Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06386-9

ORIGINAL ARTICLE

Multidomain image-to-image translation model based on hidden space sharing

Ding Yuxin[1] · Wang Longfei[1]

Received: 5 July 2020 / Accepted: 26 July 2021

Abstract
Image-to-image translation translates an image from one domain to another. The goal is to learn the translation relationship between different image domains. Compared with the translation models to be trained using paired training data, CycleGAN has the advantage of learning to translate between domains without paired input–output training examples. However, when using CycleGAN to translate images among multiple domains, the complexity of the model increases nonlinearly with the number of domains. To reduce the model complexity of CycleGAN-based translation models, we assume that there is a hidden space shared by different domains, and this space stores the common features of images. Then, we design a common encoder to learn image features in the hidden space. Based on the hidden space, we propose a translation model that scales linearly with the number of domains. To further improve the common feature representation accuracy, we introduce the adversarial component in the hidden space to learn the common features. We test the proposed models on different datasets, including painting style and season transfer datasets and achieve good results.

Keywords Image-to-image translation · Generative adversarial networks · Adversarial learning · Encoder

1 Introduction

Image translation refers to transferring the style of an image from one domain to another domain, such as changing a scene from spring to winter and changing the painting style of pictures. Analogous to automatic language translation, image-to-image translation can be defined as the problem of translating the representation of an image from the source domain into that of the target domain while still retaining the semantic content of the source image.

Generative adversarial networks (GANs) [1, 2] can avoid the difficulty of approximating many intractable probabilistic computations and have been effectively used to train generative models. At present, different GANs have been proposed to transfer one image style into another. Isola et al. [3] proposed conditional adversarial networks for image-to-image translation tasks. These networks not only learn the mapping from the input image to the output image but also learn a loss function to train this mapping. However, this model needs to be trained using paired training data (one object represented using two different styles), which are difficult to obtain. To solve this issue, CycleGAN is proposed. CycleGAN can be trained using unpaired data samples. The model couples with inverse mapping and introduces a cycle consistency loss to guarantee convergence.

Cycle GAN is only applied to translate images between two domains. If the model is extended to multiple domains, the model complexity increases nonlinearly. For example, if there are n domains, $(n^2 - n)$ generators (including decoders and encoders) and $(n^2 - n)$ discriminators are required. In this paper, we study how to reduce the computational complexity of multidomain image translation models. We make the following contributions.

(1) We assume there is a hidden common space shared by different image domains, and this space stores the common features of images. Based on this assumption, we propose a novel GAN model, named the generative adversarial networks based on hidden space sharing (HssGAN). HssGAN contains only one common encoder that is used to learn the features of

✉ Ding Yuxin
yxding@hit.edu.cn

[1] Computer Science Dept, Harbin Institute of Technology (Shenzhen), Shenzhen University Town, Shenzhen, China

Published online: 05 August 2021

a hidden space. Therefore, the model complexity of HssGAN increases linearly with the number of image domains.

(2) To further improve the representation power of the common encoder, we add a generative adversarial component in an HssGAN and use adversarial training to learn the features of the hidden space.

2 Related work

The goal of the image translation task is to translate an image a from domain A to image b in another domain B. The following methods have been proposed for image translation.

2.1 Methods separating the content from the image style

References [4, 5] used a convolutional neural network to learn image representations. The neural algorithm of artistic style was proposed to separate the image content from its style. The algorithm can mix the content and style independently to synthesize new, perceptually meaningful images.

Ulyanov et al. [6] proposed an alternative approach that can reduce the computational complexity in the learning stage. Given a single example of a texture, a convolutional neural network is trained to generate different samples of the same texture of arbitrary size and synthesize the artistic style from a given image with any other image.

Johnson et al. [22] combined feed-forward image transformation networks with optimization-based methods for image generation and used perceptual loss functions to train convolutional neural networks. The experimental results on image style translation show that compared to the optimization-based method, the proposed network can generate images of similar quality, but the speed is three orders of magnitude faster.

Long et al. [7] adapted several classic classification networks into fully convolutional networks and transferred the learned representations by fine-tuning to the segmentation task. They proposed a skip structure that can fuse both the semantic information from a deep layer and the appearance information from a shallow layer to generate detailed and accurate segmentations. Xu et al. [8] used a fully convolutional generator based on two subnetworks to implement image-to-image translation. The first subnetwork generates the outline of an image in a new domain, and the second subnetwork translates the outline to a visually realistic image. Karatsiolis et al. [9] proposed a hierarchical structure that can encapsulate the information of the target domain using a separately trained network. This hierarchical structure is then trained into a unified depth network for image translation.

Convolutional neural network-based translation models are trained to minimize a loss function. Therefore, one key problem is how to design an effective loss function. In previous papers [10, 11], the authors proposed that the loss function designed to minimize the Euclidean distance between two domains easily generates ambiguous images.

2.2 Image-based methods

These kinds of methods use image processing methods to translate images. Efroset et al. [5] proposed the image quilting algorithm to synthesize image texture and extended the algorithm to perform texture transfer by replacing the texture of an image with the texture from a different image. Hertzmann et al. [12] proposed a framework for generalizing texture synthesis for the case of two corresponding image pairs. They used image pairs (one image is a "filtered" version of the other) to train artistic filters and then used the filters to change the painting style of an image.

2.3 GAN-based methods

Liu et al. [16] proposed a coupled generative adversarial network (CoGAN) for learning a joint distribution of multidomain images. CoGAN can learn a joint distribution from samples selected from the marginal distributions. The joint distribution is calculated by implementing a weight-sharing constraint that limits the network capacity. Taigman et al. [26] proposed the domain transfer network to translate images. The network employs a compound loss function that includes a multi-class GAN loss, an f-constancy component and a regular component that encourages the generator to map images from one domain to another.

Isola et al. [3] proposed conditional adversarial networks to solve image-to-image translation problems. The model can simultaneously learn the mapping from the input image to the output image and a loss function to train this mapping. This approach makes it possible to use the same generic method to problems that have different loss functions. Zhu et al. [2] proposed CycleGAN to solve this problem. In this model, the cycle consistency loss is designed to train a generative adversarial network. Unlike the pix2pix method [3, 5], CycleGAN does not require paired data for training. StarGAN [13] and SemiStarGAN [14] are CycleGAN-based image-to-image translation models, which can implement multidomain image translation. There is only one shared generator in these models. In fact, it is very difficult to train a general generator which

can generate images belonging to different domains. Anoosheh et al. [15] pointed out that these models share one generator, and they are only suitable for translating images between image domains that are sufficiently similar to each other. Currently, StarGAN and SemiStarGAN were only applied to the task of face attribute modification, where all the domains were slight shifts in qualities of the same category of images: human faces.

CycleGAN translates images only between two image domains. To translate images among n domains, we need to train $O(n^2)$ models. In this paper, we propose a novel GAN structure based on CycleGAN. Similar to CycleGAN, the proposed model also uses the cycle consistency loss to learn the mapping relations between different domains. Unlike CycleGAN, the proposed model can translate images among multiple domains; the generator in our model is divided into two parts, the encoder and decoder; and the encoder is shared by all the domains. Therefore, for n domains, we need to train only $O(n)$ models.

3 Image translation model based on hidden space sharing

3.1 Problems

Image-to-image translation learns the mapping between an input image and an output image [16]. Usually, a training set of aligned image pairs is required to train an image translation model. In practice, it is difficult to collect a paired training set. To solve this issue, CycleGAN [2] is proposed. CycleGAN includes two adversarial models: one model contains a generator G_{A2B} and a discriminator Dis_B (see Fig. 1).G_{A2B} is trained to translate an image from domain A to an image in domain B, and Dis_B is trained to discriminate between real images in domain B and images translated through G_{A2B}. In this model, an input image I_A from domain A is translated into an image I_{A2B} ($I_{A2B} = G_{A2B}(I_A)$) of domain B through G_{A2B}, and then the discriminator Dis_B of domain B determines whether the translated image satisfies the characteristics of domain B. To learn from unpaired examples, an inverse translation is introduced to force $I_{\mathrm{Cycle\_}A} \approx I_A$ ($I_{\mathrm{Cycle\_}A} = G_{B2A}(G_{A2B}(I_A))$), and vice versa. This process is implemented by introducing a cycle consistency loss $L_{\mathrm{cycle}} = \|G_{B2A}(G_{A2B}(I_A)) - I_A\|_1$. The other adversarial model adopts the same strategy to translate an image from domain B into that of domain A,we do not discuss this strategy in detail here.

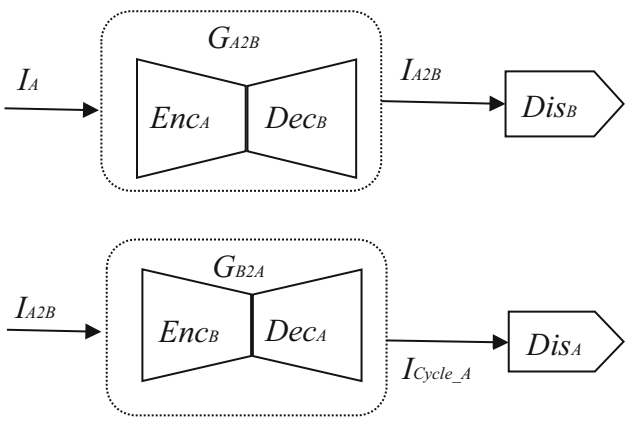

Fig. 1 The structure of a CycleGAN model

In CycleGAN, the generator for each domain includes two parts, namely, an encoder and a decoder, and the two parts are closely coupled. For example, G_{A2B} consists of an encoder and a decoder, and the two parts work together to translate an image from A to B. The close coupling between the encoder and decoder makes it difficult for CycleGAN to translate images among multiple domains. For example, if we add a new image domain C for translation, we need to create six generators, G_{A2B}, G_{A2C}, G_{B2A}, G_{B2C}, G_{C2A} and G_{C2B}. If there are N image domains, $n(n-1)$ generators would be trained. We can see that the model complexity increases nonlinearly with the increase in image domains. It is time consuming to use CycleGAN to translate images among multiple domains. Therefore, the motivation of our research is how to design an efficient model for translating images among multiple domains.

3.2 GAN based on hidden space sharing (HssGAN)

In an adversarial network, the generator includes two parts, an encoder and a decoder. The encoder converts an image from domain A into a feature representation in a feature space, and the decoder translates the feature representation into an image in domain B. To correctly translate an image from domain A to domain B, the feature representation needs to contain two kinds of features: common features not specific to a certain domain and the style features specific to domain B. For example, a house can be painted in different styles, but all houses have common features, such as doors and windows. In addition, each house also has its own style features. Based on this assumption, we assume there is a hidden space called the common feature space, and all the images from different domains can be translated into a feature representation (common feature representation) in this space. Thus, we separate the encoder and the decoder of the generator. The encoder is responsible for generating the common feature representation of an image, and the decoder is responsible for translating a

common feature vector into an image and adding style features into the image at the same time.

The proposed translation model (HssGAN) is shown in Fig. 2. For the convenience of comparison, Fig. 2 gives the structures of HssGAN and CycleGAN for translation among three image domains. In the HssGAN model, the generator contains one common encoder Enc and the decoders for each image domain, and all the decoders share one encoder. The common encoder extracts the common features of an image and then sends the feature vector to a decoder to generate an image. Dec_i is the decoder for domain i; it decodes a common feature representation into an image of domain i. Dis_i is the discriminator corresponding to Dec_i; it is responsible for deciding whether the image generated by Dec_i satisfies the characteristics of domain i. In Fig. 2, the graph on the right gives the structure of the CycleGAN. In this structure, *Gi2j* represents a generator that can translate an image from domain i to domain j. In CycleGAN, the generator contains one encoder and one decoder, and the two parts work together and cannot be separated from each other. For HssGAN, an image from any domain is input into the common encoder, and the decoders translate the image into images of other domains. For example, image *I1*from domain one is the input data. The common encoder encodes *I1* into a feature vector Enc(*I1*), and then Enc(*I1*)is sent to decoders Dec_2 and Dec_3 to generate images for domain two and domain three. For CycleGAN, the generator only translates an image from one domain to another. Therefore, we need to design a generator for any domain pair. In Fig. 2, if a new domain, such as domain three, is added, for the HssGAN, we need to add only one decoder Dec_3 and one discriminator Dis_3, while for the CycleGAN, we need to design four generators and four discriminators, which are *G123*, *G223*, *G321*, *G322*, Dis*123*, Dis*223*, Dis*321* and Dis*322*. Compared with CycleGAN, the model complexity of HssGAN increases linearly with the number of image domains.

In the HssGAN, the encoder, decoder and discriminator are trained simultaneously. We construct the adversarial loss and the cycle consistency loss to train them. Suppose the training set is $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where x_i is an image from domain y_i. For convenience, we use (x, y) to represent a training sample in D. Our goal is to learn the parameters of the common encoder and the decoders for each domain. With the learned encoder and decoders, we can translate an input image x from any domain to the image of the target domain T (output of Dec_T).

$$
\begin{aligned}
&L_{\text{GAN}}(\theta_{\text{Enc}}, \theta_{\text{Dec}_T}, \theta_{\text{Dis}_T}, T) \\
&\quad = \sum_{(x,y)\in D,\, y=T} \left[(\text{Dis}_T(x) - 1)^2\right] \\
&\qquad + \sum_{(x,y)\in D,\, y\neq T} \left[\text{Dis}_T(\text{Dec}_T(\text{Enc}(x)))^2\right]
\end{aligned}
$$

The adversarial loss for domain T is defined by Eq. (1). For domain T, we need to train the generator Enc-Dec_T (Enc is the common encoder) and the discriminator Dis_T. In Eq. (1), θ_{Dis_T} denotes the parameters of the discriminator Dis_T, which discriminates whether an image is a real image from domain T or a generated image in domain T. θ_{Dec_T}

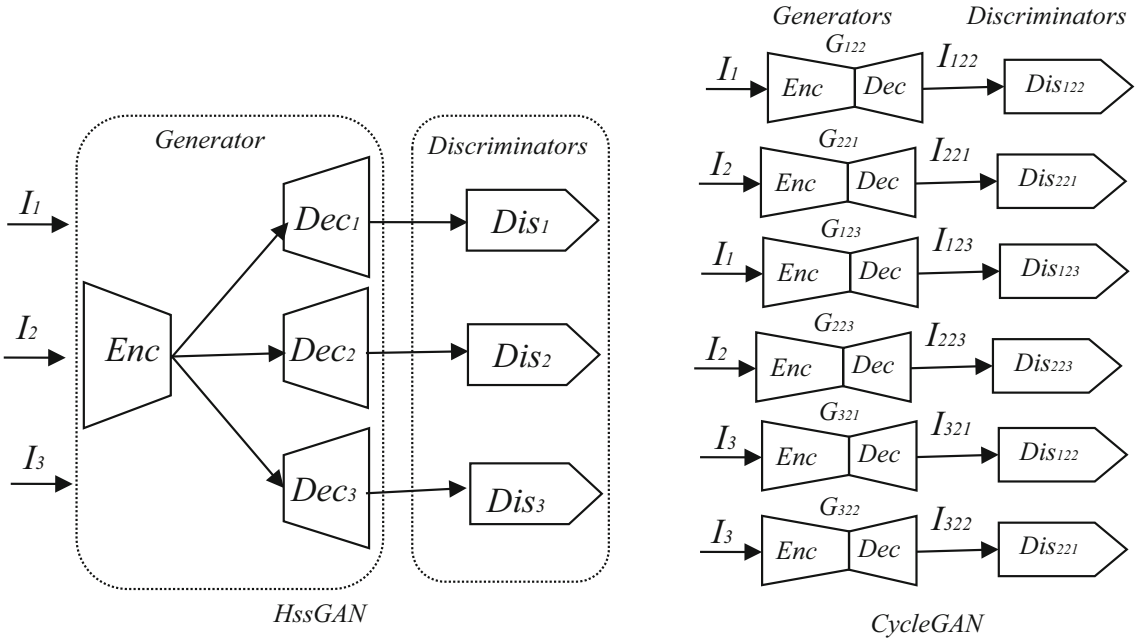

Fig. 2 Image translation model based on hidden space sharing (HssGAN)

denotes the parameters of the decoder Dec_T, which generates an image in domain T. θ_{Enc} denotes the parameters of the common encoder Enc. $L_{\mathrm{GAN}}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}, \theta_{\mathrm{Dis}_T}, T)$ consists of two terms. The first term is the loss generated by Dis_T when determining whether x ($x \in T$) is a real image from T, and the second term is the loss generated by Dis_T when determining whether $\mathrm{Dec}_T(\mathrm{Enc}(x))$ ($x \notin T$) is a translated image from domain T. We train Dis_T to minimize $L_{\mathrm{GAN}}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}, \theta_{\mathrm{Dis}_T}, T)$; at the same time, we train the generator (Enc-Dec_T) to maximize the second term. In the same way, the adversarial loss for other domains can be constructed.

Similar to CycleGAN, we also use the cycle consistency loss to regularize the highly unconstrained problem of translating an image in only one direction. The cycle consistency loss considers the mappings from one domain to another to be inverses of each other such that $\mathrm{Dec}_y(\mathrm{Enc}(\mathrm{Dec}_T)(\mathrm{Enc}(x))) \approx x$ ($y \neq T$). In Eq. (2), we use the 1-norm to define the distance between two images.

$$L_{\mathrm{cycle}}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}, T) = \sum_{(x,y)\in D,\, y\neq T} ||\mathrm{Dec}_y(\mathrm{Enc}(\mathrm{Dec}_T(\mathrm{Enc}(x)))) - x||_1 \quad (2)$$

The optimization objective function for encoder Enc and decoder Dec_T is shown in Eq. (3). The optimization objective function consists of two parts: a GAN loss and a cycle consistency loss. The encoder Enc, decoder Dec_T and discriminator Dis_T are trained by minimizing the optimization objective function. In the same way, we can construct the optimization objective function for other domains.

$$L(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}|\theta_{\mathrm{Dis}_T}, T) = L_{\mathrm{cycle}}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}, T) - L_{\mathrm{GAN}}(\theta_{\mathrm{Enc}}, \theta_{\mathrm{Dec}_T}, \theta_{\mathrm{Dis}_T}, T) \quad (3)$$

The flow chart of HssGAN is shown in Fig. 3. In Fig. 3, we use a HssGAN that can translate images between two domains to illustrate how the proposed model works. First, we construct the GAN loss L_{GAN} and use it to train the model (see the labels marked as 1 in Fig. 3). In Fig. 3, a real image I_1 from domain one is sent to encoder Enc, and Enc outputs a feature representation for I_1, denoted by $\mathrm{Enc}(I_1)$. $\mathrm{Enc}(I_1)$ is sent to decoder Dec_1, and Dec_1 generates an image for domain one, denoted by $\mathrm{Dec}_1(\mathrm{Enc}(I_1))$. The discriminator Dis_1 decides whether $\mathrm{Dec}_1(\mathrm{Enc}(I_1))$ is a real image from domain one or a generated image for domain one. If Dis_1 makes a wrong decision, the GAN loss L_{GAN} is greater than zero, thus we can use L_{GAN} to train Dis_1 by minimizing L_{GAN} and train Dec_1 and Enc by maxmizing L_{GAN}.

Second, we construct the cycle consistency loss L_{cycle} and use it to train the model (see the labels marked as 2 in Fig. 3). The cycle consistency loss for image I_1 is constructed as follows. I_1 is sent to encoder Enc and decoder Dec_2 to generate an image for domain two. The translated image is denoted by $\mathrm{Dec}_2(\mathrm{Enc}(I_1))$. Then, the translated image is sent to Enc and Dec_1 to reconstruct image I_1, denoted by Cyc_I_1. We can calculate the cycle consistency loss L_{cycle} from I_1 and Cyc_I_1, and use L_{cycle} to train Enc and Dec_1 by minimizing the loss.

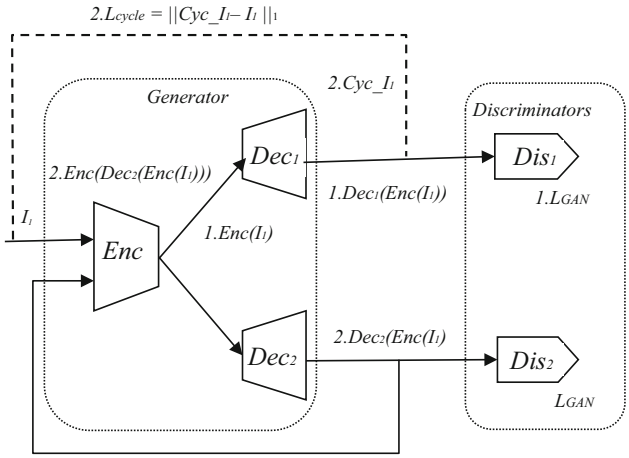

Fig. 3 Flow chart of HssGAN

In our study, we implement HssGAN and test its performance using different datasets (see Sect. 4). Compared with the images generated by CycleGAN (see Fig. 9), the textures of the images generated by HssGAN are more reasonable (see Fig. 7 and the analysis in Sect. 4.2.2). However, the images translated by HssGAN lose some obvious features of the translated domain. Sometimes the images translated by HssGAN show that blended features exist in different domains. For example, in Fig. 7, the third row shows the images that are translated from the real image from the autumn domain. We can see that all the translated images have some features belonging to the winter domain; for example, some trees in all the translated images are covered with snow. This phenomenon is unlikely to be caused by the decoders because the probability that all three decoders commit the same error is very low. One possible reason is that the feature vector generated by the encoder cannot accurately represent the features of different domains. Especially with the increase in the number of domains, it becomes very difficult for HssGAN to train a common encoder.

To improve the feature representation of the encoder, we introduce adversarial learning in the hidden space, which is shown in Fig. 4. For convenience, we call this translation model HssGAN-HAL. To enrich the feature representation contained in Enc(x), a discriminator for the hidden space, Latent_Dis, is added. Latent_D is a multi-class classifier (a softmax layer) that outputs the probability $P(y|\mathrm{Enc}(x))$ that

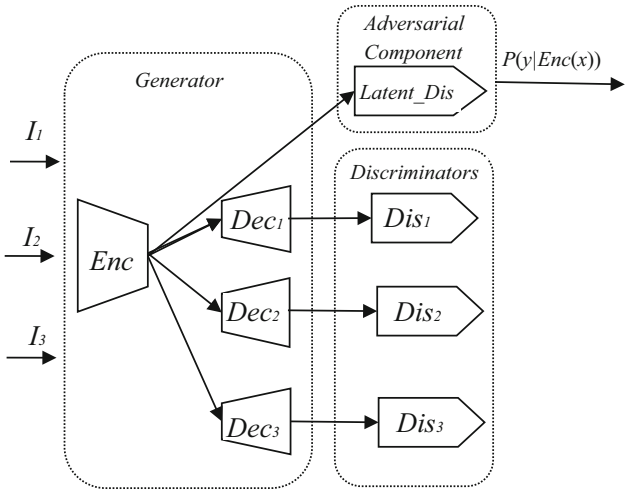

Fig. 4 HssGAN with adversarial learning in the hidden space (HssGAN-HAL)

Enc*(x)* belongs to domain y. If x is a real image from domain y, Latent_Dis will try its best to classify Enc*(x)* as y; that is, Latent_Dis maximizes $P(y|\mathrm{Enc}(x))$. Latent_Dis forces Enc*(x)* to learn more information so that it can better represent the characteristics of each domain. The adversarial loss for the hidden space is defined in Eq. (4). We use Fig. 4 to interpret the working flow of the adversarial learning module. In Figs. 4, I1, I2, I3 are the inputs of the common encoder. Different from other models, the encoder in HssGAN is a common encoder. So, the input of the common encoder can be an image from any image domain. For example, image I_1 from domain 1 is randomly selected as the input data of encoder Enc. Enc outputs a feature vector Enc(I_1) and sends it to Latent_Dis for classification. Then, the classifier and Enc are trained by maximizing the adversarial loss defined in Eq. (4). This training procedure can make the feature vector Enc*(x)* contain more characteristics specific to domain one. In fact, Latent_Dis is a multi-class classifier. The goal of training the multi-class classifier is to improve the performance of the common encoder. So, the common encoder can learn a better feature representation for the input image and then make the decoder of each image domain generate a high quality image. If the generated high quality image can deceive the discriminator of the target domain, the performance of the discriminator can be further improve by learning from the high quality image. From this view, we regard the multi-class classifier as an adversarial component.

Through adversarial training, it can be considered that the feature representation output by the encoder contains two kinds of information: a common representation and the class information. This assumption is equivalent to adding a supervised signal y into the feature representation generated by the encoder. For convenience, the feature representation is described as $< z,\ y >$, where z is the common representation of x, and y is the class label of x. When training Dec_i using $< z,\ y >$, the supervised signal can guide the decoder to generate images more consistent with the characteristics of the target domain.

$$L_{\mathrm{GAN\_Latent}}(\theta_{\mathrm{Latent\_Dis}},\ \theta_{\mathrm{Enc}}) = \sum_{(x,\ y)\in D} \log P(y|\mathrm{Enc}(x))$$

After adding the adversarial learning component, the optimization objective function for the encoder and decoder in domain T is defined in Eq. (5).

$$\begin{aligned} &L(\theta_{\mathrm{Enc}},\ \theta_{\mathrm{Dec}_T}|\theta_{\mathrm{Dis}_T},\ \theta_{\mathrm{Latent\_Dis}},\ T) \\ &= -L_{\mathrm{GAN}}(\theta_{\mathrm{Enc}},\ \theta_{\mathrm{Dec}_T},\ \theta_{\mathrm{Dis}_T},\ T) \\ &+ L_{\mathrm{cycle}}(\theta_{\mathrm{Enc}},\ \theta_{\mathrm{Dec}_T},\ T) + L_{\mathrm{GAN\_Latent}}(\theta_{\mathrm{Latent\_Dis}},\ \theta_{\mathrm{Enc}}) \end{aligned} \tag{5}$$

In HssGAN, the encoder and decoder adopt the network structure proposed by Johnson et al. (2016), and the discriminator adopts PatchGAN [23]. The detailed structure of the encoder, decoder and discriminator is shown in Table 1. In Table 1, (Nx, Kx, Sx) describes the parameters for each layer; for example, (N64, K7, S1) means that the number of neurons is 64, the kernel size is 7×7, and the stride is 1.In the encoder, all the convolutional layers and residual blocks are followed by instance normalization and ReLU nonlinearities. The neurons in the discriminator use LeakyReLU as the activation function.

Table 1 Network structure of the HssGAN

Layer	Encoders
1	CONV—(N64, K7, S1), InsNorm, ReLU
2	CONV—(N128, K3, S2), InsNorm, ReLU
3	CONV—(N256, K3, S2), InsNorm, ReLU
4–7	RESBLK—(N256, K3, S1), InsNorm, ReLU
Layer	Decoders
1–5	RESBLK—(N256, K3, S1), InsNorm, ReLU
6	DCONV—(N128, K4, S2), InsNorm, ReLU
7	DCONV—(N64, K4, S2), InsNorm, ReLU
8	CONV—(N3, K7, S1), Tanh
Layer	Discriminators
1	CONV-(N64, K4, S2), LkReLU
2	CONV—(N128, K4, S2), InsNorm, LkReLU
3	CONV—(N256, K4, S2), InsNorm, LkReLU
4	CONV—(N512, K4, S1), InsNorm, LkReLU
5	CONV—(N1, K4, S1)

Fig. 5 Samples generated by HssGAN with different loss functions

Table 2 Reconstruction errors for the translation models on the paired dataset

Reconstruction	GAN	HssGAN	HssGAN-HAL
pho.- > map- > pho	281	103	**97**

Table 3 MTurk "real vs fake" evaluation on the paired dataset

	Map- > Photo(%)	Photo- > Map(%)
Loss	Turkers labeled real	Turkers labeled real
CoGAN	0.6	0.9
BiGAN/ALI	2.1	1.9
SimGAN	0.7	2.6
Cycle GAN	**26.8**	**23.2**
HssGAN	24	22
HssGAN-HAL	24	22

4 Experiments

4.1 Datasets and evaluation metrics

We test the performance of the proposed model using three datasets. The first dataset is a collection of maps and aerial photos from Google Maps. We select 1096 images as the training data and 100 images as the testing data.

The second dataset consists of photos of the Alps from Flickr. The photos are classified into four seasons. The training dataset includes 6000 photos, and the testing dataset includes 1000 photos.

The third dataset is a collection of approximately 10,000 paintings from 14 different artists from Wikiart.org. We randomly select 6000 paintings from 8 artists. Among them, 5000 paintings are used for training, and 1000 paintings are used for testing.

It is difficult to evaluate the quality of translated images [17]. We assess the quality of images from both subjective and objective perspectives.

Amazon Mechanical Turk (MTurk) [11]: translated images are provided to Turkers, and the Turkers vote on which image is more reasonable, the more votes there are, the better the image quality.

Reconstruction loss: An image a from domain A is translated into image c_i in different domains from A, and then each image c_i is converted back to domain A (denoted by $Cycle\_a_i$). We calculate the L_1 distance between a and each $Cycle\_a_i$; the smaller the reconstruction error is, the better the translation model.

The experimental environment is a computer equipped with an Intel Xeon(R) CPU (E5-1650 v4 @ 3.60 GHz) and an Nvidia GTx1080 GPU with 64 GB memory. The deep learning framework is TensorFlow. During the training phase, we randomly select a number of training images and calculate the reconstruction error of the translation model. We stop training when the reconstruction error of the model becomes stable.

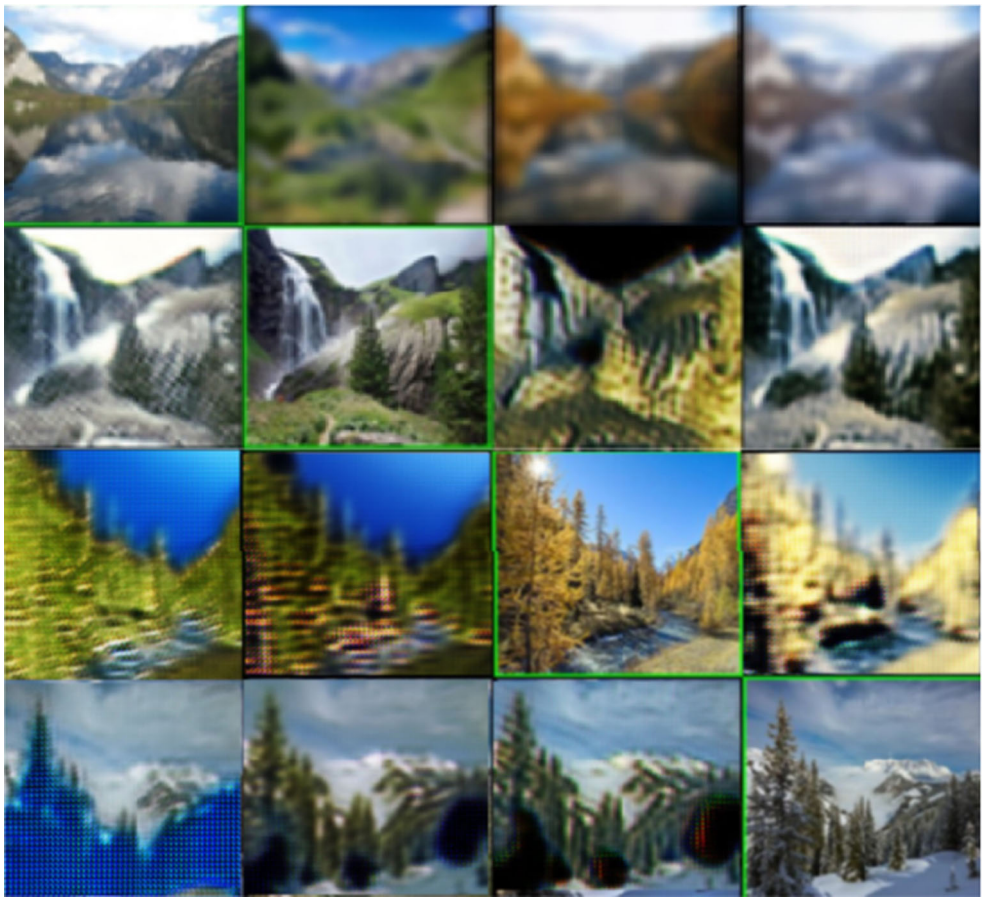

Fig. 6 Images translated using the GAN model

4.2 Experimental results and analysis

4.2.1 Experiments on a paired dataset

Experiment 1(Ablation Study): We select the first dataset as the experimental data. This dataset is a paired dataset that provides maps and the corresponding aerial photos. To evaluate the performance of the proposed models, we performed ablation experiments with different loss functions on the paired dataset. The first loss function only includes the GAN loss L_{GAN} (the first term of Eq. (5)). The second loss function includes the GAN loss L_{GAN} and the cycle consistency loss L_{cycle} (the second term of Eq. (5)). The third loss function includes L_{GAN}, L_{cycle} and the adversarial loss for the hidden space $L_{\text{GAN\_Latent}}$ (see Eq. (4)). The translation models trained using these three loss functions are denoted by GAN, HssGAN, HssGAN-HAL, respectively. Figure 5 shows several samples generated by the generators trained using the different loss functions.

From Fig. 5, we can see that compared with the images in the second column, the images in the third column and fourth column have better quality. This finding shows that the cycle consistency loss and adversarial loss for the hidden space are useful for improving the performance of the generators. For the paired dataset, we cannot see significant differences between the images in the third column and fourth column. In this experiment, there are only two image domains, and the dataset is a paired dataset; therefore, the adversarial loss for the hidden space becomes unimportant, and the encoder in HssGAN can easily learn better representations of the input data using only the cycle consistency loss.

To objectively evaluate the performance of HssGAN and HssGAN-HAL, we reconstruct the original images from the translated images and use the reconstruction error to evaluate the performance of the translation models. We select a testing image x from one domain and translate it into image y in another domain and then use y as the input to reconstruct x. The recovered image is denoted by z. The reconstruction error is defined as the number of different pixels between x and z. The reconstruction errors of the different translation models are shown in Table 2. From Table 2, we can see that the reconstruction errors of the HssGAN and HssGAN-HAL models are much smaller than that of the GAN model, and the construction error of the

Fig. 7 Images translated using the HssGAN model

HssGAN-HAL model is slightly lower than that of the HssGAN model. This finding shows that the adversarial component can push the encoder to learn a better representation of the input data and improve the performance of the HssGAN. In Tables 2, 3, 4, 5, 6, 7, and 8, the best experimental results are shown in bold.

Experiment 2 (Performance comparison of different translation models): We compare the proposed models against several translation models, CoGan (Liu et al., [16]), BiGan/ALI [18, 19], SimGAN (Sharivastava et al., 2016) and CycleGAN. Except for CycleGAN, all the other algorithms need to be trained using a paired dataset. We invited 50 participants to make the MTurk evaluation on the translation results. Table 3 gives the performance on the MTurk perceptual realism task (the performance of other models is cited from the paper by Zhu et al. [2]). The ablation study shows that for a paired dataset, we can obtain high quality images using only the cycle consistency loss. Table 3 shows similar results. In Table 3, HssGAN and HssGAN-HAL have the same performance, which can fool 24 percent of the participants in the map → photo direction and 22 percent of the participants in the photo → map direction. Like CycleGAN, HssGAN and HssGAN-HAL all use the cycle consistency loss to constrain the mapping between different domains, so their performance is comparable to that of CycleGAN and better than other generative adversarial network models.

4.2.2 Experiments on an unpaired dataset

Experiment 1(Ablation Study): We select the second dataset as the experimental data. This dataset is an unpaired dataset that includes photos of the Alps over four seasons. The images in each season represent an image domain, so we use the proposed method to train translation models that can translate images among four domains. We first perform ablation experiments to evaluate the performance of the proposed models. The loss functions are the same as those used in Sect. 4.2.1.

Fig. 8 Images translated using the HssGAN-HAL model

Some image samples translated by the GAN, HssGAN and HssGAN-HAL models are shown in Figs. 6, 7 and 8, respectively. In each figure, the images on the diagonal are the ground-truth images. From left to right, the images on the diagonal come from the spring domain, summer domain, autumn domain and winter domain. Each image on the diagonal is converted into images in the other domains, and these converted images are shown in each row. The images in each row from left to right represent images in the spring domain, summer domain, autumn domain and winter domain, respectively.

From the images in Fig. 6, we can see that using only the GAN loss, the translation model fails to generate images close to the target domain. Compared with the ground-truth images on the diagonal, all the generated images are fuzzy and unreasonable.

Compared with the quality of the images generated by the GAN model, the quality of the images generated by the HssGAN model (see Fig. 7) is significantly improved, and the styles of the images are similar to their corresponding target domains because the cycle consistency loss provides a supervised signal for image translation between different domains and can avoid generating images that contradict each other.

As shown in Figs. 7 and 8, HssGAN-HAL and HssGAN generate images of similar quality. However, the images generated by HssGAN-HAL are more reasonable than the images translated by HssGAN. The translated images in Fig. 9 show more obvious seasonal characteristics; for example, in the second row, the mountain in the winter image is covered by snow (the mountain in the corresponding image generated by HssGAN is green); in the third row, the snow on the trees in the spring and summer images disappears (there is snow on the trees in the corresponding images generated by HssGAN), and in the fourth row, the trees in the spring and summer images are green (the trees in the corresponding images generated by HssGAN are green mixed with yellow). In HssGAN-HAL, the adversarial learning component is a supervised classifier; it knows the domain label of each training data.

Fig. 9 Images translated using the CycleGAN model [2]

Table 4 MTurk "real vs fake" evaluation on the GAN, HssGAN, and HssGAN-HAL models

Seasons	GAN(%) Turkers labeled real	HssGAN(%) Turkers labeled real	HssGAN-HAL(%) Turkers labeled real
sp- > su	2	20	**26**
su- > sp	0	20	**26**
sp- > au	2	34	**58**
au- > sp	0	20	**20**
sp- > wi	0	30	**40**
wi- > sp	4	34	**36**
su- > au	0	20	**40**
au- > su	6	**46**	40
su- > wi	0	30	**66**
wi- > su	0	36	**40**
au- > wi	2	40	**52**
wi- > au	2	44	**54**
Avg	1.7	31.2	41.5

Therefore, the adversarial component in the hidden space can provide a supervised signal for the encoder and guide the encoder to learn more meaningful feature representations of different domains.

Table 5 Reconstruction error of the GAN, HssGAN and HssGAN-HAL models

	GAN	HssGAN	HssGAN-HAL
Seasons	Recon. Loss	Recon. Loss	Recon. Loss
sp- > su- > sp	229	129	**127**
sp- > au- > sp	243	137	**134**
sp- > win- > sp	239	139	**109**
su- > sp- > su	323	125	**120**
su- > au- > sur	249	123	**110**
sur- > wi- > su	272	134	**127**
au- > sp- > au	280	115	**110**
au- > su- > au	301	**61**	70
au- > wi- > au	256	142	**138**
wi- > sp- > wi	272	122	**117**
wi- > su- > wi	305	120	**105**
wi- > au- > wi	349	144	**117**
Avg. err	276	124	**115**
Training time	24 h	28 h	30 h

We invited 50 participants to make the MTurk evaluation on the translation results. Table 4 gives the performance on the MTurk perceptual realism task. In Table 4, sp, su, au and wi are abbreviations for spring, summer, autumn and winter, respectively. Table 4 shows that the performance of the HssGAN and HssGAN-HAL models is significantly better than that of the GAN model, and the performance of the HssGAN-HAL model is significantly better than that of the HssGAN model, which can be approximately 10 percent higher than that of the HssGAN model. This finding shows that both the cycle consistency loss and the adversarial loss for hidden space are critical to our results.

We also calculate the reconstruction errors of the three models, which are shown in Table 5. We obtain a similar result: the HssGAN-HAL model has the smallest reconstruction error, and the reconstruction errors of the HssGAN and HssGAN-HAL models are significantly lower than that of the GAN model.

Experiment 2 (Performance comparison of different translation models): We compared the proposed models with CycleGAN and ComboGAN [15]. For CycleGAN, we need to train 12 generators and 12 discriminators. In ComboGAN, the encoder and decoder in the generator are separate, and the output of an encoder can be sent to different decoders. Unlike HssGAN, in ComboGAN one encoder and one decoder are designed for each domain. Therefore, we need to train four encoders and four decoders in a ComboGAN model. For HssGAN and HssGAN-HAL, we only need to train one common encoder and four decoders. In our experiments, the encoders and decoders in the different models have the same network structures, which are shown in Table 1.

Some image samples translated by CycleGAN and ComboGAN are shown in Fig. 9 and Fig. 10, respectively. Compared with HssGAN, CycleGAN and ComboGAN can generate images with obvious seasonal features. For example, in Fig. 7, the images in the first and second columns generated by HssGAN are very similar, while the images in the first and second columns in Figs. 10 and 11 have distinct seasonal features. In CycleGAN and ComboGAN, each domain has an encoder. It is easy for the encoder to learn accurate feature representations of images from only one domain. However, in HssGAN, all domains share one encoder; therefore, it is difficult for the common encoder to learn accurate feature representations from different domains. This is also the reason why we add the adversarial component to HssGAN. When adding the adversarial component, the image quality generated by HssGAN-HAL (see Fig. 8) is comparable to that generated by CycleGAN and ComboGAN.

Although the images translated by CycleGAN and ComboGAN show more seasonal features, these images also contain some obvious translation errors. For example, in the images in the second row in Fig. 9, the waterfall becomes blue, some parts of the mountain become black, and in the images in the third row, some trees become black. In the images in the second row in Fig. 10, the waterfall and parts of the mountain become black. The summer image in the fourth row in Fig. 10 shows snow-covered trees. Compared with the images translated by CycleGAN and ComboGAN, there are no obvious translation errors in our images. The common encoders in HssGAN and HssGAN-HAL are trained using images from different domains; therefore, they can learn more domain knowledge, which can guide the respective encoders to generate more correct encodings for objects.

The same MTurk "real vs fake" test is performed to assess the performance of the different translation models. Table 6 gives the performance on the MTurk perceptual realism task. Table 6 also shows that the performance of HssGAN-HAL is comparable to that of CycleGAN and ComboGAN. CycleGAN has the best performance and slightly outperforms HssGAN-HAL and ComboGAN. In our experiments, there are four image domains. We need to train 12 generators for CycleGAN and six decoders and six encoders for ComboGAN. For HssGAN-HAL, we only need to train one encoder and six decoders. From Table 6, we can see that the training cost of HssGAN-HAL is significantly lower than those of CycleGAN and ComboGAN.

We also calculate the reconstruction errors of the different translation models, which are shown in Table 7. Table 7 shows similar results as those of Table 6. HssGAN,

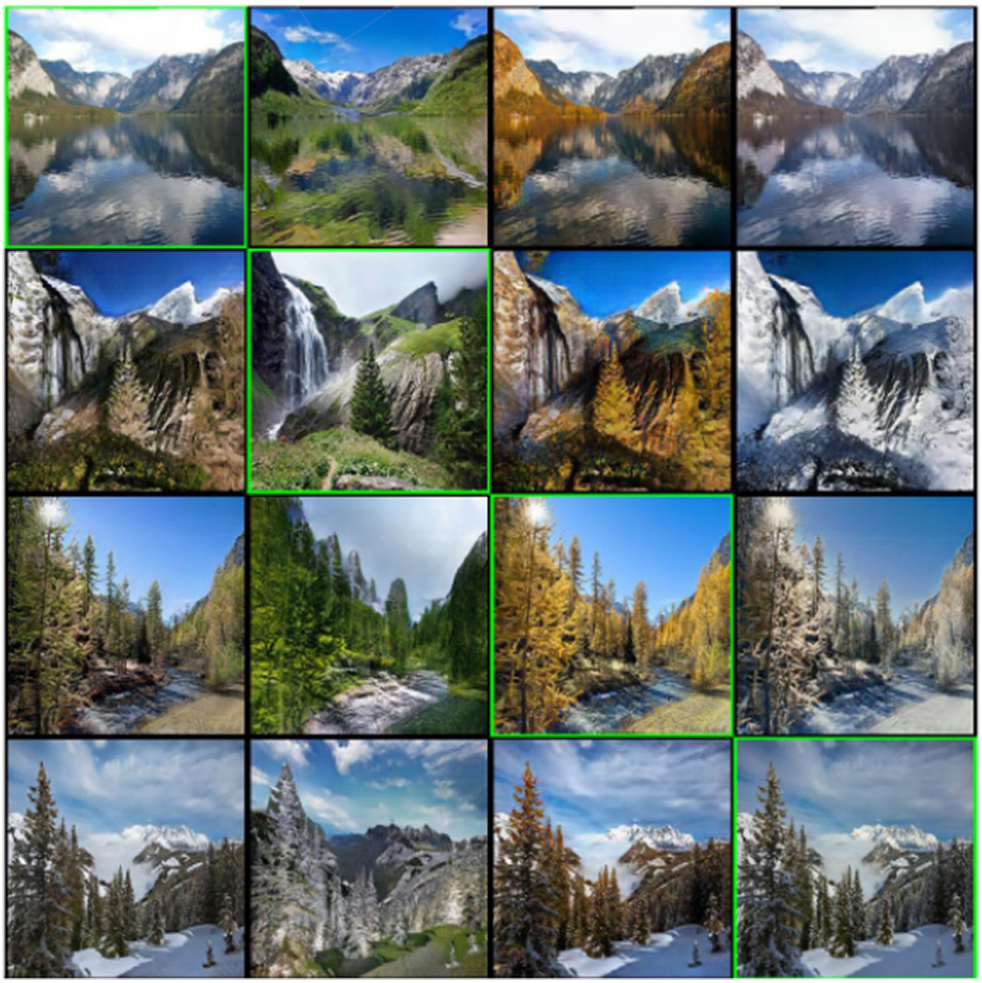

Fig. 10 Images translated using the ComboGAN model [15]

CycleGAN and ComboGAN output similar reconstruction errors. Among them, CycleGAN has the lowest reconstruction error.

Experiment 3: Besides MTurk and reconstruction loss, we also use Kernel Inception Distance (KID) (Mikolaj et al., 2018) and classification accuracy to quantitatively evaluate the overall performance of different image translation models. KID is defined as the squared maximum mean discrepancy between feature representations of real and generated images. Such feature representations are obtained from the Inception network [20]. In contrast to other criteria, KID has an unbiased estimator, which makes it more reliable, especially when there are fewer test images than the dimensionality of the inception features. A good image-to-image translation should have a low KID value which indicates higher visual similarities between the generated images and the real images. In our work, we build four datasets, and each dataset contains 50 real images and 50 generated images from the same domain. We input images into the Inception network, and then obtain the feature representations of all images. We calculate the KID value for each dataset, and use the mean KID value over the four datasets to evaluate the performance of each translation model. The experimental results are shown in Table 8.

In addition, to evaluate the classification accuracy on the generated images, we also choose Inception network [20] as the classifier due to its better performance for object recognition. We use its publicly released model pretrained on the ImageNet dataset, and use a softmax layer as the classification layer which is refined using the second dataset in our work. The training dataset contains 4000 photos of the Alps, each labeled with its season. We select 50 generated images from each domain as the testing dataset. The classification accuracy is defined as the fraction of test images which are correctly classified by the refined Inception network. The experimental results are also shown in Table 8.

We get the similar results as that in Experiment 2. The CycleGAN produces the best results. However, the

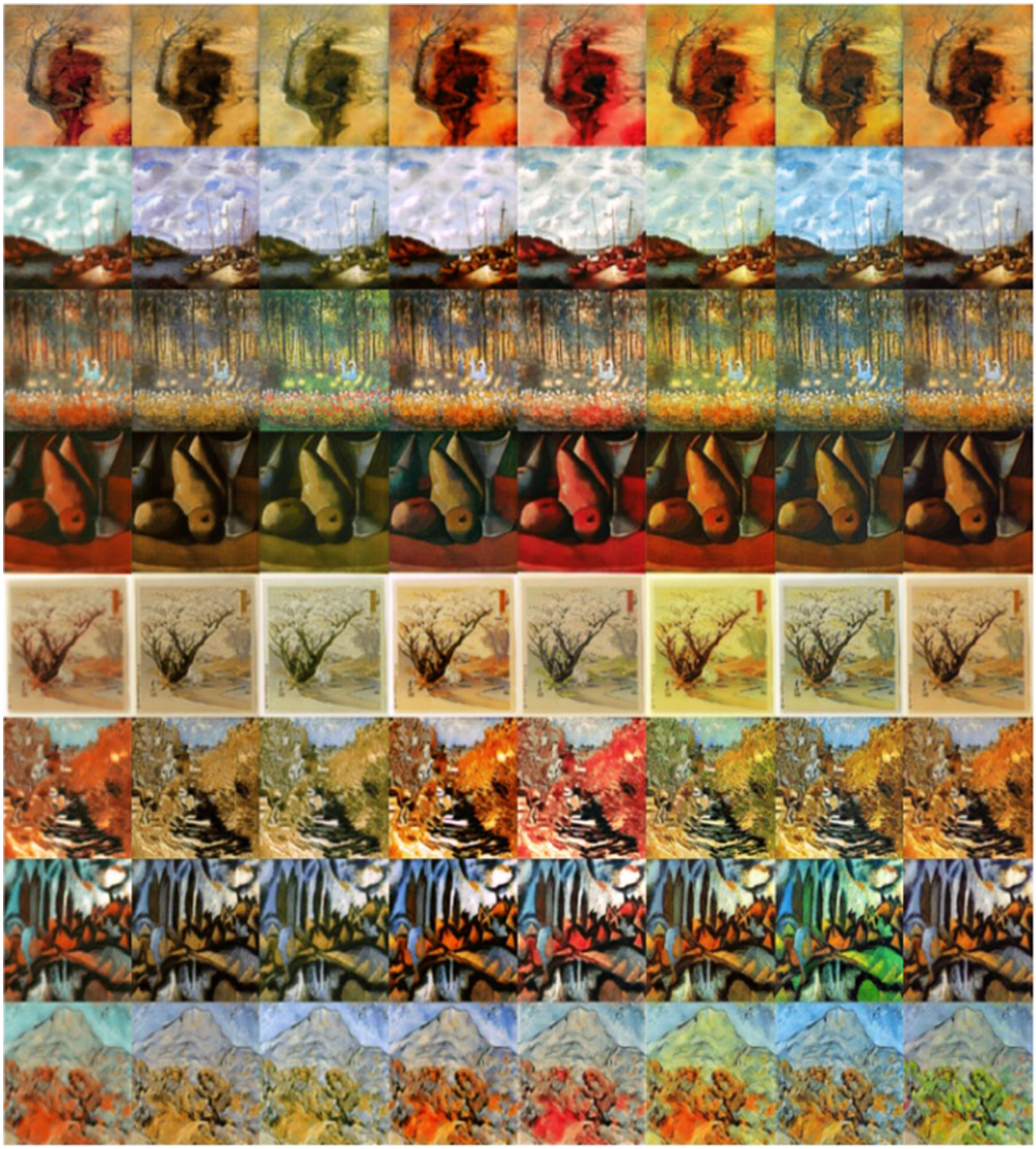

Fig. 11 The translated samples of the different artists

performance of HssGAN is comparable to CycleGAN and ComboGAN. The performance of HssGAN is significantly lower than that of HssGAN-HAL, which also proves that the adversarial component in the hidden space can improve the feature representation power of the common encoder.

Experiment 4: We use even more image domains to evaluate the performance of HssGAN-HAL. The third dataset contains paintings from fourteen artists. We select the paintings of eight artists from the dataset and train a model that can translate images among the eight domains. Due to the limited computing resources, we only evaluate

Table 6 MTurk "real vs fake" evaluation on the different translation models

Seasons	CycleGAN(%) Turkers labeled real	HssGAN(%) Turkers labeled real	HssGAN-HAL(%) Turkers labeled real	ComboGAN (%) Turkers labeled real
sp- > su	**30**	20	26	30
su- > sp	**30**	20	26	28
sp- > au	**64**	34	58	**64**
au- > sp	**20**	20	20	20
sp- > wi	30	30	**40**	30
wi- > sp	**44**	34	36	40
su- > au	30	20	**40**	30
au- > su	**46**	46	40	**46**
su- > wi	**70**	30	66	65
wi- > su	**46**	36	40	**46**
au- > wi	**60**	40	52	55
wi- > au	**54**	44	**54**	52
Avg.MTurk	**43.6**	31.2	41.5	42.1
Training Time	468 h	28 h	30 h	120 h

Table 7 Reconstruction errors of the different translation models

Seasons	CycleGAN Recon. Err	HssGAN Recon. Err	HssGAN-HAL Recon. Err	ComboGAN Recon. Err
sp- > su- > sp	124	129	127	**120**
sp- > au- > sp	**129**	137	134	132
sp- > wi- > sp	116	139	**109**	115
su- > sp- > su	**118**	125	120	123
su- > au- > su	119	123	**110**	122
sur- > wi- > su	**126**	134	127	128
au- > sp- > au	**100**	115	110	108
au- > su- > au	**58**	61	70	60
au- > wi- > au	**120**	142	138	125
wi- > sp- > wi	115	122	117	**110**
wi- > su- > wi	**96**	120	105	100
wi- > au- > wi	**112**	144	117	120
Avg. Recon. Err	**111**	124	115	113

Table 8 KID value and classification accuracy of the different translation models

Model	CycleGAN	HssGAN	HssGAN-HAL	ComboGAN
KID Value	**10.85**	12.10	11.02	11.21
Class. Acc	**83.0%**	76%	81.5%	80.5%

the performance of HssGAN-HAL. Some translated samples generated by HssGAN-HAL are shown in Fig. 11. The images on the diagonal are the ground-truth images, and each column represents a painting style. It is more difficult to evaluate whether a translated painting has the painting style of a target domain in this dataset than in the other datasets. We can simply see that the paintings in each column have similar textures and similar colors, which shows that HssGAN-HAL has the capability to translate images among even more domains.

5 Conclusion

In this paper, we propose a novel GAN structure named HssGAN to reduce the computing complexity of CycleGAN for translating images among multiple domains. HssGAN contains only one common encoder that is used to learn the common features of the different domains. Therefore, the model complexity of HssGAN increases linearly with the number of image domains. In addition, we

introduce the generative adversarial component into HssGAN and use adversarial training to further improve the representational power of the common encoder.

The experimental results show that the proposed model can significantly decrease the time cost for model training, and the translated results are comparable to those of CycleGAN and ComboGAN. In our study, we run MTurk "real vs fake" test to evaluate the quality of the generated images, which is a labor intensive work. How to evaluate the quality of the generated images objectively and effectively is our future research work.

Acknowledgements This work was partially supported by the National Key R&D Program of China under Grant no. 2018YFB1003800, 2018YFB1003805, the National Natural Science Foundation of China (Grant No. 61872107), Scientific Research Foundation in Shenzhen (Grant No. JCYJ20180306172156841, JCYJ20180507183608379).

Funding The National Key R&D Program of China under Grant no. 2018YFB1003800, 2018YFB1003805, the National Natural Science Foundation of China (Grant No. 61872107), Scientific Research Foundation in Shenzhen (Grant No. JCYJ20180306172156841, JCYJ20180507183608379).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets, In Proceedings of the International Conference on Neural Information Processing Systems, pp. 2672–2680, Montreal, MIT Press, Canada
2. Zhu J-Y, Park T, Isola P, et al. (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of International Conference on Computer Vision, Pages 2223–2230, Venice, Italy, IEEE.
3. Isola P, Zhu J-Y, Zhou T, et al (2017) Image-to-image translation with conditional adversarial networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Pages 5967–5976, Hawaii, USA. IEEE.
4. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Pages 2414–2423, Las Vegas, USA.IEEE.
5. Gatys LA, Ecker AS, Bethge M (2015) A neural algorithm of artistic style. arXiv:1508.06576.
6. Ulyanov D, Lebedev V, Vedaldi A, et al. (2016) Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In the Proceedings of International Conference on Machine Learning. Pages 1349–1357, NY, USA. IMLS.
7. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In the proc. of IEEE Conference on Computer Vision and Pattern Recognition. Pages 3431–3440, Boston, USA. IEEE.
8. Xu S, Zhu Q, Wang J (2020) Generative image completion with image-to-image translation. Neural Comput & Applic 32:7333–7345
9. Karatsiolis S, Schizas CN, Petkov N (2020) Modular domain-to-domain translation network. Neural Comput & Applic 32:6779–6791
10. Pathak D, Krahenbuhl P, Donahue J, et al.(2016) Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Pages 2536–2544, Las Vegas , USA. IEEE.
11. Zhang R, Isola P, Efros A A (2016) Colorful image colorization. In Proceedings of European Conference on Computer Vision. Pages 649–666, Amsterdam, Netherlands. Springer.
12. Hertzmann A, Jacobs C E, Oliver N, et al (2001) Image analogies.p. In Proceedings of the ACM SIGGRAPH, Pages 327–340. Los Angeles, USA. ACM.
13. Choi Y, Choi M, Kim M, et al. (2017) StarGAN: Unified Generative Adversarial Networks forMulti-Domain Image-to-Image Translation. ArXiv e-prints. arXiv:1711.09020v1.
14. Hsu SY,Yang CY, Huang CC,et al. (2018) SemiStarGAN: Semi-supervised Generative Adversarial Networks for Multi-domain Image-to-Image Translation. Asian Conference on Computer Vision, pages 338–353.
15. Anoosheh A, Agustsson E, Timofte R, Van Gool L (2018) ComboGAN: unrestrained scalability for image domain translation. IEEE/CVF Con Comput Vis Pattern Recognit Workshops (CVPRW) 2018:896–8967
16. Jing Y, Yang Y, Feng Z, et al. (2017) Neural Style Transfer: A Review. arXiv:1705.04058.
17. Salimans T, Goodfellow I, Zaremba W, et al. (2016) Improved techniques for training gans. In Proc. of International Conference on Neural Information Processing Systems. Pages 2234–2242, Barcelona, Spain. MIT Press.
18. Donahue J, Krahenbuhl P, Darrell T (2016) Adversarial feature learning. arXiv:1605.09782
19. Dumoulin V, Belghazi I et al. (2016) Adversarially learned inference. arXiv:1606.00704
20. Szegedy C., Vanhoucke V., Ioffe S., et al. (2016) Rethinking the Inception Architecture for Computer Vision. IEEE Conference on Computer Vision and Pattern Recognition, Pages 1–10, Las Vegas, USA, IEEE.
21. Efros AA, Freeman WT (2001) Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and Interactive Techniques. Pages 341–346, Los Angeles, USA. ACM.
22. Justin J, Alexandre A, Li Fei-Fei (2016) Perceptual losses for real-time style transfer and super-resolution. In the Proceedings of European Conference on Computer Vision. Pages 694–711, Amsterdam, Netherlands. Springer.
23. Li C, Wand M (2016) Precomputed real-time texture synthesis with markovian generative adversarial networks. In Proc. of European Conference on Computer Vision. Pages 702–716, Amsterdam, Netherlands. Springer.
24. Liu M-Y, Tuzel O(2016) Coupled generative adversarial networks. In Proc. of International Conference on Neural Information Processing Systems. Pages 469–477, Barcelona, Spain. MIT Press.
25. Mikołaj B., Danica J. , Michael A., et al. (2018) Demystifying MMD GANs. International Conference on Learning Representations, Pages 1–36, Vancouver, Canada.
26. Taigman Y, Polyak A, Wolf L (2017) Unsupervised cross-domain image generation. In Proc. 5th International Conference on Learning Representations, Pages 1–6, Toulon, France. ICLR.