Article

Background Augmentation Generative Adversarial Networks (BAGANs): Effective Data Generation Based on GAN-Augmented 3D Synthesizing [†]

Yan Ma ¹ , Kang Liu ^{1,*} , Zhibin Guan ¹, Xinkai Xu ^{1,2}, Xu Qian ¹ and Hong Bao ³

¹ School of Mechanical Electronic & Information Engineering, China University of Mining & Technology, Beijing 100083, China; yan.ma@student.cumtb.edu.cn (Y.M.); guanzhibin@yeah.net (Z.G.); ldxinkai@bnu.edu.cn (X.X.); xuqian@cumtb.edu.cn (X.Q.)

² Demonstration Center of Experimental Teaching in Comprehensive Engineering, Beijing Union University, Beijing 100101, China

³ Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China; baohong@bnu.edu.cn

* Correspondence: kangliu@cumtb.edu.cn; Tel.: +86-188-0131-1165

[†] This paper is an extended version of our published in BICS2018.

Received: 8 November 2018; Accepted: 6 December 2018; Published: 8 December 2018

Abstract: Augmented Reality (AR) is crucial for immersive Human–Computer Interaction (HCI) and the vision of Artificial Intelligence (AI). Labeled data drives object recognition in AR. However, manually annotating data is expensive, labor-intensive, and data distribution asymmetry. Scantly labeled data limits the application of AR. Aiming at solving the problem of insufficient and asymmetry training data in AR object recognition, an automated vision data synthesis method, i.e., background augmentation generative adversarial networks (BAGANs), is proposed in this paper based on 3D modeling and the Generative Adversarial Network (GAN) algorithm. Our approach has been validated to have better performance than other methods through image recognition tasks with respect to the natural image database ObjectNet3D. This study can shorten the algorithm development time of AR and expand its application scope, which is of great significance for immersive interactive systems.

Keywords: object recognition; image data synthesizing; human–computer interaction; data synthesizing for immersive HCI; generative adversarial nets; background augmentation generative adversarial networks (BAGANs)

1. Introduction

Augmented Reality (AR) is an essential part of immersive Human–Computer Interaction (HCI). Using an advanced sensing system, AR provides an essential platform for human–computer interaction, such as Google Glass [1,2] and Microsoft HoloLens [3]. AR has great prospects in medical, industrial, and office education fields, among others. Therefore, the vision algorithm is an essential sub-topic of human–computer interaction research. On the other side, benefited by deep learning and big-data, data-driven computer vision algorithms based on deep learning have had many significant breakthroughs [4]. More and more computer vision algorithms based on deep learning have been built and have achieved state-of-the-art performance.

However, the existing data sets cannot fulfill the demands of AR and significantly limit the application of AR in development. Due to the broad applications of AR technology, the visual data of AR needs rich multi-class visually labeled data. More than that, people have increasing requirements for advanced visual tasks of AR with the development of human–computer interaction.

The authors in [1–3] presented strengthened methods of visual interaction because advanced visual intelligence can finish complex visual tasks. A high-performance visual intelligence (or computer vision) algorithm is vital for AR.

These factors lead to two formidable problems. On the one hand, the quantity of annotated data impacts the performance of an algorithm. On the other hand, annotating data is arduous work, because of the increasing advanced visual tasks required by the interaction design [5–8].

To solve the lack of annotation data, researchers have (1) used unsupervised vision learning algorithms to decrease the demands of annotated data and (2) obtained more annotation data for supervised learning using automated methods.

Unsupervised learning algorithms do not rely on an annotation to indicate the relationship between input and output. Figure 1 indicates the difference between supervised learning and unsupervised learning. Unsupervised learning can deal with some tasks more easily than supervised learning because it does not need the guidance of annotation in data. However, the research of unsupervised learning is in its infancy, and it cannot replace supervised learning algorithms.

Figure 1. Supervised learning and unsupervised learning. Supervised learning uses annotation guidance to draw learning-task-related conclusions about the data. Unsupervised learning uses the latent factors in data to conclude the relationship between data and the corresponding learning task, and there is no need to mark the data.

Visual data enhancement algorithms are designed to reduce the cost of manual marking, a problem in computer vision based on deep learning. At present, supervised learning algorithms are the primary engine driving the development of artificial intelligence. Due to its data-driven nature, supervised learning is or has been outperformed by traditional learning methods in some respects. Generally, using deep supervised learning algorithms should collect large data and be annotated by users.

Traditional visual data augmentation algorithms use spoofing methods to enhance the annotated data. Modern visual computing algorithms have a funny bug: when a simple transformation (such as 1 degree to the right) occurs, the converted image and the original image will be treated as two different images with the same annotation. Krizhevsky mentioned in [9] that the use of traditional visual data augmentation could significantly improve the classification accuracy of the model and enhance the generalization performance in the real world. As a result, AlexNet was competed in the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) in 2012. With the deepening of research, many different traditional vision algorithms, such as image rescaling [10], have been proposed and have been shown to improve the effectiveness of the model significantly.

However, traditional visual data augmentation methods have two congenital shortcomings. First, simply converting visual data could not bring diversity to the appearance distribution of the original data. Visual intelligence algorithms cannot benefit much from the converting data. Second, in some advanced visual tasks (based on image classification but not limited to image classification), some methods are unable to transform the annotation of data. For example, in object detection, rotation methods (a traditional method of visual data augmentation which rotates an image as a new image) cannot be used because the bounding box annotation cannot be transformed directly.

Generating visual data augmentation, making various image data by unsupervised learning, is an effective way to solve the shortcoming of labeled data. Although unsupervised learning is not mature enough to take the place of supervised learning, the ability to extract latent space (or random noise) is useful for increasing the diversity of the original data. In Krizhevsky's paper [9] (2012), they not only use the traditional visual data enhancement algorithm but also change hidden features, extracted by primary component analysis (PCA) [11], to enrich the appearance of the original data. In 2014, a generative adversarial network (GAN) [12] provided an alternative way of producing various visual data from the original data. Although a GAN is an algorithm that approximates the distribution of data, uncontrollable random processes cannot provide useful data for supervised learning. In 2016, Odena [13] proposed conditional generative adversarial networks (CGAN or conditional-GAN) to generate visual data with annotation by controlling the random noise generation. However, both GAN and CGAN has a simple architecture which cannot support the model to create a nature image. In 2016, a deep convolutional generative adversarial network (DCGAN) [14] synthesizes more natural and high-resolution images using complex random noise generation and unique lose function.

However, two difficult research gaps exist in generating visual data augmentation. First, the guaranteed quality of each image results in serious data mismatch problems in model training progress. Secondly, creating visual data augmentation cannot produce multiple annotations for visual data. At present, only image classification data annotations are possible.

According to the IEEE conference on computer vision and pattern recognition 2017 (CVPR2017), Shrivastava [15] did not use original GAN produce images from latent space. They used computer graphic methods rendered a coarse image. After that, a GAN to refine the rendered images was implemented. The GAN produced labeled training data, which decreased the demands of visual intelligence. Finally, they generated ideal training data. As an endorsement, this article [15] received the CVPR2017 Best Paper award.

In this article, using foreground images (rendering from 3D shapes) and random noises as input, we proposed a background augmentation GAN to decrease the complex task of GANs in data synthesis. Our work could generate guaranteed labeled vision data. To increase the edge appearance of the foreground object, we implemented a compositing layer which applied alpha compositing algorithms [16,17]. Figure 2 illustrates the schematic. In experiments, contrasting with the cycle-GAN [18] (73.64% accuracy) and ACGANs [13] (85.23% accuracy), our methods obtained the best performance (93.51% accuracy).

Figure 2. Background augmentation generative adversarial nets.

2. Related Work

2.1. Generative Adversarial Networks

Generative adversarial networks are the epoch-making unsupervised learning algorithm framework. In 2014, Goodfellow [12] introduced the novel unsupervised learning algorithm of generative adversarial nets. It made three breakthroughs for deep learning: (1) GAN only uses back propagation to train the network, avoiding the use of Markov chain; (2) The progressive adversarial process of GAN makes the design of value function flexible and simple; (3) it is always valid, even if the probability density of target domain is challenging to calculate. However, it also has two shortcomings: (1) the freedom from simple constraints increases the uncertainty of the final result and the difficulty of the training process; (2) the generator will improve, and the discriminator's return gradient will become increasingly small, making the network difficult to converge. Radford [14] introduced a GAN into computer vision through deconvolutional and convolutional networks. The deep convolutional generative adversarial nets (DCGANs) had an elaborate network architecture. It stabilized the training process, but it is not an optimum solution to the significant problems of GANs. Enumeration is always the wrong way to find a capable network structure. Arjovsky and Gulrajani [19–21] found the reason for these two problems in a mathematical way, solving problems by introducing the Wasserstein distance and the gradient penalty so that most of the networks could avoid the above two issues.

2.2. Conditional Generative Adversarial Networks

It is worth revisiting one of the critical reasons that cause GAN training and design difficulties: the freedom from simple constraints. It is indeed possible to solve some of the problems with GAN by adding constraints. Moreover, additional restrictions have extended its application. The conditional generative adversarial nets (conditional GAN or CGAN) proposed by Mirza [22] found that adding category constraints to GAN impelled the training process to become more stable and the final result more multiform. Their work converted the binary minimax game into the probabilistic binary minimax game. Odena (semi-GAN) [23] attempts to improve sample generation and classifier performance by training GANs with classifiers simultaneously. Their approach achieved the original intention of the author and shortened the training time on the premise of improving the quality of the generated sample. InfoGAN proposed by Chen [24] not only considers the classification effect of real data

but also tries to use the method of mutual information to add the category information of the generated sample to the training process. Auxiliary classifier generative adversarial nets (ACGANs) proposed by Odena [13] changes the GAN energy function to add the discrimination class error of the generated sample and the real sample. Their approach demonstrates that a complex latent coder could boost the generative sample's resolution. Figure 3 illustrated the development of CGANs.

Figure 3. This figure shows the development of conditional generative networks structures.

3. Materials and Methods

3.1. Previous Work: Synthetic Image Data from 3D Models

Learning appearance from source data is the main purpose of GAN. Two considerable difficulties affect GAN applications: (1) GAN cannot produce complex annotation data because its constraints information is only category annotation. (2) The generative images are unnatural because of the lack of geometric characteristics. Computer graphic methods can finish data synthesis but are rigorously lacking in appearance properties.

In Section 2.2, the CGAN was shown to solve the freedom of the GAN by increasing latent random space, in order to find a new way for CGAN, by using other resources to light the task of a generator. Odena [13,23] provided many robust ways to increase the quality of GAN results. The authors in [15] did not use a generator to produce images from the latent space before using a GAN to generate images. Computer graphic methods rendered a coarse image. They then used a GAN to refine the rendered images. Their work made the GAN produce labeled training data, which decreased the demands of visual intelligence.

Different from earlier GAN works, our visual data synthesis methods decrease the complexity of GAN tasks (Figure 4). First, a 3D–2D rendering process was introduced into the data synthesis pipeline. In previous work, we used a pipeline that generated multiple annotations images Figure 5 based on pose annotations from the ObjectNet3D dataset (a large-scale 2D–3D image dataset) [25] and 3D shapes from the ShapeNet dataset (a large-scale 3D shape dataset) [26]. The data synthesizing method from 3D models was separated into four parts: data collection (3D models), annotation upsampling (upsampling the object pose annotations from ObjectNet3D), image rendering (using Blender <https://www.blender.org/> scripts to render foreground images automatically), and background adding. Finally, we found that improving the background quality of synthetic images can enhance the accuracy of the visual intelligence algorithm (Figure 6).

Figure 4. Ideas of this paper's image and GANs. Different from methods only based on GAN, our works introduced 3D models into the generator.

Figure 5. Multi-annotation of previous work: image recognition (category), the image fine-grained classification subcategory, object detection (bounding box), object pose estimation (pose information), and image instance segmentation.

Unlike earlier GAN jobs, our work reduces GAN complexity (Figure 4). Background augmentation generative adversarial networks (BAGANs) do not directly generate foreground target information associated with the visual intelligence algorithm. This ensures any picture of our synthesis data has guaranteed foreground appearance. The BAGAN is responsible for generating related backgrounds only according to category information and pose annotations for foreground objects.

Figure 6. Test accuracy of diverse backgrounds. The figure exhibits the test accuracy (the test sets are ObjectNet3D and Pascal VOC 2012) of four models trained on four training sets. The results illustrate that the rendered images from 3D models help the classifier to recognize objects in natural images. *Syn_uniform*: the model's training sets are rendered images with uniform noise background; *Syn_nobkg*: the model's training sets are rendered images without a background; *Syn_SUN*: the model's training sets are rendered images with a background from the SUN database.

3.2. Background Augmentation Generative Adversarial Networks (BAGANs)

3.2.1. Importance of Augment the Synthesis Image's Background

Data-driven means that an algorithm should let data “speak”. Practical synthesis data should have the ability to make an algorithm learn more visual appearance. Figure 6 indicates that high-quality background helps the visual intelligence analysis salient foreground appearance. Therefore, it is a significantly challenging task to strengthen the background appearance of generating images using the powerful image generating capability of a GAN.

3.2.2. The Value Function of a BAGAN

Two basic models comprise a GAN: one is the generator model (denote as G , generator), and the other is the discriminator model (denote as D , discriminator). The task of G is to produce generating samples (denote as $D(x)$ or $D(G(z))$) from a random noise vector (denote as z). The generator would try to produce a generated sample $G(z)$ like a real sample.

Formula (1) is the value function of GAN. It urges the generator and discriminator to achieve a harmonious balance so that the generator could generate natural samples, where $\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)]$ is the value function corresponding to the discriminator, which means that the discriminator can identify whether an image is generated by the generator or from real samples. $\mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))]$ corresponds to the generator that obfuscated the discriminator so that the discriminator cannot judge the difference between the generated samples from the generator and the real samples. Through the game between the two modules, the formula will achieve the optimal solution when the generated samples are close to the real samples.

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

To control the random noise generation process, conditional generative adversarial networks (CGANs) [22] introduce the conditional constraints y into random noise $z + y$. The CGAN value function is described in Equation (2).

$$\min_G \max_D \mathcal{L}(D, G) = \mathbb{E}_{x \sim p_{data}(x|y)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log D(G(z|y))]. \quad (2)$$

Odena constructed auxiliary classifier generative adversarial nets (ACGANs) [13] and found that building a complicated structure for a random noise vector z into the generator G can yield more realistic samples. Moreover, adding an auxiliary classifier can smooth the training process for a GAN. Therefore, the ACGAN value functions were redesigned (Equation (3)).

$$\begin{aligned} \mathcal{L}_S &= \mathbb{E}[\log P(S = \text{real}|X_{\text{real}})] + \mathbb{E}[\log P(S = \text{fake}|G(z))] \\ \mathcal{L}_C &= \mathbb{E}[\log P(C = c|X_{\text{real}})] + \mathbb{E}[\log P(C = c|G(z))] \end{aligned} \quad (3)$$

where the discriminator aimed at maximize $\mathcal{L}_S + \mathcal{L}_C$, and the generator aimed to maximize $\mathcal{L}_C - \mathcal{L}_S$.

Background augmentation generative adversarial networks (BAGANs) are aimed at providing background images (denoted as x_{back} and $x_{\text{back}} = G(z)$) according to the foreground images (denoted as x_{syn}). The discriminator D would discriminate that background images x_{back} or the final images $x_{\text{final}} = x_{\text{back}} + x_{\text{syn}}$ was a real sample. Our final images are made up of two portable images: generated foreground images and background images. Based on the ACGAN, CGAN, and GAN, we redefined the value function (Equation (4)).

$$\begin{aligned} \mathcal{L}_S &= \mathbb{E}[\log P(S = \text{real}|X_{\text{real}})] \\ &+ (1 - \lambda) \mathbb{E}[\log P(S = \text{fake}|X_{\text{back}})] \\ &+ \lambda \mathbb{E}[\log P(S = \text{fake}|X_{\text{final}})] \\ \mathcal{L}_C &= \mathbb{E}[\log P(C = c|X_{\text{real}})] + \mathbb{E}[\log P(C = c|X_{\text{final}})]. \end{aligned} \quad (4)$$

For further explanation of Equation (4), λ is the parameter used to change the reference level of x_{back} . Considering the degree of integrity of x_{back} and x_{final} will have a particular impact on the training process of the BAGAN. If the foreground image x_{syn} is removed, the generated background image x_{back} should also be a complete image.

If the integrity of the background image x_{back} is considered too great, the BAGAN degenerate to the network sample generated by the ACGAN. If the integrity of the final image x_{final} is too great, the BAGAN would not generate the image background because of the relatively complete foreground image.

3.2.3. Composite Layer for Foreground Object Adding

Inspired by Zhao's approach [27], we found that the artifacts surrounding the foreground affect the reality of the BAGAN results. Fortunately, the object image input has four channels: red, green, blue, and alpha channels (RGBA).

In our work, alpha compositing is related to the compositing process of the final image and resizing operation of the rendering image. An RGBA image contains an external channel alpha in the storage, which used to be an element of the alpha compositing algorithm. Alpha compositing is a classical algorithm in computer graphics that combines an image with a background. This algorithm is useful for image rendering in computer graphics [16,17,28]:

$$C_{\text{res}} = C_{\text{obj}} \times \alpha_{\text{obj}} + C_{\text{back}} \times (1 - \alpha_{\text{obj}}) \quad (5)$$

where C_{obj} is the RGB channels of the foreground objects, and C_{back} is the RGB channels of the background objects. Figure 7 indicates the comparison of using alpha compositing or not.

General image resizing methods only consider an RGB image. However, using general algorithms to resize the alpha channels is harmful to the edge of the foreground. Resizing an image should use an upsampling method to supplement the channel value of the resulting missing region of the image when a small image is resized to a large one. To maintain the alpha channel information, we used the optimal alpha resizing method to perform the foreground image resizing process (Algorithm 1).

Algorithm 1 Optimal RGBA image resizing method.

Input: IMG_{in} ▷ IMG_{in} , Input foreground image (RGBA)
Output: IMG_{out} ▷ IMG_{out} , Resized foreground image (RGBA)

```

1:  $pixels \leftarrow IMG_{in}$ 
2: for  $y < -0$  to  $len(IMG_{in}.height)$  do
3:   for  $x < -0$  to  $len(IMG_{out}.width)$  do
4:      $C_{in}, \alpha \leftarrow pixels[x, y]$  ▷  $C_{in}$  is RGB channels value;
5:                                     ▷  $\alpha$  is alpha channels value
6:     if  $\alpha \neq 255$  then
7:        $C_{in} \leftarrow C_{in} \times \alpha \div 255$ 
8:        $pixels[x, y] \leftarrow (C_{in}, \alpha)$ 
9:     end if
10:  end for
11: end for
12:  $IMG_{out} = RESIZE(IMG_{in})$ 
13:  $pixels \leftarrow IMG_{out}$ 
14: for  $y < -0$  to  $len(IMG_{out}.height)$  do
15:   for  $x < -0$  to  $len(IMG_{out}.width)$  do
16:      $C_{out}, \alpha \leftarrow pixels[x, y]$  ▷  $C_{out}$  is RGB channels value;
17:                                     ▷  $\alpha$  is alpha channels value
18:     if  $\alpha \neq 255$  and  $\alpha \neq 0$  then
19:       if  $C_{out} \geq \alpha$  then
20:          $C_{out} \leftarrow 255$ 
21:       else
22:          $C_{out} \leftarrow 255 \times C_{out} \div \alpha$ 
23:       end if
24:        $pixels[x, y] \leftarrow (C_{out}, \alpha)$ 
25:     end if
26:   end for
27: end for
```

(a) simple copy

Figure 7. Cont.

(b) alpha channel

(c) alpha compositing

Figure 7. Results of background compositing algorithms. This figure presents the effect of different background compositing algorithms. (a) fills the foreground object into the background (black) with the alpha position. (b) shows the alpha channels in the same background. (c) presents the result of alpha compositing.

4. Results

4.1. Datasets

ObjectNet3D is a large-scale 2D–3D image datasets made by Stanford Computer vision and Geometry Lab [25]. Collected from MSCOCO, PASCAL VOC, IMAGENET, and so on, the images in ObjectNet3D are compelling and consist of 100 categories, 90,127 images, 201,888 objects, and 44,147 3D models. The annotation of ObjectNet3D contains not only the essential annotation in computer vision such as object category and object detection but also the corresponding 3D model, posture, and other advanced annotation. Therefore, it has a significant impact on automatic driving, augmented reality (AR), and other applications.

To prevent the classifier from seeing the 3D model related to the test data during the training process, the rendered image does not use the 3D models of the ObjectNet3D but selects the 3D models of the ShapeNet database. Unlike ObjectNet3D's 3D model data, ShapeNet's 3D model is better in quality than ObjectNet3D's, and the ShapeNet team proposed a way to measure the scores of 3D models to help us automatically capture high-quality 3D model information.

4.2. Evaluation Metrics

There is no reliable metric to explain the quality of the synthesized data. One of the more intuitive ways is to observe the differences in the results. In the following chapters, we present the result graphs of several different models for comparison. Another practical approach in this paper is to use computer vision tasks to verify the quality of the generated data, because the ultimate purpose of generating data is to provide training data for a computer vision algorithm.

Classification Model Selection. As a basic algorithm of the computer vision method, the visual cognitive task (or image classification) is used to measure the quality of different synthesizing data. In order to experiment with contrast, we use the VGG16 depth convolutional network [29] to train in different training data and evaluate the classifier in the same real image test set. Therefore, the classifier score (accuracy, precision, recall, and F-1 score) can help us to obtain an objective measurement when we cannot distinguish between good and bad images from the resulting figure. At the same time, this method can also avoid some of the subjective errors of the generated data.

The experimental computer configuration as follows: Intel (R) Core (TM) i7-5930k 3.5 GHz CPU, 64 GB RAM, and Nvidia Geforce (R) Titan X (Pascal) GPU.

4.3. Comparison with Different Generative Models

One of the significant improvements in the BAGAN is the use of multiple source inputs (rendered images and random noise) to solve the problems of traditional GANs. Therefore, to illustrate the benefits of the BAGAN, the most advanced algorithms of the single inputs of ACGAN (random noise) and cycle-GAN (images) were used as a comparison.

4.3.1. BAGAN vs. ACGAN

Based on Figure 8, the ACGAN has some foreground objects that are superior to the BAGAN, because the BAGAN foreground object is real image rendering with computer geometry characteristics. This ensures that any of the BAGAN's main targets related to tags have a good texture and have geometric features whose texture and geometric features are determined by the quality of the model. Moreover, regarding classification accuracy, the ACGAN can achieve only 83.32% with the VGG16 classifier on the ObjectNet image dataset. The BAGAN achieves 93.51% in the Objectnet3D model, which proves that the BAGAN generation result with a good foreground texture and good geometric features is superior to that of ACGAN.

Figure 8. Synthetic images produced by the ACGAN.

4.3.2. BAGAN vs. Cycle-GAN

Unlike the GAN, the cycle-GAN consists of two pairs of GANs: G_{AN_A} (G_A, D_A) is responsible for converting X_{syn} to X_{real} ; G_{AN_B} (G_B, D_B) converts X_{real} to X_{syn} . The cycle-GAN uses the binary game of the GAN and the game of two pairs of GANs. To achieve good image to image conversion effect. In the experiment, however, Figure 9 shows that the cycle-GAN changes more for the salient objects than for the background in the image. Therefore, the cycle-GAN is more suitable for works similar to image style transferring. Not only that, it can be seen from the graph that cycle-GAN changes the appearance information of foreground objects more, which means that the classification result of cycle-GAN is not superior to that of the ACGAN, although the foreground objects of the generated images have geometric characteristics. At the same time, from the score of the classifier, we can see that the training set score of the cycle-GAN is lower than that of the ACGAN. This shows that preserving the foreground appearance is vital for the data synthesis algorithm.

Figure 9. Synthetic images produced by the cycle-GAN.

4.4. Lambda Parameters

Based on Equation (4), λ is the parameter used to fix the reference level of x_{back} . The first part, $\mathbb{E}[\log P(S = real|X_{real})]$, was responsible for the source judgement of the real image, the second part, $(1 - \lambda)\mathbb{E}[\log P(S = fake|X_{back})]$, was responsible for the source judgment of the background image, and the third part, $\lambda\mathbb{E}[\log P(S = fake|X_{final})]$, was responsible for the source decision of the final synthetic image. These contents control the gradient effect of training samples on the network in the process of BAGAN training. The second studied the authenticity of the background image separately, while the third was responsible for the authenticity of the final image.

Furthermore, in experiments, the optimal choice of λ was in 0.70–0.75, and our model in 0.72 achieved the best performance. Figures 10 and 11 indicates the influence of different λ values.

1. If λ is 0, the BAGAN will only consider the integrity of the background images (x_{back}). After compositing the foreground and the background images, the BAGAN will fail to find a balance between x_{back} and x_{final} . Therefore, the generated background images look like random noise.
2. If λ is 0.25, the BAGAN will try to find a balance between the background and the foreground. Figure 10 shows that the BAGAN achieved a much better background than that when λ is 0.

3. If λ is 0.5, the generated backgrounds are more natural than $\lambda = 0$ and 0.25. This stems from the training balance between x_{back} and x_{final} .
4. After tuning the λ , when its value is 0.72, the BAGAN generates the best backgrounds.
5. If λ is 1, the BAGAN does not consider x_{back} , so the generated backgrounds resemble noise images again.

Figure 10. Synthetic images produced by the BAGAN. This figure displays the various λ BAGAN results and reveals the influence of λ on BAGANs.

Figure 11. Performance of the λ values of the BAGAN.

4.5. Effect of the Alpha Compositing Layer

An alpha compositing algorithm was built in the networks to enhance the edge appearance information for synthesizing vision data. In experiments, this algorithm increased classifier performance. Firstly, Figure 7 shows that alpha compositing algorithms can improve the edge appearance of the synthesizing final images. Secondly, Figure 12 shows the difference between using and not using a compositing layer. Thirdly, Table 1 indicates that using alpha compositing algorithms can slightly enhance the performance of the classifier when $\lambda = 0.5$ or 0.72. Figure 10 indicates that the generator can produce an available background. Using compositing algorithms can increase the classifier's precision score. However, when the background is not strong enough, alpha compositing algorithms do not increase the accuracy because, compared to the appearance of the background, the influence of edge appearance on the classifier was slight.

Compositing

None

Figure 12. Compositing layer. Compositing indicates generated samples with a compositing algorithm. None indicates generated samples without the compositing algorithm.

Table 1. This figure indicates the effect of using an alpha compositing algorithm, where BAGAN-CL refers to BAGANs using alpha compositing algorithms (discussed in Section 3.2.3). When $\lambda = 0, 0.25$, and 1, the generator produces a “noise-like” background, and the score of the classifier is not changed. When $\lambda = 0.5$ or 0.72, the training data generated by BAGAN-CL achieves a higher score compared with the BAGANs.

Trainingdata	Accuracy	Precision	Recall	F1-Score
BAGANs ($\lambda = 0$)	88.12%	92.06%	91.67%	91.63%
BAGANs-CL ($\lambda = 0$)	88.32%	92.06%	91.67%	91.51%
BAGANs ($\lambda = 0.25$)	90.53%	91.15%	94.77%	93.44%
BAGANs-CL ($\lambda = 0.25$)	90.52%	90.25%	95.54%	93.26%
BAGANs ($\lambda = 0.5$)	91.64%	94.65%	94.58%	94.62%
BAGANs-CL ($\lambda = 0.5$)	91.97%	96.03%	94.55%	95.28%
BAGANs ($\lambda = 0.72$)	93.12%	94.23%	97.64%	95.90%
BAGANs-CL ($\lambda = 0.72$)	93.51%	95.27%	97.02%	96.14%
BAGANs ($\lambda = 1$)	90.42%	91.54%	94.53%	93.01%
BAGANs-CL ($\lambda = 1$)	90.39%	92.22%	96.509%	94.12%

4.6. Classification Results of Different Training Data

In the classification experiment, we use accuracy and F1-score to evaluate the performance and sensitivity of the classification model ($F1_score = \frac{2 \times precision \times recall}{precision + recall}$). Figure 11 illustrates the comparison with other methods. In data synthesis, the classification score of ACGAN and cycle-GAN shows that simple inputs (only the random noise vector or the image) make the performance of classifiers worse than that of multiple input sources. Data synthesis benefited from reducing the “burden” of the GAN.

Compared to different backgrounds images, Figure 13 illustrated that increasing the quality of backgrounds helped improve the performance of intelligent visual algorithms (Table 2).

Figure 13. To enhance image appearance information, three types of background were used: rendered images x_{syn} without a background; rendered images x_{syn} with a uniform noise background; rendered images x_{syn} with a randomly selected background from the SUN database [30].

In the classification score, the BAGAN with an alpha compositing algorithm achieved the best performance (accuracy: 93.51%; F1-score: 96.14%).

Table 2. Classification results of different training sets. Training sets involved adding backgrounds with different methods, generated samples, and natural images. ObjNet_3D: Natural images from a large-scale 2D–3D annotation image database [25]. No_bkg: synthesized data are rendered images from 3D models. Uniform_bkg: Synthesized data are composed of rendered images x_{syn} and uniform noise images (the same size with rendered images). SUN_bkg: Synthesized data are composed of rendered images and random selected images of the SUN database [30]. Cycle_GANs: Synthesized data are composed of rendered images and generated images by the cycle-GAN. BAGAN: Synthesized data are composed of rendered images and generated samples by the BAGAN with various λ .

Trainingdata	Accuracy	Precision	Recall	F1-Score
No_bkg	87.98%	90.58%	94.23%	92.37%
Uniform_bkg	89.99%	91.80%	95.25%	93.50%
SUN_bkg	92.24%	93.65%	96.19%	94.90%
ObjNet_3D	90.29%	92.14%	95.47%	93.78%
Cycle_GANs	73.64%	77.52%	82.17%	79.78%
BAGANs ($\lambda = 0$)	88.12%	92.06%	91.67%	91.63%
BAGANs ($\lambda = 0.25$)	90.53%	91.15%	94.77%	93.44%
BAGANs ($\lambda = 0.5$)	91.64%	94.65%	94.58%	94.62%
BAGANs ($\lambda = 0.72$)	93.12%	94.23%	97.64%	95.90%
BAGANs ($\lambda = 1$)	90.42%	91.54%	94.53%	93.01%

5. Discussion

This work study using 3D models and a GAN produced visual data. To improve visual intelligence algorithms, providing massive guaranteed annotated data was effective. This conclusion was made by Zhang [4] and is confirmed by our classification results. Our synthesized data provides massive labeled data for deep network training and achieves the best performance, with an accuracy score of 93.51%, a precision of 95.27%, a recall score of 97.02%, and an F1-score of 96.14%. Compared with the manually labeled data, our synthetic images look unnatural, but the synthetic data help the classifier recognize objects in natural images.

Complexity of our study. Data synthesizing algorithms are vital for artificial intelligence in training and data preprocessing, and are irrelevant to the inference speed and running speed of the algorithm in the deployment phase. The main purpose of data synthesis is to help supervised or semi-supervised artificial intelligence algorithms to achieve higher accuracy. Therefore, in previous data synthesizing methods, the analysis of the complexity of the algorithm is almost empty [2,12,13,18,23,24].

Benefits of Augmented Reality. The object recognition algorithm of AR requires a great deal of tag data for the business scene in the practical application process. In some special industrial scenes, the annotations of the picture sometimes need professional knowledge. Therefore, using data-driven methods to develop AR applications is a very time-consuming and costly approach. However, our research relies on GANs and 3D models, which are easy to obtain in the relevant industrial chain, and a small number of models can generate a large amount of tag data, which is a valuable attribute in the real industrial scene. In this paper, we do not use a specific task scene to verify the application of AR target recognition, and this is due to the lack of relevant images and 3D model data. We cannot produce validation data. However, our results have achieved good results in the generic categories, which is sufficient to demonstrate the effectiveness of this method in the AR scenario.

Significance and limitations of our works. Training data is crucial to developing computational algorithms, especially in the augmented reality domain. For augmented reality, large numbers of training data can apply AR to all aspects of generation and life. Our method can automatically generate image training samples according to the types of 3D model which is very conducive to application and deployment of immersive interactive systems. Compared with artificial marker training sets,

our synthetic data can effectively control the quantity and quality of samples. Moreover, for industrial development, high-quality data are easier to obtain from real images than from real photos. Therefore, our approach is to promote immersive HCI with good exploration.

Our work also provides an alternative to fixing GANs in the field of data synthesis, a significant strength of this work. However, the generated samples by the BAGAN are still poor compared with natural images.

The cycle-GAN performed well in other tasks, such as image-to-image conversion. The energy function should be considered for compositing in a BAGAN in follow-up research.

Major findings in this article are as follows:

1. Using 3D shapes can decrease the complex task of GANs in data synthesis.
2. Designing a BAGAN makes synthesis images more natural.
3. Using alpha compositing algorithms increases foreground edge appearance.
4. Training visual data produced by our method enhances the classifier, with 93% accuracy.

Considering subsequent research, there are four ways our work can be continued:

- Wang [31] applied visual attention algorithms to video saliency detection. Adding the attention module may allow a GAN to produce an image background better.
- In light of stacked stages from networks (mentioned by Jaime [32]), manufacturing multi-stage networks is perhaps an adequate approach to fixing the problem of high-resolution images.
- Fusing the loss function of cycle-GAN image to image.
- Considering classification results, saliency object recognition [33] could be an alternative research direction.

Author Contributions: Y.M. proposed the idea of this paper; K.L., X.Q. and H.B. reviewed this paper and provided information; Y.M. and Z.G. conceived and designed the experiments; Y.M. and X.X. performed the experiments; X.X. reviewed the codes in this paper. Y.M. wrote this paper.

Funding: This work was partially funded by the National Key R&G Program of China under Grant (2018YFB1004600).

Acknowledgments: Yu-Zhe Chang, Yi-jin Xiong, and Qi Chen have contributed to this paper by organizing the materials and literature research. Yan Ma thanks Ruo-Ning Cao for her patience and understanding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HCI	human–computer interaction
GANs	generative adversarial networks
PCA	principal component analysis
ILSVRC	ImageNet Large Scale Visual Recognition Challenge

References

1. Richer, R.; Maiwald, T.; Pasluosta, C.; Hensel, B.; Eskofier, B.M. Novel human computer interaction principles for cardiac feedback using google glass and Android wear. In Proceedings of the 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Cambridge, MA, USA, 9–12 June 2015; pp. 1–6.
2. Hong, J.I. Considering privacy issues in the context of Google glass. *Commun. ACM* **2013**, *56*, 10–11. [[CrossRef](#)]
3. Evans, G.; Miller, J.; Pena, M.I.; Macallister, A.; Winer, E.H. Evaluating the Microsoft HoloLens through an augmented reality assembly application. *Proc. SPIE* **2017**, *10197*. [[CrossRef](#)]
4. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [[CrossRef](#)]

5. Jain, N.; Kumar, S.; Kumar, A.; Shamsolmoali, P.; Zareapoor, M. Hybrid deep neural networks for face emotion recognition. *Pattern Recognit. Lett.* **2018**, *115*, 101–106. [[CrossRef](#)]
6. Wang, Z.; Lu, D.; Zhang, D.; Sun, M.; Zhou, Y. Fake modern Chinese painting identification based on spectral-spatial feature fusion on hyperspectral image. *Multidimens. Syst. Signal Process.* **2016**, *27*, 1031–1044. [[CrossRef](#)]
7. Sun, M.; Zhang, D.; Ren, J.; Wang, Z.; Jin, J.S. Brushstroke based sparse hybrid convolutional neural networks for author classification of Chinese ink-wash paintings. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 626–630.
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, 2012, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
10. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
11. Tipping, M.E.; Bishop, C.M. Probabilistic Principal Component Analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1999**, *61*, 611–622. [[CrossRef](#)]
12. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
13. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
14. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
15. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning from Simulated and Unsupervised Images through Adversarial Training. *CVPR* **2017**, *2*, 5.
16. Zongker, D.E.; Werner, D.M.; Curless, B.; Salesin, D.H. Environment Matting and Compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 1999; pp. 205–214.
17. Porter, T.; Duff, T. Compositing Digital Images. *SIGGRAPH Comput. Graph.* **1984**, *18*, 253–259. [[CrossRef](#)]
18. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
19. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv* **2017**, arXiv:1701.04862.
20. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv* **2017**, arXiv:1701.07875.
21. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5769–5779.
22. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784v1.
23. Odena, A. Semi-Supervised Learning with Generative Adversarial Networks. *arXiv* **2016**, arXiv:1606.01583.
24. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2172–2180.
25. Xiang, Y.; Kim, W.; Chen, W.; Ji, J.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S. ObjectNet3D: A Large Scale Database for 3D Object Recognition. In *European Conference Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2016.
26. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012v1.

27. Zhao, D.; Zheng, J.; Ren, J. Effective Removal of Artifacts from Views Synthesized using Depth Image Based Rendering. In Proceedings of the International Conference on Distributed Multimedia Systems, Vancouver, BC, USA, 31 August–2 September 2015; pp. 65–71.
28. Smith, A.R.; Blinn, J.F. Blue Screen Matting. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*; ACM: New York, NY, USA, 1996; pp. 259–268.
29. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
30. Xiao, J.; Hays, J.; Ehinger, K.A.; Oliva, A.; Torralba, A. SUN database: Large-scale scene recognition from abbey to zoo. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–8 June 2010; pp. 3485–3492.
31. Wang, Z.; Ren, J.; Zhang, D.; Sun, M.; Jiang, J. A Deep-Learning Based Feature Hybrid Framework for Spatiotemporal Saliency Detection inside Videos. *Neurocomputing* **2018**, *287*, 68–83.
32. Zabalza, J.; Ren, J.; Zheng, J.; Zhao, H.; Qing, C.; Yang, Z.; Du, P.; Marshall, S. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. *Neurocomputing* **2016**, *185*, 1–10. [[CrossRef](#)]
33. Han, J.; Zhang, D.; Hu, X.; Guo, L.; Ren, J.; Wu, F. Background Prior-Based Salient Object Detection via Deep Reconstruction Residual. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1309–1321.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).