Peer-reviewed version available at Symmetry 2018, 10, 734; doi:10.3390/sym10120734

Article

BAGAN:Effective data generation based on GAN augmented 3D synthesizing§

Yan Ma [1], Kang Liu [1]* , Zhibin Guan [1], Xinkai Xu[2], Xu Qian[1], Hong Bao[2]

[1] School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing, China
[2] Beijing Union University, Beijing, China
* Correspondence: kangliu@cumtb.edu.cn; Tel.: +86-188-0131-1165
§ This paper is an extended version of our paper published in BICS2018[1] .

Abstract: Augment reality (AR) is crucial for immersive human-computer interaction (HCI) and vision of artificial intelligence (AI). Labeled data drove object recognition in AR. However, manual annotating data is expensive and labor-intensive, and furthermore, scanty labeled data limits the application of AR. Aiming at solving the problem of insufficient training data in AR object recognition, an automated vision data synthesis method called BAGAN is proposed in this paper based on the 3D modeling and GAN algorithm. Our approach has been validated to have better performance than other methods through image recognition task on natural image database ObjectNet3D. This study can shorten the algorithm development time of AR and expand the application scope of AR, which is of great significance to immersive interactive systems.

Keywords: object recognition; image data synthesizing; Human-computer interaction; data synthesizing for immersive HCI; generative adversarial nets;BAGAN

1. Introduction

Augment Reality (AR) is an essential part of the immersive human-computer interaction. Using advanced sensing system, AR provides an essential platform for human-computer interaction, such as Google Glass[2,3] and Microsoft HoloLens[4]. AR has much prospect in the fields of medical, industrial, office education and so on. 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 had many significant breakthroughs[5]. More and more computer vision algorithms based on deep learning were built and 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 the development. Due to the broad applications of AR technology, the visual data of AR needs rich multi-class visual labeled data. More than that, people have increasing requirements for advanced visual tasks of AR with the development of human-computer interaction.

[2,3] and Microsoft HoloLens[4] presented the strengthen ways of visual interaction because advanced visual intelligence can finish complex visual tasks than before. High-performance visual intelligence (or computer vision) algorithm is vital for Augment Reality.

These factors lead to two formidable problems, on the one hand, who state the quantity of annotated data impact the performance of an algorithm. On the other side, annotating data become an arduous work, because of the increasing advanced visual tasks required by interaction design[6–9].

To solve the lack of annotation data, many researchers find two possible ways: (1) Use unsupervised vision learning algorithms to decrease the demands of annotated data. (2) Get more annotations data for supervised learning using automated methods.

Different supervised learning algorithms, 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 the task which is difficult to deal with supervised learning because it does not need the guidance of annotation in data. However, the research of unsupervised learning is in the ascendant, and it can not replace the supervised learning algorithm.

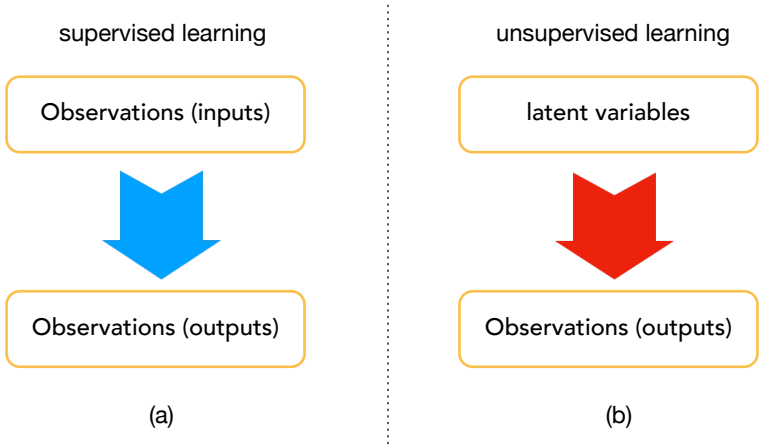

Figure 1. Difference between supervised learning and unsupervised learning. Supervised learning uses annotation guidance to draw learning-task-related conclusions about the data. Unsupervised learning uses the latent factors in data to conclude the relationship between data and the corresponding the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

Visual data enhancement algorithm is designed to reduce the cost of manual marking, and it is also one of the traditional problems 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 annotate by the human.

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), the converted image and the original image will be treated as two different images with the same annotation. Krizhevsky mentioned in this paper [10] 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 (proposed in) 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[11] have been proposed and proved to improve the effectiveness of the model significantly.

However, traditional visual data augmentation methods have two congenital shortage. First, simply converting visual data could not bring diversity to the appearance distribution of original data. That made visual intelligence algorithms cannot benefit lots form 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, we cannot use rotation methods (one of traditional visual data augmentation which rotate a image as a new image) dose not make sense because the bounding box annotation can not be transformed directly.

Generated visual data augmentation, making various image data by unsupervised learning, is an effective way to solve the shortage 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 original data. In Krizhevsky's paper [10] (2012), they not only use the traditional visual data enhancement algorithm but also change

hidden features, extracted by Primary Components Analysis (PCA) [12], to enrich the appearance of original data. In 2014, generative adversarial network (GAN)[13] provide an alternative way to produce various visual data from the original data. Although GAN is an algorithm which computed to approximate the distribution of any data, uncontrollable random processes cannot provide useful data for supervised learning. In 2016, Odena[14]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, deep convolutional generative adversarial network (DCGAN) [15] synthesize more natural and high-resolution images by using complex random noise generation, and unique lose function.

However, two difficult research gap existed in generated visual data augmentation. First, guaranteed quality of each image results in serious data mismatch problems in model training progress. Secondly, created visual data augmentation cannot make the multiple annotations for visual data. At presents, that only allow image classification data annotations.

In IEEE conference on computer vision and pattern recognition 2017 (CVPR2017), Shrivastava[16] give a positive inspire, they are not using a generator to product images from the latent space, before using GAN to generate images. Computer graphic methods rendered a corse image. After that, they use GAN to refine the rendering images. Their work makes GAN produce labeled training data, which decreased the demands of visual intelligence. Finally, we generate ideal training data. As an endorsement, this article gets CVPR2017 best paper award. In this article,

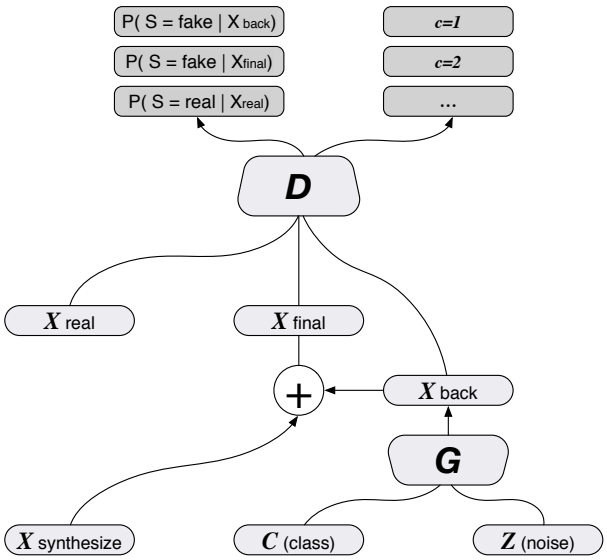

Figure 2. Background augmentation generative adversarial nets

using foreground images (rendering from 3D shapes) and random noises as input, we proposed Background Augmentation GAN to decrease the complex task of GAN in data synthesis. Our work could generate guaranteed labeled vision data. To increase the edge appearance of the foreground object, we implement compositing layer which applied alpha compositing algorithms [17,18] composited foreground and background images. Figure 2 illustrates the schematic. In experiments, contrasting with Cycle-GAN[19] (73.64% accuracy) and ACGANs[14] (85.23% accuracy), our methods got the best performance (93.51% accuracy).

2. Related works

2.1. Generative adversarial networks

Generative adversarial networks are the epoch-making unsupervised learning algorithm framework. In 2014, Goodfellow[13] introduced the novel unsupervised learning algorithm generative adversarial nets. It brought three breakthroughs for deep learning: (1) escape Markov chain learning methods; (2) high capability of sophisticated loss functions; (3) it is always valid, even if the probability density of target domain is challenging to calculate. However, it also has two shortages: (1) the freedom from simple constraints increases the uncertainty of the final result and the difficulty of the training process; (2) the generator will get better, and the discriminator's return gradient will be smaller and smaller, making the network hard to converge. Radford[15] introduced GAN into computer vision through deconvolutional and convolutional networks. The deep convolutional generative adversarial nets (DCGAN) had the elaborated network architecture. It animated the training process more stable, 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[20–22] found the reason of the two problems by a mathematical way, and it solved problems through 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

Let's revisit one of the critical reasons that cause GAN training and design difficulties: the freedom from simple constraints. Is it possible to solve part of the problems with GAN by adding constraints? Yes, it is. Moreover, additional restrictions extended its application. The conditional generative adversarial nets(Conditional GAN or CGAN) proposed by Mriza[23] found that adding category constraints to GAN impelled the training process more stable and the final result more multiform. Their work converted the binary minimax game into the probabilistic binary minimax game. Odena (Semi-GAN)[24] attempts to improve sample generation and classifier performance by training GAN 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[25] 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 (ACGAN) proposed by Odena[14] changes the GAN energy function to add the discrimination class error of the generated sample and the real sample. Their approach demonstrates that complex latent coder could boost generative sample's resolution.

One of the essential reasons that cause GAN training and design difficulties is freedom from simple constraints. Additional constraints could solve the problem of GAN. The conditional generative adversarial nets(Conditional GAN or CGAN) proposed by Mriza[23] found that adding category constraints to GAN impelled the training process more stable and the final result more multiform. Odena (Semi-GAN)[24] attempted to improve sample generation and classifier performance by training GAN with classifiers simultaneously. InfoGAN proposed by chen[25] 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 (ACGAN) proposed by Odena[14] changes the GAN energy function to add the discrimination class error of the generated sample and the real sample.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2018 doi:10.20944/preprints201811.0252.v1

Peer-reviewed version available at Symmetry 2018, 10, 734; doi:10.3390/sym10120734

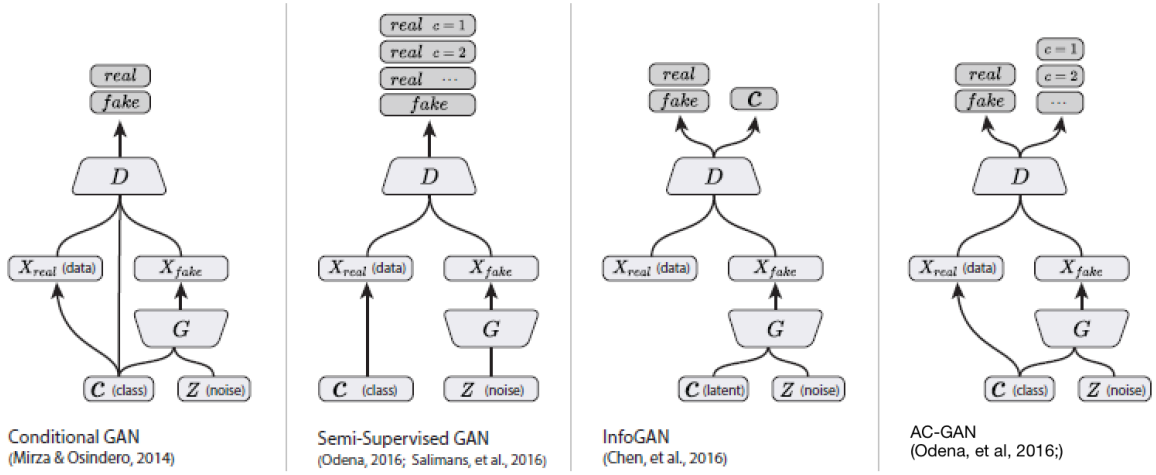

Figure 3. This figure shows the development of conditional generative networks structures.

Their approach demonstrates that complex latent coder could boost generative sample's resolution. Figure 3 shows the brief history of conditional generative networks structures[1].

3. Materials and Methods

3.1. Previous work: Synthetic image data from 3D models

Learning appearance from source data is the significant purpose of GAN. Two considerable difficulties affect GAN application: (1) GAN cannot produce complex annotations because of criticism information restrains only classes information. Because results are a start of random noise, few factors to affect GAN make an excellent orientated image. (2)The generative images are unnatural because of the lack of geometric characteristics. Computer graphic methods could finish data synthesis but rigorously lacking in appearance properties.

In Section 2.2, CGAN solved the freedom of GAN by increasing latent random space, To find a new way for CGAN, using other resources to light the task of a generator. Odena [14,24]provided many robust ways to increase the quality of GAN results. How about introducing other resources to rescue task complexity of generator? The authors[16] give a positive inspire, they are not using a generator to product images from the latent space, before using gan to generate images. Computer graphic methods rendered a corse image. After that, they use GAN to refine the rendering images. Their work makes 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 (Figure4). First 3D-2D rendering process was introduced into the data synthesis pipeline.

Unlike earlier GAN jobs, our work reduces GAN complexity (Figure 4). BAGAN does not directly generate foreground target information associated with the visual intelligence algorithm. That ensures any picture of our synthesis data has guaranteed foreground appearance. BAGAN is responsible for generating related backgrounds only according to category information and pose annotations for foreground objects. In previous work, we used a pipeline that generated multiple annotations images 5 based on pose annotations from the ObjecNet3D dataset(large scale 2D-3D image dataset)[26] and 3D shapes from the ShapeNet dataset (large scale 3D shape dataset)[27]. And

[1] This figure comes from https://www.cnblogs.com/punkcure/p/7873566.html

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2018 doi:10.20944/preprints201811.0252.v1

Peer-reviewed version available at Symmetry 2018, 10, 734; doi:10.3390/sym10120734

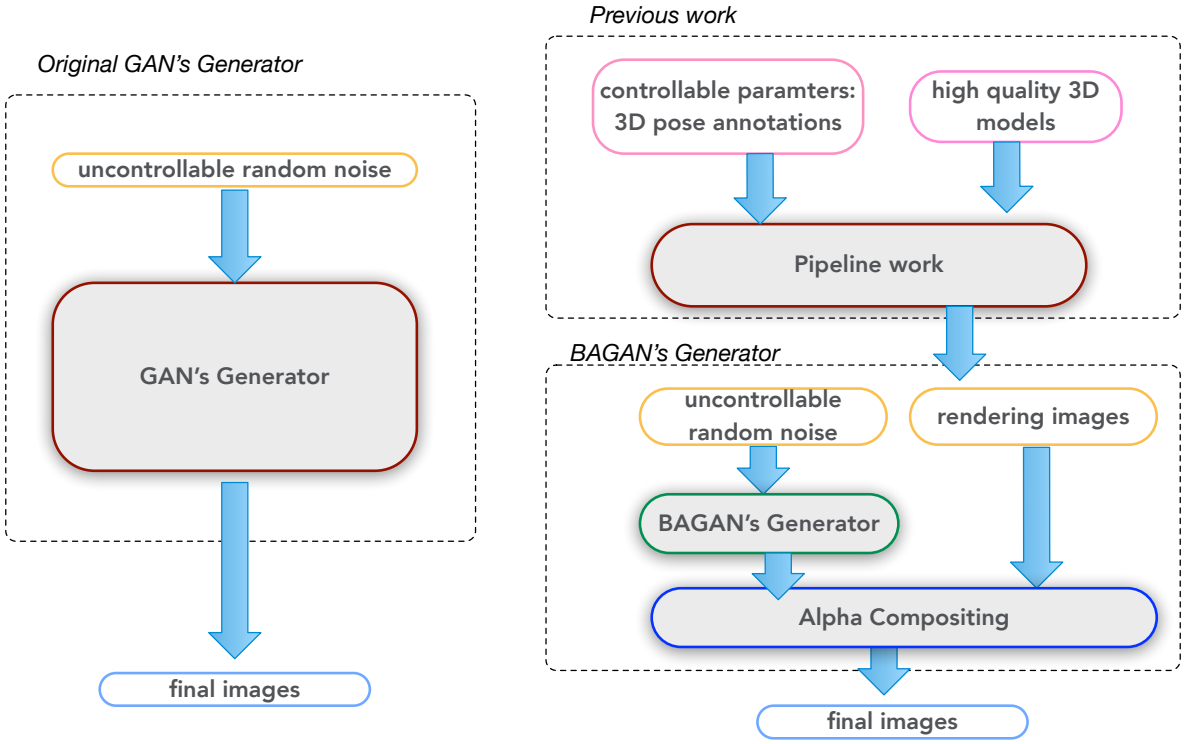

Figure 4. Idea comparison between this paper image and GANs. Different with the methods only based on GAN, our works introduced 3D models into the generator.

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

we found that improving the background quality of synthetic images can enhance the accuracy of the visual intelligence algorithm (Figure 6).

3.2. Background augmentation generative adversarial networks (BAGAN)

3.2.1. Importance of augment the synthesis image's background

Data-driven means that an algorithm should let data 'speak'. A practical synthesis data should have the ability that makes an algorithms learning more visual appearance than others

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2018 doi:10.20944/preprints201811.0252.v1

Peer-reviewed version available at Symmetry 2018, 10, 734; doi:10.3390/sym10120734

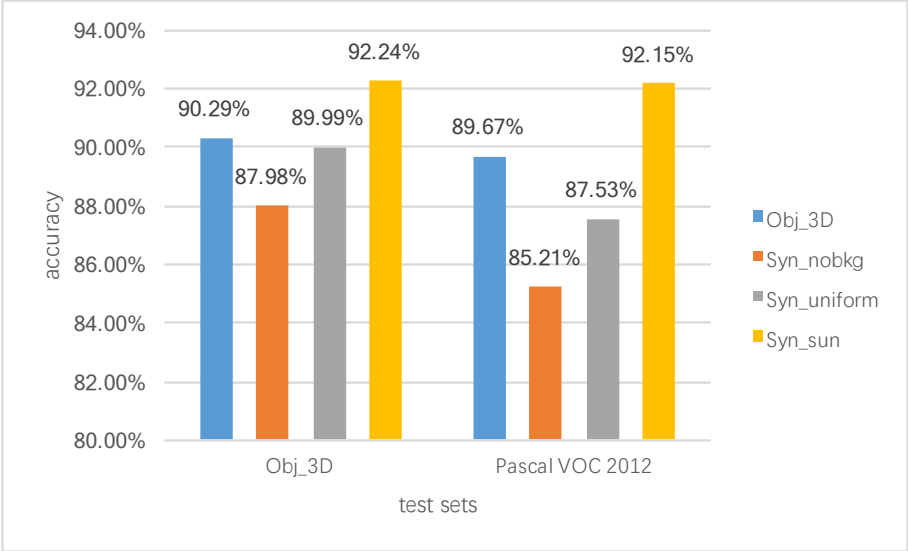

Figure 6. Test accuracy of diverse backgrounds. The figure exhibits the test accuracy (the test sets are ObjectNet3D and Pascal VOC 2012) of four models trained on four training sets. The results illustrate that the rendering images from 3D models support the classifier to recognize objects in natural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

data. 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 of generating images using GAN powerful image generating capability.

3.2.2. The value function of BAGAN

Two basic models composited Generative adversarial networks: one is the generator model (denote as G, generator), the other is the discriminator model (denote as D, discriminator). The task of G was producing generating samples (denote as $D(x)$ or $D(G(z))$) from random noise vector (denote as z). For G, it would try to produce a generated sample $G(z)$ like a real sample.

The responsibility of D was ruling the inputs were coming from real x or fake $G(z)$. Equation1 explained that G and D would make two-player minimax game.

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

To control the random noise generation process, conditional generative adversarial networks (CGAN)[23] introduce the conditional constraints y into random noise $z+y$. The CGAN value function is described in Equation 2.

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

Odena constructed Auxiliary Classifier Generative Adversarial Nets(ACGAN) [14] and found building a complicated structure for random noise vector z into the generator G can get more realistic samples. Moreover, adding an auxiliary classifier can smooth the training process for GAN. Therefore, the ACGAN value function are redesigned(Equation 3).

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

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

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

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

For further explanation of Equation4, $lambda$ is the parameter used to change the reference level of x_{back}. Because considering the degree of integrity of x_{back} and x_{final} will have a particular impact on the training process of BAGAN. If removed the foreground image x_{syn}, the generated background image x_{back} should also be a complete image. Considering the integrity of the background image x_{back} too much, the BAGAN would degenerate to the network sample generated by ACGAN. Considering the integrity of the final image x_{final}, BAGAN could develop in the direction of not generating the image background because of the relatively complete foreground image.

3.2.3. Composits Layer for foreground object adding.

Inspired by Zhao's approach [28], we found that the artifacts surrounding foreground affect the reality of BAGAN results. Fortunately, the object image input has four channels: red, green, blue and alpha channels (RGBA).

In our work, alpha compositing related to the compositing process of the final image and the resizing process of the rendering image. An RGBA image contains an external channel alpha in the storage which used to be an element of alpha compositing algorithm. Alpha compositing is a classical algorithm in computer graphics, which combines an image with a background. This algorithm is useful for image rendering in computer graphics. [17,18,29].

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

Where, C_{obj} is the RGB channels of foreground objects, C_{back} is the RGB channels of background objects. Figure 7 indicates the comparison of alpha compositing and copy an object to the background.

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. Because resizing an image should use an upsampling method to supplement the channel value of the resulting missing region of the image when resizing a small image to a big one. To maintain the alpha channels information, we use the optimal alpha resizing method to perform the foreground image resizing process (Algorithm 1).

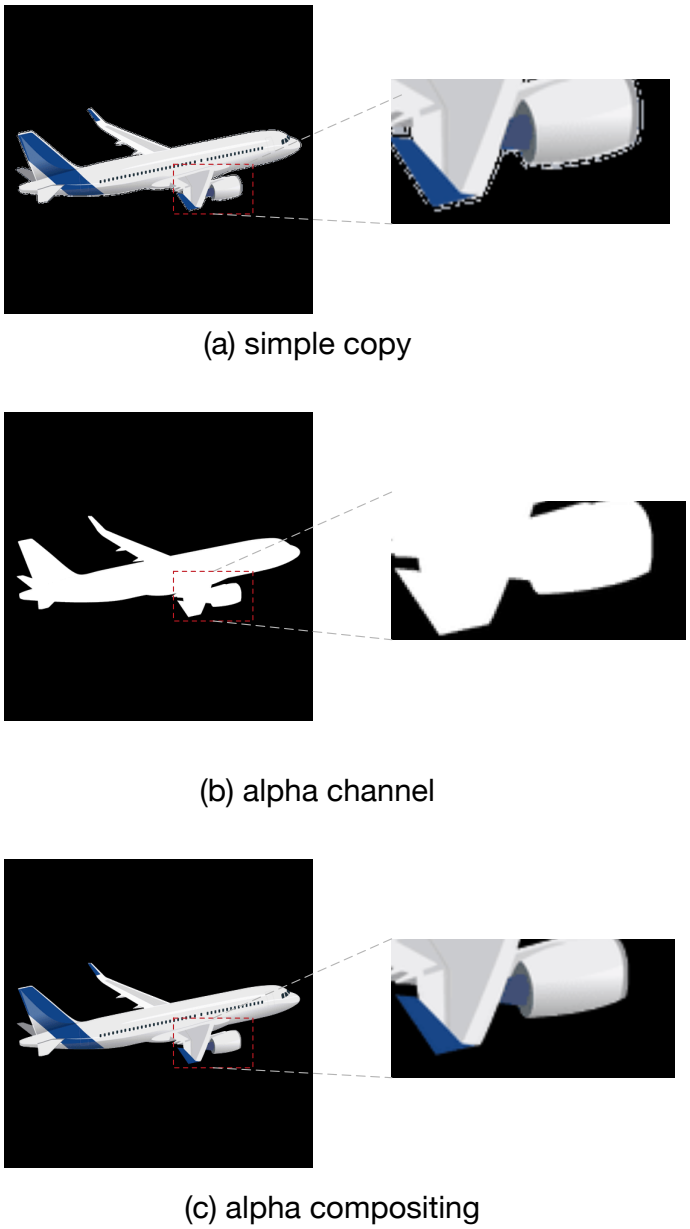

Figure 7. Results of background compositing algorithms. This figure presents the effect of different background compositing algorithms. (a) fills the foreground object into the background (black) with the alpha position. (b) shows the alpha channels in the same background. (c) presents the result of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

4. Results

4.1. Datasets

ObjectNet3D is a large-scale 2D-3D image datasets made by Standford Computer vision and Geometry Lab [26]. 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.

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)

$pixels \leftarrow IMG_{in}$
for $y <- 0$ *to* $len(IMG_{in}.height)$ **do**
  for $x <- 0$ *to* $len(IMG_{out}.width)$ **do**
    $C_{in}, \alpha \leftarrow pixels[x, y]$ ▷ C_{in} is RGB channels value;
    ▷ α is alpha channels value
    if $\alpha \neq 255$ **then**
      $C_{in} \leftarrow C_{in} \times \alpha \div 255$
      $pixels[x, y] \leftarrow (C_{in}, \alpha)$
    end if
  end for
end for
IMG_{out} = RESIZE(IMG_{in})
$pixels \leftarrow IMG_{out}$
for $y <- 0$ *to* $len(IMG_{out}.height)$ **do**
  for $x <- 0$ *to* $len(IMG_{out}.width)$ **do**
    $C_{out}, \alpha \leftarrow pixels[x, y]$ ▷ C_{out} is RGB channels value;
    ▷ α is alpha channels value
    if $\alpha \neq 255$ and $\alpha \neq 0$ **then**
      if $C_{out} \geq \alpha$ **then**
        $C_{out} \leftarrow 255$
      else
        $C_{out} \leftarrow 255 \times C_{out} \div \alpha$
      end if
      $pixels[x, y] \leftarrow (C_{out}, \alpha)$
    end if
  end for
end for

In order to avoid the classifier to see 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.

As the 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[30] to train in different training data and evaluate the classifier in the same real image test set. So the classifier score (accuracy, precision, recall, and F-1 score) can help us to get the 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, 64GB RAM, and Nvidia Geforce (R) Titan X (Pascal) GPU.

4.3. Comparison with different generative models

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

4.3.1. BAGAN vs ACGAN

Figure 8. Synthetic images produced by ACGAN. Target sets is cifar-10 database.

Figure 9. Synthetic images produced by BAGAN. This figure displayed the various *lambda* BAGANs results and revealed the influence of λ on BAGANs.

From the result of image, ACGAN has some foreground objects better than BAGAN, because BAGAN foreground object is real image rendering with computer geometry characteristic. This ensures that any BAGAN's main targets related to tags have good texture and geometric features whose texture and geometric features are determined by the quality of the model. Moreover, on the result of classification accuracy, ACGAN can achieve only 83.32% classification accuracy with VGG16 classifier on ObjectNet image dataset. BAGAN achieves 93.51% values in OBjectnet3D model and proves that BAGAN generation result with good foreground texture and geometric feature is better than ACGAN from classification precision.

4.3.2. BAGAN vs Cycle-GAN

Unlike GAN, Cycle-GAN consists of two pairs of GAN: GAN_A (G_A, D_A) is responsible for converting X_{syn} to X_{real} ; GAN_B (G_B, D_B) converts X_{real} to X_{syn}. Cycle-GAN uses the binary game of GAN and the game of two pairs of GAN. To achieve good image to image conversion effect. In the experiment, however, Figure 10 shows that the Cycle-GAN changes more for the salient objects than for the background in the image. Therefore, Cycle-GAN is more suitable for the works similar to the image style transferring. Not only that, it can be seen from the graph that Cycle-GAN changes the appearance information of foreground objects more, this phenomenon causes the classification result of Cycle-GAN is not better than 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 Cycle-GAN is lower than that of ACGAN. That shows that preserving the foreground appearance is vital for the data synthesis algorithm.

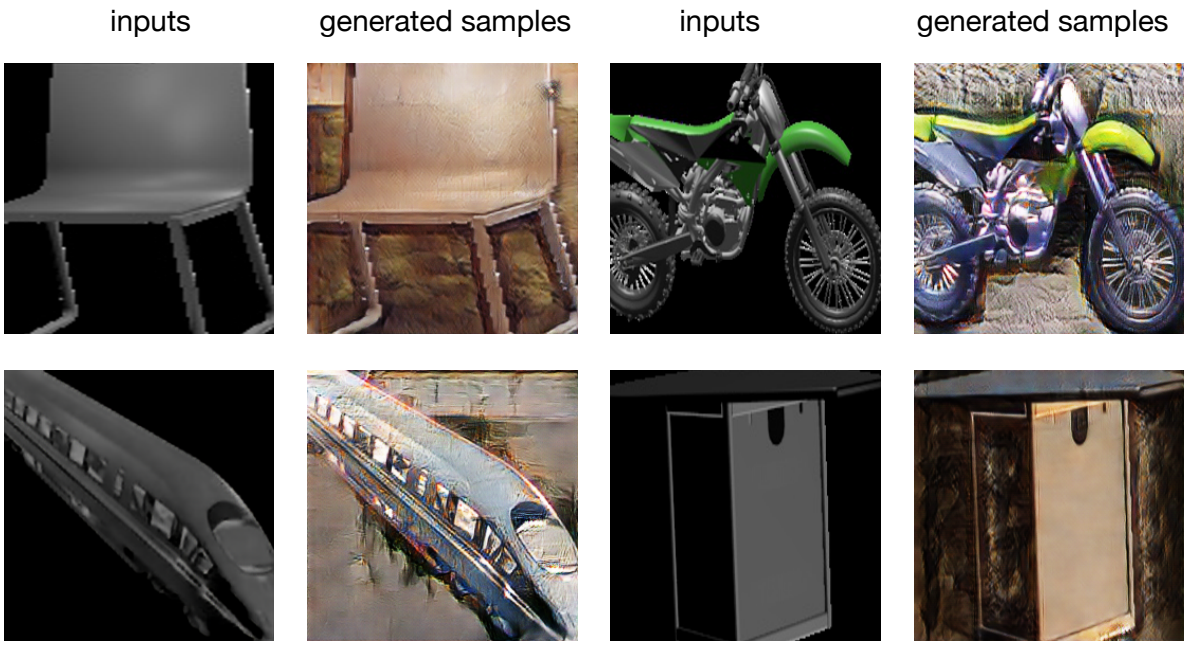

Figure 10. Synthetic images produced by Cycle-GAN.

4.4. Lambda parameters

λ is the parameter used to fix the reference level of x_{back} (described by Euquation 4). Because considering the degree of integrity of x_{back} and x_{final} will have a particular impact on the training process of BAGAN. If removed the foreground image x_{syn}, the generated background image x_{back} should also be a complete image. Considering the integrity of the background image x_{back} too much, the BAGAN would degenerate to the network sample generated by ACGAN. Considering the integrity of the final image x_{final} , BAGAN could develop in the direction of not generating the image

background because of the relatively complete foreground image. Figure 9 indicates the influence of different λ. If λ is 0, BAGAN would only consider about the integrity of background images (x_{back}). After compositing the foreground and the background images, BAGAN would fall to find the balance between x_{back} and x_{final}.Therefore, the generated background images looks like random noise.

1. If λ is 0.25, BAGAN would try to find balance between the background and the foreground. Figure 9 pointed out that BAGAN got much better background than λ is 0.
2. If λ is 0.5, the generated backgrounds are more natural than $\lambda = 0$ and 0.25. That stems from the training balance between x_{back} and x_{final}.
3. After tuning the λ, when its value is 0.72, BAGAN generated the best backgrounds.
4. If λ is 1, BAGAB would not consider the x_{back}, therefore the generated backgrounds are like noise image again.

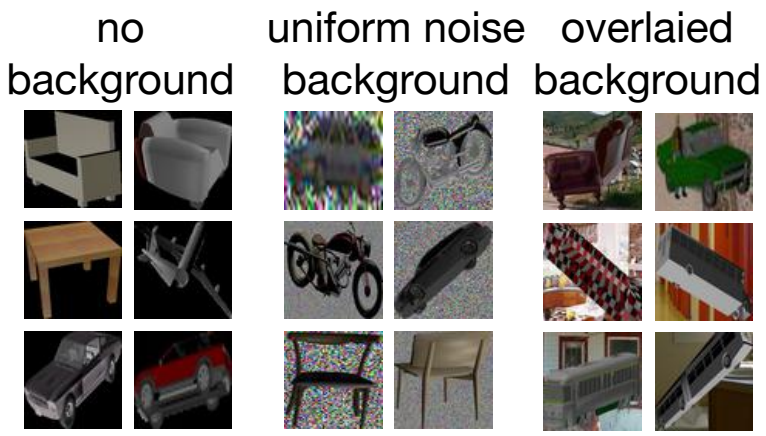

Figure 11. To enhance images appearance information, three types background were used: rendering images x_{syn} without background; rendering images x_{syn} with uniform noise background; rendering images x_{syn} with randomly selected background from SUN database [31].

4.5. Effectiness of alpha compositing layer

Figure 12. Comparison of Compositing layer. Compositing means generated sample with the composting algorithm. None means generated samples without the compositing algorithm.

An alpha compositing algorithm was built in networks to enhance the edge appearance information for synthesizing vision data. In experiments, that algorithm made the classifier performance increasing. Firstly, figure 7 argues that alpha compositing algorithms cloud improves the edge appearance of the synthesizing final images. Secondly, figure 12 shown the comparison of using compositing layer or not. Thirdly, table 2 indicates that using alpha compositing algorithms cloud enhance the performance of the classifier slightly. When $\lambda = 0.5$ or 0.72, Figure 9 indicates the generator can produce available background. Using compositing algorithms can make the classifier get a higher precision score. However, when the background was not strong enough, alpha composting algorithms had not increase of the accuracy, because compared to the appearance of the background the influence of edge appearance on the classifier was little.

Table 1. Classification results of different training sets. Training sets were adding background with different methods, generated samples, and natural image: Large scale 2D-3D annotaion image database[26] (ObjNet_3D), Rendering images x_{syn} without background (Syn_nobkg), Rendering images x_{syn} with uniform noise background (Syn_uniform), Rendering images x_{syn} with random background from SUN database[31](Syn_SUN), Compositing generated background by Cycle-GANs(Cycle_GANs) and Compositing generated samples of 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%

Table 2. This figure indicates the effect of using alpha compositing algorithm. Where BAGANs-CL means BAGANs using alpha compositing algorithms (discussed in sec3.2.3). When λ = 0,0.25and 1, the generator produce "noise-like" background, and the score of classifier are not changed. When $\lambda = 0.5$ or 0.72, the training data generated by BAGANs-CL got better score compare with 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

As the compared methods, the classification score of ACGAN and Cycle-GAN argued that simple inputs (only random noise vector and image) would the performance of single source input (random noise vector or image) was worse than that of multiple input sources (random noise vector or image) in data synthesis tasks. And data synthesis benefited from reducing the "burden" of GAN.

Compared to different backgrounds images, the significant finding is that increasing the quality of backgrounds helped improve the performance of intelligent visual algorithms (Table 1).

In the classification score, BAGAN with alpha compositing algorithm achieved the best performance (accuracy 93.51%, precision 95.27%, recall score 97.02%, and f1-score 96.14%).

5. Discussion

This work study using 3D models and GAN produced visual data. To improve visual intelligence algorithms, providing massive guaranteed annotated data was an effective way. This conclusion has been proved by zhang[5], and proved by our classification results again. Our synthesized data provided massive labeled data for deep networks training and achieved the best performance which accuracy score is 93.51%, precision is 95.27%, recall score is 97.02%, and f1-score is

96.14%, compare with the manual labeled data, our synthetic images looks unnatural but the synthetic data help the classifier recognized objects in natural images.

Benifits for Augment Reality. The objects recognition algorithm of AR needs a lot of tag data for the business scene in the practical application process. In some special industrial scene, 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 relied on GAN and 3D models, which are easy to obtain in the relevant industrial chain, and a small number of models can generate a large number 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 can not 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 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 from real images is more comfortable to obtain than real photos. Therefore, our approach is to promote immersive interactive systems with good exploration.

Our works also Provided an alternative choice to fix GAN into the field of data synthesis is a significant strength of our works. However, the generated samples by BAGAN still were poor compared with the natural image.

Cycle-GAN performed well in other tasks, for instance, image to image conversation. The energy function should be considered to compositing in BAGAN of the follow-up research.

Major findings in this article are:

1. Using 3D shapes to decrease the complex task of GAN in data synthesis.
2. Designing BAGAN improves the synthesis pictures more natural.
3. Using alpha compositing algorithms to increase foreground edge appearance.
4. Training visual data produced by our method enhanced the classifier get 93% accuracy.

Considering the subsequent research of this article, five ways could promote our works in future:

- Wang [32] applied visual attention algorithms for video saliency detection. Adding the attention module may let GAN spend more consideration on the image background.
- Stacked stages fro networks (mentioned by Jaime[33]), manufacturing multi-stages networks perhaps an adequate approach to fix the problem of the high-resolution images.
- Adjusting the energy function to absorb Cycle-GAN image to image.
- Considering the classification results, saliency object recognition[34] perhaps an alternative research direction.

Acknowledgments: Thank Yu-Zhe Chang, Yi-jin Xiong and Qi Chen they have contributed a lot to this paper in sorting out the materials and literature research. Moreover, the author Yan Ma thanks Ruo-Ning Cao for her patience and understanding.

Author Contributions: Yan Ma proposed the idea of this paper; Kang Liu, Xu Qian, and Hong Bao reviewed this paper and gave many pieces of information; Yan Ma. and Zhi-bin Guan conceived and designed the experiments; Yan Ma and Xin-kai Xu performed the experiments; Xin-Kai Xu reviewed the codes in this paper. Yan Ma wrote this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:
HCI: Human-Computer Interaction.
GAN: Generative Adversarial Networks.
PCA: Principal Component Analysis.
ILSVRC: ImageNet Large Scale Visual Recognition Challenge.
ACGAN: Auxiliary Classifier Generative Adversarial Networks proposed by [14].
DCGAN: Deep Convolutional Generative Adversarial Networks proposed by [15].
CGAN or Conditional GAN: Conditional Generative Adversarial Nets proposed by [23].
Semi-GAN: Semi-Supervised Learning with Generative Adversarial Networks proposed by [24].
Info-GAN: Information Maximizing Generative Adversarial Nets proposed by [25].
ObjectNet3D: Large-Scale 2D-3D image dataset proposed by Stanford University [26].
ShapeNet: Large-Scale 3D shapes dataset proposed by Stanford University [27].
SUN Database: Large-Scale scene understanding dataset proposed by Princeton University [8].

References

1. Ma, Y.; Liu, K.; Guan, Z.b.; Xu, X.K.; Qian, X.; Bao, H. Using GAN to Augment the Synthesizing Images from 3D Models. Advances in Brain Inspired Cognitive Systems; Ren, J.; Hussain, A.; Zheng, J.; Liu, C.L.; Luo, B.; Zhao, H.; Zhao, X., Eds.; Springer International Publishing: Cham, 2018; pp. 96–105.
2. 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. IEEE International Conference on Wearable and Implantable Body Sensor Networks, 2015, pp. 1–6.
3. Hong, J.I. Considering privacy issues in the context of Google glass. *Communications of The ACM* **2013**, *56*, 10–11.
4. Evans, G.; Miller, J.; Pena, M.I.; Macallister, A.; Winer, E.H. Evaluating the Microsoft HoloLens through an augmented reality assembly application. *Proceedings of SPIE* **2017**, *10197*.
5. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Information Fusion* **2018**, *42*, 146–157.
6. Jain, N.; Kumar, S.; Kumar, A.; Shamsolmoali, P.; Zareapoor, M. Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters* **2018**.
7. Wang, Z.; Lu, D.; Zhang, D.; Sun, M.; Zhou, Y. Fake modern Chinese painting identification based on spectral–spatial feature fusion on hyperspectral image. *Multidimensional Systems and Signal Processing* **2016**, *27*, 1031–1044.
8. 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. Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015, pp. 626–630.
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017, pp. 2980–2988.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.
11. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *computer vision and pattern recognition* **2015**, pp. 1–9.
12. Tipping, M.E.; Bishop, C.M. Probabilistic Principal Component Analysis. *Journal of The Royal Statistical Society Series B-statistical Methodology* **1999**, *61*, 611–622.
13. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. International Conference on Neural Information Processing Systems, 2014, pp. 2672–2680.
14. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. *international conference on machine learning* **2016**, pp. 2642–2651.
15. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *international conference on learning representations* **2016**.

16. Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R. Learning from Simulated and Unsupervised Images through Adversarial Training. CVPR, 2017, Vol. 2, p. 5.
17. Zongker, D.E.; Werner, D.M.; Curless, B.; Salesin, D.H. Environment Matting and Compositing. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 1999; SIGGRAPH '99, pp. 205–214.
18. Porter, T.; Duff, T. Compositing Digital Images. *SIGGRAPH Comput. Graph.* **1984**, *18*, 253–259.
19. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss. Computer Vision (ICCV), 2017 IEEE International Conference on, 2017.
20. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* **2017**.
21. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875* **2017**.
22. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. Advances in Neural Information Processing Systems, 2017, pp. 5769–5779.
23. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *Computer Science* **2014**, pp. 2672–2680.
24. Odena, A. Semi-Supervised Learning with Generative Adversarial Networks. *arXiv: Machine Learning* **2016**.
25. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *neural information processing systems* **2016**, pp. 2172–2180.
26. 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. European Conference Computer Vision (ECCV), 2016.
27. 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. *Computer Science* **2015**.
28. Zhao, D.; Zheng, J.; Ren, J. Effective Removal of Artifacts from Views Synthesized using Depth Image Based Rendering. The International Conference on Distributed Multimedia Systems, 2015, pp. 65–71.
29. Smith, A.R.; Blinn, J.F. Blue Screen Matting. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques; ACM: New York, NY, USA, 1996; SIGGRAPH '96, pp. 259–268.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *international conference on learning representations* **2015**.
31. Xiao, J.; Hays, J.; Ehinger, K.A.; Oliva, A.; Torralba, A. SUN database: Large-scale scene recognition from abbey to zoo. Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.
32. 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**.
33. 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.
34. Han, J.; Zhang, D.; Hu, X.; Guo, L.; Ren, J.; Wu, F. Background Prior-Based Salient Object Detection via Deep Reconstruction Residual. *IEEE Transactions on Circuits & Systems for Video Technology* **2015**, *25*, 1309–1321.