noise, overcoming overfitting by preventing models from learning specific visual patterns, and encouraging texture invariance. Unlike neural style transfer methods, rapid advances in generative modeling techniques have made it possible to easily create large volumes of artificial images with desired properties for specific tasks. However, these methods typically model objects and scenes in 2D, making it difficult to to transfer encoded knowledge to 3D scene understanding tasks. Also, by ignoring the 3D structure of the real world, achieving physically-grounded object manipulation is a challenging task. Synthesis methods based on 3D graphics modeling overcomes these limitations by providing a means to generate realistic 3D scenes. However, the modeling process is typically laborious, limiting the amount of detail and the range of dynamic attributes that can be supported. Neural rendering can be used to introduce more nuanced details without manually creating the desired visual appearance. In particular, differential neural rendering approaches based on implicit representations allow deep learning models to encode 3D-grounded representation, as well as visual attributes such as color and varying lighting conditions in the artificial generated images. Only in the last few years have there been cost-effective methods for generating large-scale virtual scenes using implicit neural representation. Recently, NeRF been a subject of considerable research. This intensive research has led to the development of new deep learning methods and models that adequately represent 3D data in sophisticated and efficient ways. As a consequence of the development these methods, a wide range of possibilities are emerging regarding the training data for complex tasks like autonomous driving, where the cost of camera-captured data is exorbitant. As well as being able to generate more plausible pixel images, modern differential neural rendering models such as Block-NeRF [202] and Mega-NeRF [312] can synthesize realistic, large-scale 3D videos of entire scenes using only a few 2D images as input. These models basically map pixel space into the context of a continuous 3D scene. This capability is highly promising in complex visual perception tasks that require physics-aware interpretation of input data. Conceivably, in the near future more powerful and efficient NeRF models capable of generating large-scale, fully dynamic and realistic 4D scenes will replace the existing methods for synthesizing training data for applications such as robot navigation, autonomous driving and many 3D perception tasks.

11 Future research directions

The four main classes of synthetic data augmentation methods surveyed in this work are computer graphics modeling, neural rendering, generative modeling and neural style transfer. Overall, judging by recent trends, we expect significant progress in generative modeling and neural rendering techniques and minimal progress in neural style transfer methods. We also expect the applications of these methods in more challenging machine intelligence tasks such as affordance learning, human-machine interaction and extended reality. Progress in this areas will undoubtedly have a profound impact on the development of machine vision and artificial intelligence as a whole. We outline most promising future research directions in the following paragraphs.

a. Modeling multiple sensory modalities in an integrated and adaptive way

Recent interest in generative modeling and neural rendering approaches has led to the development of new hybrid deep learning methods that combine the power of state-of-the-art neural rendering techniques like NeRFs and generative models such as GANs and VAEs to adequately represent both 2D and 3D data in more effective and controllable ways. These approaches primarily exploit conditional GAN and conditional VAE techniques to provide a form of control over the visual properties of the synthesized data, allowing models to learn multiple salient features to enhance the realism of representations. This additional dimension of flexibility and control can be leveraged by future synthesis models to specifically generate more dynamic data whose appearance and properties change adaptively with the visual context. For applications such as high-level perception and long-term scene understanding, the emergence of techniques that allow view-specific attributes to be modified online in response to real-world conditions

Approach	Main advantages	Disadvantages
Generative modeling	• Automates the process of generating synthetic data • Can generate highly photorealistic data without additional effort	• Lack of 3D interpretation • Requires representative examples of the target class • Problems with mode collapse and overfitting can lead to poor results
Computer graphics modeling	• Requires no training data whatsoever • End product fully controllable by developer • Can support physically-plausible behaviors • Can generate very large virtual environments	• Subject to bias and perceptual limitations of human developers • Extremely laborious • Requires extensive domain knowledge • May require the use of proprietary tools
Neural rendering	• Can easily support 2D and 3D tasks • Allows easy manipulation of generated data • Can extend the capabilities of existing 3D models	• Highly complex model architectures • Scene generation capacity limited to small environments • Requires enormous amounts of data
Neural style transfer	• Relatively simple architectures • Low resource requirements • Easier to implement	• Limited in the ability to generate photorealistic data • Can only synthesize 2D images

Figure 28. A comparison of different data synthesis approaches.

would be extremely useful. More generally, the development of new data synthesis methods based on implicit neural rendering and generative modeling techniques will offer opportunities to radically extend the capabilities of state- of-the-art machine vision systems. Because implicit neural representation techniques are spatio-temporally continuous and differentiable, in addition to 3D visual information, they can be used as universal function approximators to model diverse sensory signals as well as complex physical processes of the real world. Future research is expected to produce new and improved ways to enable these diverse sensory modalities and physical properties of objects to be compactly integrated into a kind of universal framework. Incorporating information-rich representation in this manner will undoubtedly help to improve "common sense" reasoning capabilities of intelligent and robotic systems.

b. Towards more effective and efficient representation and training

A major shortcoming with current data synthesis approaches, particularly generative modeling and neural rendering techniques, is the need to use several examples from the target domain to guide the synthesis process, especially when training for 3D scene understanding tasks. Many recent works have focused on achieving more computationally-efficient representation that enables scene data to scale up to city or metropolitan level environments. Application domains such as outdoor scene understanding and autonomous driving particularly require very large continuous scenes which are currently challenging to synthesize owing to the enormous computational resource requirements. Today's large-scale synthetic environments are typically realized by stacking several smaller, image-level scenes together, where each constituent "mini scene" is encoded by a dedicated NeRF model. Obviously, this is not the most natural and effective way to represent scenes. A large amount of future research efforts will increasingly focus on achieving more sample-efficient training. One of the most promising research goals is the development of unconditional 3D-aware data synthesis techniques that allow to generate high-quality, realistic 3D synthetic scenes without the need for reference images. New synthesis methods will also allow to utilize more parse representations to synthesize realistic data with detailed visual information. The integration of neural radiance fields (NeRF) models into state-of-the-art generative modeling architectures will increasingly provide better ways to achieve more compact and unified differentiable representations of complex scenes. New techniques resulting from further breakthroughs in this line of works, in the not-so-distant future, can be used to generate seemingly continuous and infinitely large scenes by exploiting more effective and more efficient representation techniques.

c. Towards synthesis and representation of context-relevant scene properties

Machine vision models mainly rely on visual appearance of input data to make predictions. The realism of visual features is, thus, the sole concern of developers and researchers when designing methods for synthesizing training data. However, in more complex tasks such as affordance learning, robot perception and dexterous manipulation, in addition to synthesizing appearance and geometry, it is often helpful to model non-visual properties such as friction, mass and other semantic information about objects in the scene. We expect that approaches that rely on jointly modeling visual information and high-level non-visual attributes that reflect properties of the real world – or physics-aware data synthesis – will become an important research topic in the foreseeable future. Research in this direction will facilitate new ways to represent visual and semantic context information in more unified and coherent fashion. This will allow the synthesis fully interactive 3D environments without explicitly modeling object properties and behavior to become possible.

d. Simulating less intuitive augmentation schemes

Current data synthesis approaches assume visual similarity of the original data or target domain and the generated image samples. Consequently, the generative modeling process primarily aims to generate clean data that is as close as possible to the target data. It is, however, known that techniques such as random image perturbations can sometimes provide the most useful augmentations for improving generalization. Thus, by focusing on more aligned semantic content, data synthesis approaches based on generative modeling usually ignore useful augmentation strategies that rely on non-realistic data (e.g., methods such as blurring and noise injection). At present, there is no workaround that allows to generate implausible but effective data using generative modeling.

12 Conclusion

Synthetic data augmentation is a way to overcome data scarcity in practical machine learning applications by creating artificial samples from scratch. This survey explores the most important approaches to generating synthetic data for training computer vision models. We present a detailed coverage of the methods, unique properties, application scenarios, as well as the important limitations of data synthesis methods for extending training data. We also summarized the main features, generation methods, supported tasks and application domains of common publicly available, large-scale synthetic datasets. Lastly, we investigate the effectiveness of data synthesis approaches to data augmentation. The survey shows that synthetic data augmentation methods provides an effective means to obtain good generalization performance in situations where it is difficult to access real data for training. Moreover, for tasks such as optical flow, depth estimation and visual odometry, where photorealism plays no role in inference, training with synthetic data sometimes yield better performance than with real data.

Acknowledgments

The authors would like to thank...

References

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[2] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[3] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 109–117.

[4] G. Rogez and C. Schmid, "Mocap-guided data augmentation for 3d pose estimation in the wild," *Advances in neural information processing systems*, vol. 29, 2016.

[5] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2o-afford: Annotation-free large-scale object-object affordance learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 1666–1677.

[6] F.-J. Chu, R. Xu, and P. A. Vela, "Learning affordance segmentation for real-world robotic manipulation via synthetic images," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1140–1147, 2019.

[7] Y. Lin, C. Tang, F.-J. Chu, and P. A. Vela, "Using synthetic data and deep networks to recognize primitive shapes for object grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 494–10 501.

[8] A. Ummadisingu, K. Takahashi, and N. Fukaya, "Cluttered food grasping with adaptive fingers and synthetic-data trained object detection," *arXiv preprint arXiv:2203.05187*, 2022.

[9] T. Kollar, M. Laskey, K. Stone, B. Thananjeyan, and M. Tjersland, "Simnet: Enabling robust unknown object manipulation from pure synthetic data via stereo," in *Conference on Robot Learning*. PMLR, 2022, pp. 938–948.

[10] Z. Luo, W. Xue, J. Chae, and G. Fu, "Skp: Semantic 3d keypoint detection for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5437–5444, 2022.

[11] A. H. Ornek and M. Ceylan, "Comparison of traditional transformations for data augmentation in deep learning of medical thermography," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2019, pp.191–194.

[12] K. Wang, B. Fang, J. Qian, S. Yang, X. Zhou, and J. Zhou, "Perspective transformation data augmentation for object detection," *IEEE Access*, vol. 8, pp. 4935–4943, 2019.

[13] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[14] E. K. Kim, H. Lee, J. Y. Kim, and S. Kim, "Data augmentation method by applying color perturbation of inverse psnr and geometric transformations for object recognition based on deep learning," *Applied Sciences*, vol. 10, no. 11, p. 3755, 2020.

[15] D. Sakkos, H. P. Shum, and E. S. Ho, "Illumination-based data augmentation for robust background subtraction," in *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, 2019, pp. 1–8.

[16] O. Mazhar and J. Kober, "Random shadows and highlights: A new data augmentation method for extreme lighting conditions," *arXiv preprint arXiv:2101.05361*, 2021.

[17] A. Kotwal, R. Bhalodia, and S. P. Awate, "Joint desmoking and denoising of laparoscopy images," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2016, pp. 1050–1054.

[18] H. Li, X. Zhang, Q. Tian, and H. Xiong, "Attribute mix: semantic data augmentation for fine grained recognition," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2020, pp. 243–246.

[19] S. Feng, S. Yang, Z. Niu, J. Xie, M. Wei, and P. Li, "Grid cut and mix: flexible and efficient data augmentation," in *Twelfth International Conference on Graphics and Image Processing (ICGIP 2020)*, vol. 11720. International Society for Optics and Photonics, 2021, p. 1172028.

[20] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[21] J. Yoo, N. Ahn, and K.-A. Sohn, "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8375–8384.

[22] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *Ieee Access*, vol. 5, pp. 5858–5869, 2017.

[23] X. Li, Y. Dai, Y. Ge, J. Liu, Y. Shan, and L.-Y. Duan, "Uncertainty modeling for out-of-distribution generalization," *arXiv preprint arXiv:2202.03958*, 2022.

[24] X. Bouthillier, K. Konda, P. Vincent, and R. Memisevic, "Dropout as data augmentation," *arXiv preprint arXiv:1506.08700*, 2015.

[25] B.-B. Jia and M.-L. Zhang, "Multi-dimensional classification via selective feature augmentation," *Machine Intelligence Research*, vol. 19, no. 1, pp. 38–51, 2022.

[26] K. Maharana, S. Mondal, and B. Nemade, "A review: Data preprocessing and data augmentation techniques," *Global Transitions Proceedings*, 2022.

[27] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[28] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv preprint arXiv:2204.08610*, 2022.

[29] C. Khosla and B. S. Saini, "Enhancing performance of deep learning models with different data augmentation techniques: A survey," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*. IEEE, 2020, pp. 79–85.

[30] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, pp. 1–27, 2021.

[31] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, vol. 448. Citeseer, 1983, pp. 448–453.

[32] P. R. Jeyaraj and E. R. S. Nadar, "Deep boltzmann machine algorithm for accurate medical image analysis for classification of cancerous region," *Cognitive Computation and Systems*, vol. 1, no. 3, pp. 85–90, 2019.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] H. Akaike, "autoregressive models for regression," *Annals of the Institute of Statistical Mathematics*, vol. 21, pp. 243–247, 1969.

[36] J. M. Susskind, G. E. Hinton, J. R. Movellan, and A. K. Anderson, "Generating facial expressions with deep belief nets," *Affective Computing, Emotion Modelling, Synthesis and Recognition*, pp. 421–440, 2008.

[37] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," *Advances in neural information processing systems*, vol. 30, 2017.

[38] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2391–2400.

[39] J. Peng, D. Liu, S. Xu, and H. Li, "Generating diverse structure for image inpainting with hierarchical vq-vae," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 775–10 784.

[40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[41] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," *Advances in neural information processing systems*, vol. 28, 2015.

[42] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in neural information processing systems*, vol. 29, 2016.

[43] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[44] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 1857–1865.

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[46] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[47] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.

[48] Y. Xue, Y. Li, K. K. Singh, and Y. J. Lee, "Giraffe hd: A high-resolution 3d-aware generative model," *arXiv preprint arXiv:2203.14954*, 2022.

[49] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5799–5809.

[50] X. Zhang, Z. Zheng, D. Gao, B. Zhang, P. Pan, and Y. Yang, "Multi-view consistent generative adversarial networks for 3d-aware image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 450–18 459.

[51] H. Ohno, "Auto-encoder-based generative models for data augmentation on regression problems," *Soft Computing*, vol. 24, no. 11, pp. 7999–8009, 2020.

[52] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[53] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[54] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[55] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *arXiv preprint arXiv:1802.07228*, 2018.

[56] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[57] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[58] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 289–293.

[59] D. Ribli, A. Horváth, Z. Unger, P. Pollner, and I. Csabai, "Detecting and classifying lesions in mammograms with deep learning," *Scientific reports*, vol. 8, no. 1, pp. 1–7, 2018.

[60] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2606–2615.

[61] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2226–2234.

[62] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8990–8999.

[63] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[64] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Gan augmentation: Augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, 2018.

[65] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Chest x-ray generation and data augmentation for cardiovascular abnormality classification," in *Medical Imaging 2018: Image Processing*, vol. 10574. International Society for Optics and Photonics, 2018, p. 105741M.

[66] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5688–5696.

[67] S. Kaur, H. Aggarwal, and R. Rani, "Mr image synthesis using generative adversarial networks for parkinson's disease classification," in *Proceedings of International Conference on Artificial Intelligence and Applications*. Spring, 2021, pp. 317–327.

[68] S. Kaplan, L. Lensu, L. Laaksonen, and H. Uusitalo, "Evaluation of unconditioned deep generative synthesis of retinal images," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2020, pp. 262–273.

[69] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "A method for improving cnn-based image recognition using dcgan," *Computers, Materials and Continua*, vol. 57, no. 1, pp. 167–178, 2018.

[70] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "St-gan: Spatial transformer generative adversarial networks for image compositing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9455–9464.

[71] S. C. Medin, B. Egger, A. Cherian, Y. Wang, J. B. Tenenbaum, X. Liu, and T. K. Marks, "Most-gan: 3d morphable stylegan for disentangled face image manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 1962–1971.

[72] Z. Chen, Z. Zeng, H. Shen, X. Zheng, P. Dai, and P. Ouyang, "Dn-gan: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images," *Biomedical Signal Processing and Control*, vol. 55, p. 101632, 2020.

[73] D.-P. Fan, Z. Huang, P. Zheng, H. Liu, X. Qin, and L. Van Gool, "Facial-sketch synthesis: a new challenge," *Machine Intelligence Research*, vol. 19, no. 4, pp. 257–287, 2022.

[74] P. L. Suárez, A. D. Sappa, and B. X. Vintimilla, "Infrared image colorization based on a triplet dcgan architecture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 18–23.

[75] X. Zhao, F. Ma, D. Güera, Z. Ren, A. G. Schwing, and A. Colburn, "Generative multiplane images: Making a 2d gan 3d-aware," in *European Conference on Computer Vision*. Springer, 2022, pp. 18–35.

[76] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2439–2448.

[77] Y.-J. Ju, G.-H. Lee, J.-H. Hong, and S.-W. Lee, "Complete face recovery gan: Unsupervised joint face rotation and de-occlusion from a single-view image," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 3711–3721.

[78] X. Chen, X. Luo, J. Weng, W. Luo, H. Li, and Q. Tian, "Multi-view gait image generation for cross-view gait recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 3041–3055, 2021.

[79] S. Kim, J. Lee, and B. C. Ko, "Ssl-mot: self-supervised learning based multi-object tracking," *Applied Intelligence*, pp. 1–11, 2022.

[80] X. Wang, C. Li, B. Luo, and J. Tang, "Sint++: Robust visual tracking via adversarial positive instance generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4864–4873.

[81] Q. Wu, Z. Chen, L. Cheng, Y. Yan, B. Li, and H. Wang, "Hallucinated adversarial learning for robust visual tracking," *arXiv preprint arXiv:1906.07008*, 2019.

[82] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4099–4108.

[83] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3754–3762.

[84] M. Zanfir, A.-I. Popa, A. Zanfir, and C. Sminchisescu, "Human appearance transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5391–5399.

[85] K. Saleh, S. Szénási, and Z. Vámossy, "Occlusion handling in generic object detection: A review," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2021, pp. 000 477–000 484.

[86] L. Minciullo, F. Manhardt, K. Yoshikawa, S. Meier, F. Tombari, and N. Kobori, "Db-gan: Boosting object recognition under strong lighting conditions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2939–2949.

[87] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[88] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[89] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[90] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 11, pp. 3943–3956, 2019.

[91] S. K. Jemni, M. A. Souibgui, Y. Kessentini, and A. Fornés, "Enhance to read better: A multi-task adversarial network for handwritten document image enhancement," *Pattern Recognition*, vol. 123, p. 108370, 2022.

[92] J. Liang, M. Li, Y. Jia, and R. Sun, "Single image dehazing in 3d space with more stable gans," in *Proceedings of 2021 Chinese Intelligent Systems Conference*. Springer, 2022, pp. 581–590.

[93] X. Li, G. Teng, P. An, H. Yao, and Y. Chen, "Advertisement logo compositing via adversarial geometric consistency pursuit," in *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2019, pp. 1–4.

[94] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, "Gagan: Geometry-aware generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 878–887.

[95] F. Zhan, C. Xue, and S. Lu, "Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9105–9115.

[96] S. Treneska, E. Zdravevski, I. M. Pires, P. Lameski, and S. Gievska, "Gan-based image colorization for self-supervised visual feature learning," *Sensors*, vol. 22, no. 4, p. 1599, 2022.

[97] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[98] W. Wang, H. Wang, S. Yang, X. Zhang, X. Wang, J. Wang, J. Lei, Z. Zhang, and Z. Dong, "Resolution enhancement in microscopic imaging based on generative adversarial network with unpaired data," *Optics Communications*, vol. 503, p. 127454, 2022.

[99] S. N. Rai and C. Jawahar, "Removing atmospheric turbulence via deep adversarial learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 2633–2646, 2022.

[100] S. Tripathi, Z. C. Lipton, and T. Q. Nguyen, "Correction by projection: Denoising images with generative adversarial networks," *arXiv preprint arXiv:1803.04477*, 2018.

[101] Q. Lyu, C. You, H. Shan, Y. Zhang, and G. Wang, "Super-resolution mri and ct through gan-circle," in *Developments in X-ray tomography XII*, vol. 11113. International Society for Optics and Photonics, 2019, p. 111130X.

[102] F. Chiaroni, M.-C. Rahal, N. Hueber, and F. Dufaux, "Hallucinating a cleanly labeled augmented dataset from a noisy labeled dataset using gan," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 3616–3620.

[103] W. Lira, J. Merz, D. Ritchie, D. Cohen-Or, and H. Zhang, "Ganhopper: Multi-hop gan for unsupervised image-to-image translation," in *European conference on computer vision*. Springer, 2020, pp. 363–379.

[104] E. Ntavelis, M. Shahbazi, I. Kastanis, R. Timofte, M. Danelljan, and L. Van Gool, "Arbitrary-scale image synthesis," *arXiv preprint arXiv:2204.02273*, 2022.

[105] L. Sixt, B. Wild, and T. Landgraf, "Rendergan: Generating realistic labeled data," *Frontiers in Robotics and AI*, vol. 5, p. 66, 2018.

[106] J. Zhao, L. Xiong, P. Karlekar Jayashree, J. Li, F. Zhao, Z. Wang, P. Sugiri Pranata, P. Shengmei Shen, S. Yan, and J. Feng, "Dual-agent gans for photorealistic and identity preserving profile face synthesis," *Advances in neural information processing systems*, vol. 30, 2017.

[107] A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *Advances in neural information processing systems*, vol. 30, 2017.

[108] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *Advances in neural information processing systems*, vol. 33, pp. 7559–7570, 2020.

[109] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 104–12 114, 2020.

[110] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, "Towards good practices for data augmentation in gan training," *arXiv preprint arXiv:2006.05338*, vol. 2, no. 3, p. 3, 2020.

[111] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," *arXiv preprint arXiv:1910.12027*, 2019.

[112] Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena, and H. Zhang, "Improved consistency regularization for gans," in *Proceedings of*

the AAAI conference on artificial intelligence, vol. 35, no. 12, 2021, pp. 11 033–11 041.

[113] S. Park, Y.-J. Yeo, and Y.-G. Shin, "Generative adversarial network using perturbed-convolutions," *arXiv preprint arXiv:2101.10841*, vol. 1, no. 3, p. 8, 2021.

[114] B. Dodin and M. Sirvanci, "Stochastic networks and the extreme value distribution," *Computers & operations research*, vol. 17, no. 4, pp. 397–409, 1990.

[115] S. Bhatia, A. Jain, and B. Hooi, "Exgan: Adversarial generation of extreme samples," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6750–6758.

[116] L. Liu, M. Muelly, J. Deng, T. Pfister, and L.-J. Li, "Generative modeling for small-data object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6073–6081.

[117] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[118] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[119] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[120] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[121] C. Nash, J. Menick, S. Dieleman, and P. W. Battaglia, "Generating images with sparse representations," *arXiv preprint arXiv:2103.03841*, 2021.

[122] M. J. Chong and D. Forsyth, "Effectively unbiased fid and inception score and where to find them," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6070–6079.

[123] C.-Y. Bai, H.-T. Lin, C. Raffel, and W. C.-w. Kan, "On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2534–2542.

[124] S. Liu, Y. Wei, J. Lu, and J. Zhou, "An improved evaluation framework for generative adversarial networks," *arXiv preprint arXiv:1803.07474*, 2018.

[125] S. Zhou, M. Gordon, R. Krishna, A. Narcomey, L. F. Fei-Fei, and M. Bernstein, "Hype: A benchmark for human eye perceptual evaluation of generative models," *Advances in neural information processing systems*, vol. 32, 2019.

[126] P. Salehi, A. Chalechale, and M. Taghizadeh, "Generative adversarial networks (gans): An overview of theoretical model, evaluation metrics, and recent developments," *arXiv preprint arXiv:2005.13178*, 2020.

[127] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *2020 international joint conference on neural networks (ijcnn)*. IEEE, 2020, pp. 1–10.

[128] L. Xu, X. Zeng, Z. Huang, W. Li, and H. Zhang, "Low-dose chest x-ray image super-resolution using generative adversarial nets with spectral normalization," *Biomedical Signal Processing and Control*, vol. 55, p. 101600, 2020.

[129] M. Lee and J. Seok, "Regularization methods for generative adversarial networks: An overview of recent studies," *arXiv preprint arXiv:2005.09165*, 2020.

[130] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "Mgan: Training generative adversarial nets with multiple generators," in *International conference on learning representations*, 2018.

[131] M. M. Saad, R. O' Reilly, and M. H. Rehmani, "A survey on training challenges in generative adversarial networks for biomedical image analysis," *Artificial Intelligence Review*, vol. 57, no. 2, p. 19, 2024.

[132] Z. Zhou, Q. Zhang, G. Lu, H. Wang, W. Zhang, and Y. Yu, "Adashift: Decorrelation and convergence of adaptive learning rate methods," *arXiv preprint arXiv:1810.00143*, 2018.

[133] Y. Gan, T. Xiang, H. Liu, M. Ye, and M. Zhou, "Generative adversarial networks with adaptive learning strategy for noise-to-image synthesis," *Neural Computing and Applications*, vol. 35, no. 8, pp. 6197–6206, 2023.

[134] K. Li and D.-K. Kang, "Enhanced generative adversarial networks with restart learning rate in discriminator," *Applied Sciences*, vol. 12, no. 3, p. 1191, 2022.

[135] C. G. Korde, M. Vasantha *et al.*, "Training of generative adversarial networks with hybrid evolutionary optimization technique," in *2019 IEEE 16th India Council International Conference (INDICON)*. IEEE, 2019, pp. 1–4.

[136] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2686–2694.

[137] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1278–1286.

[138] S. Liu and S. Ostadabbas, "A semi-supervised data augmentation approach using 3d graphical engines," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[139] R. Sulzer, L. Landrieu, A. Boulch, R. Marlet, and B. Vallet, "Deep surface reconstruction from point clouds with visibility information," *arXiv preprint arXiv:2202.01810*, 2022.

[140] J. Malik, S. Shimada, A. Elhayek, S. A. Ali, V. Golyanik, C. Theobalt, and D. Stricker, "Handvoxnet++: 3d hand shape and pose estimation using voxel-based neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[141] F. Bongini, L. Berlincioni, M. Bertini, and A. Del Bimbo, "Partially fake it till you make it: mixing real and fake thermal images for improved object detection," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5482–5490.

[142] V. Hegde and R. Zadeh, "Fusionnet: 3d object classification using multiple data representations," *arXiv preprint arXiv:1607.05695*, 2016.

[143] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 912–10 922.

[144] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora *et al.*, "Abo: Dataset and benchmarks for real-world 3d object understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 21 126–21 136.

[145] H. Hattori, V. Naresh Boddeti, K. M. Kitani, and T. Kanade, "Learning scene-specific pedestrian detectors without real data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3819–3827.

[146] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[147] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[148] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis, "Looking beyond appearances: Synthetic training data for deep cnns in re-identification," *Computer Vision and Image Understanding*, vol. 167, pp. 50–62, 2018.

[149] K. Ashish and S. Shital, "Microsoft extends airsim to include autonomous car research."

[150] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[151] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 961–972, 2018.

[152] N. Jaipuria, X. Zhang, R. Bhasin, M. Arafa, P. Chakravarty, S. Shrivastava, S. Manglani, and V. N. Murali, "Deflating dataset bias using synthetic data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 772–773.

[153] S. Borkman, A. Crespi, S. Dhakad, S. Ganguly, J. Hogins, Y.-C. Jhang, M. Kamalzadeh, B. Li, S. Leal, P. Parisi *et al.*, "Unity perception: Generate synthetic data for computer vision," *arXiv preprint arXiv:2107.04259*, 2021.

[154] J. Jang, H. Lee, and J.-C. Kim, "Carfree: Hassle-free object detection dataset generation using carla autonomous driving simulator," *Applied Sciences*, vol. 12, no. 1, p. 281, 2021.

[155] K. M. Hart, A. B. Goodman, and R. P. O'Shea, "Automatic generation of machine learning synthetic data using ros," in *International Conference on Human-Computer Interaction*. Springer, 2021, pp. 310–325.

[156] M. S. Mueller and B. Jutzi, "Uas navigation with squeezeposenet accuracy boosting for pose regression by data augmentation," *Drones*, vol. 2, no. 1, p. 7, 2018.

[157] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[158] A. Kerim, L. Soriano Marcolino, and R. Jiang, "Silver: Novel rendering engine for data hungry computer vision models," in *2nd International Workshop on Data Quality Assessment for Machine Learning*, 2021.

[159] A. Shafaei, J. J. Little, and M. Schmidt, "Play and learn: Using video games to train computer vision models," *arXiv preprint arXiv:1608.01745*, 2016.

[160] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*. Springer, 2016, pp. 102–118.

[161] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European conference on computer vision*. Springer, 2012, pp. 611–625.

[162] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4340–4349.

[163] C. Roberto de Souza, A. Gaidon, Y. Cabon, and A. Manuel Lopez, "Procedural generation of videos to train deep action recognition networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4757–4767.

[164] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[165] M. Wrenninge and J. Unger, "Synscapes: A photorealistic synthetic dataset for street scene parsing," *arXiv preprint arXiv:1810.08705*, 2018.

[166] E. Cheung, T. K. Wong, A. Bera, X. Wang, and D. Manocha, "Lcrowdv: Generating labeled videos for simulation-based crowd behavior learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 709–727.

[167] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi *et al.*, "Openrooms: An open framework for photorealistic indoor scene datasets," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7190–7199.

[168] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1459–1473, 2009.

[169] K. Vyas, L. Jiang, S. Liu, and S. Ostadabbas, "An efficient 3d synthetic model generation pipeline for human pose data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1542–1552.

[170] F. Bogo, M. J. Black, M. Loper, and J. Romero, "Detailed full-body reconstructions of moving people from monocular rgb-d sequences," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2300–2308.

[171] N. Hesse, S. Pujades, M. J. Black, M. Arens, U. G. Hofmann, and A. S. Schroeder, "Learning and tracking the 3d body shape of freely moving infants from rgb-d sequences," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2540–2551, 2019.

[172] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[173] G. Chogovadze, R. Pautrat, and M. Pollefeys, "Controllable data augmentation through deep relighting," *arXiv preprint arXiv:2110.13996*, 2021.

[174] C. Sevastopoulos, S. Konstantopoulos, K. Balaji, M. Zaki Zadeh, and F. Makedon, "A simulated environment for robot vision experiments," *Technologies*, vol. 10, no. 1, p. 7, 2022.

[175] S. Moro and T. Komuro, "Generation of virtual reality environment based on 3d scanned indoor physical space," in *International Symposium on Visual Computing*. Springer, 2021, pp. 492–503.

[176] M. Sra, S. Garrido-Jurado, and P. Maes, "Oasis: Procedurally generated social virtual spaces from 3d scanned real spaces," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 12, pp. 3174–3187, 2017.

[177] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets deep learning for car instance segmentation in urban scenes," in *British machine vision conference*, vol. 1, 2017, p. 2.

[178] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[179] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[180] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu, "Objectfolder 2.0: A multisensory object dataset for sim2real transfer," *arXiv preprint arXiv:2204.02389*, 2022.

[181] A. Barisic, F. Petric, and S. Bogdan, "Sim2air-synthetic aerial dataset for uav monitoring," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3757–3764, 2022.

[182] K. Dimitropoulos, I. Hatzilygeroudis, and K. Chatzilygeroudis, "A brief survey of sim2real methods for robot learning," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2022, pp. 133–140.

[183] T. Ikeda, S. Tanishige, A. Amma, M. Sudano, H. Audren, and K. Nishiwaki, "Sim2real instance-level style transfer for 6d pose estimation," *arXiv preprint arXiv:2203.02069*, 2022.

[184] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2107–2116.

[185] D.-Y. She and K. Xu, "Contrastive self-supervised representation learning using synthetic data," *International Journal of Automation and Computing*, vol. 18, no. 4, pp. 556–567, 2021.

[186] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2800–2810.

[187] S. Huang and D. Ramanan, "Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2243–2252.

[188] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer, 2016, pp. 868–884.

[189] Z. Chen, W. Ouyang, T. Liu, and D. Tao, "A shape transformation-based dataset augmentation framework for pedestrian detection," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1121–1138, 2021.

[190] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," *Advances in neural information processing systems*, vol. 30, 2017.

[191] Y. Pang, J. Cao, J. Wang, and J. Han, "Jcs-net: Joint classification and super-resolution network for small-scale pedestrian detection in surveillance images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3322–3331, 2019.

[192] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[193] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3386–3394.

[194] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional

networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4040–4048.

[195] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "Segstereo: Exploiting semantic information for disparity estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 636–651.

[196] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[197] "From traditional rendering to differentiable rendering: Theories, methods and applications. scientia sinica informationis, vol.51, no.7, pp.1043-1067, 2021." 2017.

[198] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907–3916.

[199] M. de La Gorce, N. Paragios, and D. J. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[200] J. Liu, C.-H. Wu, Y. Wang, Q. Xu, Y. Zhou, H. Huang, C. Wang, S. Cai, Y. Ding, H. Fan *et al.*, "Learning raw image denoising with bayer pattern unification and bayer preserving augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[201] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, and R. Ng, "Representing scenes as neural radiance fields for view synthesis," in *Proc. of European Conference on Computer Vision, Virtual*, 2020.

[202] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," *arXiv preprint arXiv:2202.05263*, 2022.

[203] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.

[204] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[205] E. Insafutdinov and A. Dosovitskiy, "Unsupervised learning of shape and pose with differentiable point clouds," *Advances in neural information processing systems*, vol. 31, 2018.

[206] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1067–1076.

[207] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[208] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, "Neural point-based graphics," in *European Conference on Computer Vision*. Springer, 2020, pp. 696–712.

[209] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[210] Z. Kuang, K. Olszewski, M. Chai, Z. Huang, P. Achlioptas, and S. Tulyakov, "Neroic: Neural rendering of objects from online image collections," *arXiv preprint arXiv:2201.02533*, 2022.

[211] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[212] S. Duggal, Z. Wang, W.-C. Ma, S. Manivasagam, J. Liang, S. Wang, and R. Urtasun, "Mending neural implicit modeling for 3d vehicle reconstruction in the wild," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1900–1909.

[213] A. R. Kosiorek, H. Strathmann, D. Zoran, P. Moreno, R. Schneider, S. Mokrá, and D. J. Rezende, "Nerf-vae: A geometry aware 3d scene generative model," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5742–5752.

[214] S. Yao, R. Zhong, Y. Yan, G. Zhai, and X. Yang, "Dfa-nerf: Personalized talking head generation via disentangled face attributes neural rendering," *arXiv preprint arXiv:2201.00791*, 2022.

[215] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 453–11 464.

[216] Y. Liu, Y.-S. Wei, H. Yan, G.-B. Li, and L. Lin, "Causal reasoning meets visual representation learning: A prospective study," *Machine Intelligence Research*, vol. 19, no. 6, pp. 485–511, 2022.

[217] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "Gancraft: Unsupervised 3d neural rendering of minecraft worlds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 072–14 082.

[218] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.

[219] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 172–189.

[220] A. Mallya, T.-C. Wang, K. Sapra, and M.-Y. Liu, "World-consistent video-to-video synthesis," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer, 2020, pp. 359–378.

[221] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[222] Z. Zhang, S. Xie, M. Chen, and H. Zhu, "Handaugment: A simple data augmentation method for depth-based 3d hand pose estimation," *arXiv preprint arXiv:2001.00702*, 2020.

[223] G. Ning, G. Chen, C. Tan, S. Luo, L. Bo, and H. Huang, "Data augmentation for object detection via differentiable neural rendering," *arXiv preprint arXiv:2103.02852*, 2021.

[224] Q. Wu, Y. Li, Y. Sun, Y. Zhou, H. Wei, J. Yu, and Y. Zhang, "An arbitrary scale super-resolution approach for 3-dimensional magnetic resonance image using implicit neural representation," *arXiv preprint arXiv:2110.14476*, 2021.

[225] Q. Wu, Y. Li, L. Xu, R. Feng, H. Wei, Q. Yang, B. Yu, X. Liu, J. Yu, and Y. Zhang, "Irem: High-resolution magnetic resonance image reconstruction via implicit neural representation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 65–74.

[226] L. Shen, J. Pauly, and L. Xing, "Nerp: Implicit neural representation learning with prior embedding for sparsely sampled image reconstruction," *arXiv preprint arXiv:2108.10991*, 2021.

[227] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng, "Learned initializations for optimizing coordinate-based neural representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2846–2855.

[228] D. B. Lindell, J. N. Martel, and G. Wetzstein, "Autoint: Automatic integration for fast neural volume rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 556–14 565.

[229] K. Gupta, B. Colvert, and F. Contijoch, "Neural computed tomography," *arXiv preprint arXiv:2201.06574*, 2022.

[230] Y. Sun, J. Liu, M. Xie, B. Wohlberg, and U. S. Kamilov, "Coil: Coordinate-based internal learning for imaging inverse problems," *arXiv preprint arXiv:2102.05181*, 2021.

[231] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.

[232] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, "Ibrnet: Learning multi-view image-based rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.

[233] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "Synsin: End-to-end view synthesis from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7467–7477.

[234] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner *et al.*, "State of the art on neural rendering," in *Computer Graphics Forum*, vol. 39, no. 2. Wiley Online Library, 2020, pp. 701–727.

[235] C. Liu, X.-F. Chen, C.-J. Bo, and D. Wang, "Long-term visual tracking: review and experimental comparison," *Machine Intelligence Research*, vol. 19, no. 6, pp. 512–530, 2022.

[236] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 024–16 033.

[237] J. Choe, B. Joung, F. Rameau, J. Park, and I. S. Kweon, "Deep point cloud reconstruction," *arXiv preprint arXiv:2111.11704*, 2021.

[238] P. Erler, P. Guerrero, S. Ohrhallinger, N. J. Mitra, and M. Wimmer, "Points2surf learning implicit surfaces from point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 108–124.

[239] T. Hashimoto and M. Saito, "Normal estimation for accurate 3d mesh reconstruction with point cloud model incorporating spatial structure." in *CVPR workshops*, vol. 1, 2019.

[240] A. Reed, T. Blanford, D. C. Brown, and S. Jayasuriya, "Implicit neural representations for deconvolving sas images," in *OCEANS 2021: San Diego–Porto*. IEEE, 2021, pp. 1–7.

[241] ——, "Sinr: Deconvolving circular sas images using implicit neural representations," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[242] F. Vasconcelos, B. He, N. Singh, and Y. W. Teh, "Uncertainr: Uncertainty quantification of end-to-end implicit neural representations for computed tomography," *arXiv preprint arXiv:2202.10847*, 2022.

[243] L. Shen, J. Pauly, and L. Xing, "Nerp: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[244] R. Liu, Y. Sun, J. Zhu, L. Tian, and U. S. Kamilov, "Recovery of continuous 3d refractive index maps from discrete intensity-only measurements using neural fields," *Nature Machine Intelligence*, vol. 4, no. 9, pp. 781–791, 2022.

[245] C. Gan, Y. Gu, S. Zhou, J. Schwartz, S. Alter, J. Traer, D. Gutfreund, J. B. Tenenbaum, J. H. McDermott, and A. Torralba, "Finding fallen objects via asynchronous audio-visual integration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 523–10 533.

[246] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu, "Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations," *arXiv preprint arXiv:2109.07991*, 2021.

[247] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[248] T. Chen, P. Wang, Z. Fan, and Z. Wang, "Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 191–15 202.

[249] J. Zhang, Y. Zhang, H. Fu, X. Zhou, B. Cai, J. Huang, R. Jia, B. Zhao, and X. Tang, "Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 376–18 386.

[250] S. Kulkarni, P. Yin, and S. Scherer, "360fusionnerf: Panoramic neural radiance fields with joint guidance," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7202–7209.

[251] Y. Jiang, S. Jiang, G. Sun, Z. Su, K. Guo, M. Wu, J. Yu, and L. Xu, "Neuralhofusion: Neural volumetric rendering under human-object interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6155–6165.

[252] A. Mumuni and F. Mumuni, "Cnn architectures for geometric transformation-invariant feature representation in computer vision: a review," *SN Computer Science*, vol. 2, no. 5, pp. 1–23, 2021.

[253] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[254] ——, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[255] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[256] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, "Real-time neural style transfer for videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 783–791.

[257] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos and spherical images," *International Journal of Computer Vision*, vol. 126, no. 11, pp. 1199–1219, 2018.

[258] ——, "Artistic style transfer for videos," in *German conference on pattern recognition*. Springer, 2016, pp. 26–36.

[259] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6654–6663.

[260] C. Do, "3d image augmentation using neural style transfer and generative adversarial networks," in *Applications of Digital Image Processing XLIII*, vol. 11510. SPIE, 2020, pp. 707–718.

[261] X. Zheng, T. Chalasani, K. Ghosal, S. Lutz, and A. Smolic, "Stada: Style transfer as data augmentation," *arXiv preprint arXiv:1909.01056*, 2019.

[262] I. Darma, N. Suciati, and D. Siahaan, "Neural style transfer and geometric transformations for data augmentation on balinese carving recognition using mobilenet," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 349–363, 2020.

[263] B. Georgievski, "Image augmentation with neural style transfer," in *International Conference on ICT Innovations*. Springer, 2019, pp. 212–224.

[264] P. A. Cicalese, A. Mobiny, P. Yuan, J. Becker, C. Mohan, and H. V. Nguyen, "Stypath: Style-transfer data augmentation for robust histology image classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 351–361.

[265] Y. Xu and A. Goel, "Cross-domain image classification through neural-style transfer data augmentation," *arXiv preprint arXiv:1910.05611*, 2019.

[266] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[267] S. Cygert and A. Czyżewski, "Toward robust pedestrian detection with data augmentation," *IEEE Access*, vol. 8, pp. 136 674–136 683, 2020.

[268] A. Mikołajczyk and M. Grochowski, "Style transfer-based image synthesis as an efficient regularization technique in deep learning," in *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2019, pp. 42–47.

[269] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: data augmentation via style randomization." in *CVPR Workshops*, vol. 6, 2019, pp. 10–11.

[270] Y. Yi, "Microsoft extends airsim to include autonomous car research," 2020.

[271] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[272] S. S. Kim, N. Kolkin, J. Salavon, and G. Shakhnarovich, "Deformable style transfer," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 246–261.

[273] X.-C. Liu, Y.-L. Yang, and P. Hall, "Learning to warp for style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3702–3711.

[274] S. Li, X. Xu, L. Nie, and T.-S. Chua, "Laplacian-steered neural style transfer," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1716–1724.

[275] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4990–4998.

[276] R. R. Yang, "Multi-stage optimization for photorealistic neural style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[277] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 453–468.

[278] B. Kim, V. C. Azevedo, M. Gross, and B. Solenthaler, "Transport-based neural style transfer for smoke simulations," *arXiv preprint arXiv:1905.07442*, 2019.

[279] ——, "Lagrangian neural style transfer for fluids," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 52–1, 2020.

[280] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1897–1906.

[281] Z. Wang, L. Zhao, H. Chen, L. Qiu, Q. Mo, S. Lin, W. Xing, and D. Lu, "Diversified arbitrary style transfer via deep feature perturbation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7789–7798.

[282] C. Castillo, S. De, X. Han, B. Singh, A. K. Yadav, and T. Goldstein, "Son of zorn's lemma: Targeted style transfer using instance-aware semantic segmentation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 1348–1352.

[283] Z. Chen, W. Wang, E. Xie, T. Lu, and P. Luo, "Towards ultra-resolution neural style transfer via thumbnail instance normalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 393–400.

[284] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9465–9474.

[285] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3920–3928.

[286] Z. Xu, T. Wang, F. Fang, Y. Sheng, and G. Zhang, "Stylization-based architecture for fast deep exemplar colorization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9363–9372.

[287] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6924–6932.

[288] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary style transfer with deep feature reshuffle," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8222–8231.

[289] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[290] E. Risser, P. Wilmot, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," *arXiv preprint arXiv:1701.08893*, 2017.

[291] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," *arXiv preprint arXiv:1701.01036*, 2017.

[292] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha, "Photorealistic style transfer via wavelet transforms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9036–9045.

[293] T. Yong-Jian and Z. Fan-Ju, "Neural style transfer algorithm based on laplacian operator and color preservation," *Journal of Computer Applications*, p. 0, 2022.

[294] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," *arXiv preprint arXiv:1808.03232*, 2018.

[295] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sỳkora, "Stylit: illumination-guided example-based stylization of 3d renderings," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.

[296] C. Rodriguez-Pardo and E. Garces, "Neural photometry-guided visual attribute transfer," *arXiv preprint arXiv:2112.02520*, 2021.

[297] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[298] E. Heitz, K. Vanhoey, T. Chambon, and L. Belcour, "A sliced wasserstein loss for neural texture synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9412–9420.

[299] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3985–3993.

[300] S. d'Angelo, F. Precioso, and F. Gandon, "Revisiting artistic style transfer for data augmentation in a real-case scenario," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 4178–4182.

[301] X.-C. Liu, X.-Y. Li, M.-M. Cheng, and P. Hall, "Geometric style transfer," *arXiv preprint arXiv:2007.05471*, 2020.

[302] Y. Jing, Y. Mao, Y. Yang, Y. Zhan, M. Song, X. Wang, and D. Tao, "Learning graph neural networks for image style transfer," in *European Conference on Computer Vision*. Springer, 2022, pp. 111–128.

[303] J. Tremblay, M. Meshry, A. Evans, J. Kautz, A. Keller, S. Khamis, C. Loop, N. Morrical, K. Nagano, T. Takikawa *et al.*, "Rtmv: A ray-traced multi-view synthetic dataset for novel view synthesis," *arXiv preprint arXiv:2205.07058*, 2022.

[304] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7822–7831.

[305] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3d. net: A new large-scale point cloud classification benchmark," *arXiv preprint arXiv:1704.03847*, 2017.

[306] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman, "Pix3d: Dataset and methods for single-image 3d shape modeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2974–2983.

[307] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Synthetic 2d lidar for precise vehicle localization in 3d urban environment," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1554–1559.

[308] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2678–2687.

[309] X. Wang, K. Wang, and S. Lian, "A survey on face data augmentation for the training of deep neural networks," *Neural computing and applications*, vol. 32, no. 19, pp. 15 503–15 531, 2020.

[310] P. S. Rajpura, H. Bojinov, and R. S. Hegde, "Object detection using deep cnns trained on synthetic images," *arXiv preprint arXiv:1706.06782*, 2017.

[311] Z. Zhang, L. Yang, and Y. Zheng, "Multimodal medical volumes translation and segmentation with generative adversarial network," *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp. 183–204, 2020.

[312] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 922–12 931.