Improving Variational Autoencoder with Deep Feature Consistent and Generative Adversarial Training

Xianxu Hou[a,b], Ke Sun[d], Linlin Shen[b,c], Guoping Qiu[a,b,e,*]

[a]College of Information Engineering, Shenzhen University, Shenzhen, China
[b]Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen, China
[c]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[d]Key Laboratory of Spatial Information Smarting Sensing and Services, Shenzhen University, Shenzhen, China
[e]School of Computer Science, University of Nottingham, Nottingham, United Kingdom

Abstract

We present a new method for improving the performances of variational autoencoder (VAE). In addition to enforcing the deep feature consistent principle thus ensuring the VAE output and its corresponding input images to have similar deep features, we also implement a generative adversarial training mechanism to force the VAE to output realistic and natural images. We present experimental results to show that the VAE trained with our new method outperforms state of the art in generating face images with much clearer and more natural noses, eyes, teeth, hair textures as well as reasonable backgrounds. We also show that our method can learn powerful embeddings of input face images, which can be used to achieve facial attribute manipulation. Moreover we propose a multi-view feature extraction strategy to extract effective image representations, which can be used to achieve state of the art performance in facial attribute prediction.
Keywords: Image Generation, Facial Attributes, Generative model, VAE, GAN

1. Introduction

Deep convolutional neural networks (CNNs) [1] have been used to achieve state of the art performances in many computer vision and image processing tasks such as image classification [2, 3, 4], retrieval [5], detection [6], captioning [7], human pose

*Corresponding author
Email address: qiu@szu.edu.cn, guoping.qiu@nottingham.ac.uk (Guoping Qiu)

recovery [8, 9, 10], image privacy protection [11], unsupervised dimension reduction [12] and many other applications [13, 14, 15]. Deep convolutional generative models, as a branch of unsupervised learning technique in machine learning, have become an area of active research in recent years. A generative model trained with a given image database can be useful in several ways. One is to learn the essence of a dataset and generate realistic images similar to those in the dataset from random inputs. The whole dataset is "compressed" into the learned parameters of the model, which are significantly smaller than the size of the training dataset. The other is to learn reusable feature representations from unlabeled image datasets for a variety of supervised learning tasks such as image classification.

In this paper, we propose a new method to train the variational autoencoder (VAE) [16] to improve its performance. In particular, we seek to improve the quality of the generated images to make them more realistic and less blurry. To achieve this, we employ objective functions based on deep feature consistent principle [17] and generative adversarial network [18, 19] instead of the problematic per-pixel loss functions. The deep feature consistent can help capture important perceptual features such as spatial correlation through the learned convolutional operations, while the adversarial training helps to produce images that reside on the manifold of natural images. We also introduce several techniques to improve the convergence of GAN training in this context. In particular, instead of directly using the generated images and the real images in pixel space, the corresponding deep features extracted from pretrained networks are used to train the generator and the discriminator network. We also propose to further relax the constraint on the output of the discriminator network to balance the image reconstruction loss and the adversarial loss. We present experimental results to show that our new method can generate face images with much clearer facial parts such as eyes, nose, mouth, teeth, ears and hair textures. We show that the VAE trained by our method can capture the semantic information of facial attributes, which can be modeled linearly in the learned latent space. Furthermore, we show that the trained VAE can be used to extract more discriminative facial attribute representations that can be used to achieve state of the art performance in facial attribute recognition. Concretely, our contributions are threefold:

- Our model seamlessly associates the two modalities, i.e., VAE and GAN through a common latent embedding space and we validate the effectiveness of this approach on image generation tasks.

- We show that the learned latent representations can capture conceptual and semantic information of the input face images, which can be used to achieve facial attribute manipulation.

- Lastly we introduce a multi-view feature extraction strategy on facial attribute recognition experiments in which we surpass state of the art.

The rest of the paper is organized as follows. We first briefly review the related literature in Section 2. Section 3 presents our method to improve variational autoencoder with deep feature consistent and generative adversarial training. Section 4 presents experimental results which show that our method stands out as a state of the art technique. Finally we present a discussion and conclude the paper in Section 5 and Section 6.

2. Related Work

2.1. Variational autoencoder

Deep convolutional autoencoder is a powerful learning model for representation learning and has been widely used for different applications [8, 20, 21, 22, 23, 24, 25, 9]. Variational Autoencoder (VAE) [16, 26] has become a popular generative model, allowing us to formalize image generation task in the framework of probabilistic graphical models with latent variables. Firstly it encodes an input image x to a latent vector $z = E(x) \sim q(z|x)$ with an encoder network E, and then a decoder network D is used to decode the latent vector z back to image space, i.e., $\bar{x} = D(z) \sim p(x|z)$. In order to achieve image reconstruction we need to maximize the marginal log-likelihood of each observation (pixel) in x, and the VAE reconstruction loss $\mathcal{L}_{rec}$ is the negative expected log-likelihood of the observations in x. Another key property of VAE is the ability to control the distribution of the latent vector z, which has characteristic of being independent unit Gaussian random variable, i.e., $z \sim \mathcal{N}(0, I)$. Moreover, the difference between the distribution of $q(z|x)$ and the distribution of a Gaussian distribution (called

KL Divergence) can be quantified and minimized by gradient descent algorithm [16]. Therefore, VAE models can be trained by optimizing both of the reconstruction loss $\mathcal{L}_{rec}$ and KL divergence loss $\mathcal{L}_{kl}$.

$$\mathcal{L}_{rec} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{1}$$

$$\mathcal{L}_{kl} = D_{kl}(q(z|x)||p(z)) \tag{2}$$

$$\mathcal{L}_{vae} = \mathcal{L}_{rec} + \mathcal{L}_{kl} \tag{3}$$

Several methods have been proposed to improve the performance of VAE. [27] and [28] proposed to build variational autoencoders by conditioning on either class labels or on a variety of visual attributes, and their experiments demonstrate that they are capable of generating realistic faces with diverse appearances. Deep Recurrent Attentive Writer (DRAW) [29] combines spatial attention mechanism with a sequential variational auto-encoding framework that allows iterative generation of images. [30] and [17] consider replacing per-pixel loss with perceptual similarities using either multi-scale structural similarity score or a perceptual loss based on deep features extracted from pretrained deep networks.

2.2. Generative adversarial network

Generative Adversarial Network (GAN) framework is firstly introduced by [18] to estimate generative models based on a min-max game. Under the GAN framework two models are simultaneously trained: a generator network $G(z)$ used to map a noise variable z to data space, a discriminator network $Dis(x)$ designed to distinguish between the samples from the true training data and generated samples produced by the generator $G(z)$. The discriminator $Dis(x)$ is optimized by maximizing the probability of assigning the correct label for each category. The generator network $G(z)$ is trained simultaneously to minimize $log(1-Dis(G(z)))$ by playing against the adversarial discriminator network $Dis(x)$. Thus the min-max game between $G(z)$ and $Dis(x)$ can be formulated as follows:

$$\min_{G} \max_{Dis} V(Dis, G) = \mathbb{E}_x[log(Dis(x))] + \mathbb{E}_z[log(1 - Dis(G(z)))] \tag{4}$$

Following works [19, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41] have focused on improving the perceptual quality of GAN outputs and the training stability of GAN through architectural innovations and new training techniques. Our model enjoys both the advantages of deep feature consistent VAE (DFC-VAE) [17] and Wasserstein GAN (WGAN) [19] to improve the perceptual quality of the output images generated by VAE and enhance the effectiveness of VAE representations for semi-supervised learning. In addition, a combination of VAE and GAN was also proposed by [42]. Whilst there is a similarity, there are some differences as well. We use a pre-trained VGGNet as feature extractor to extract features of the input image and the output image and calculate the loss function. In reference [42], they used the GAN discriminator network to extract image features to calculate the loss function and this discriminator was updated during the GAN training. Additionally, we adopt the framework of WGAN [19] to achieve adversarial training while DCGAN [32] was adopted in [42].

2.3. Learned features for image synthesis

Neural style transfer [43] is among the most successful applications of image synthesis based on the learned convolutional features in recent years. It tries to combine the content of one image with the style of another image by jointly optimizing content reconstruction loss and style reconstruction loss based on the features extracted from a pretrained convolutional neural network. Other works try to train a feed-forward network for real-time style transfer [44, 45, 46]. In addition, images can be also generated by maximizing classification scores or individual features [47, 48] for a better understanding of the trained networks. Furthermore high-confidence fooling images can be also synthesized through a similar optimizing technique [49, 50].

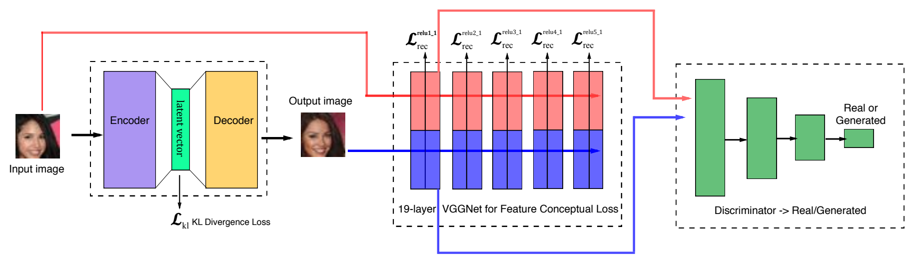

Figure 1: Model overview. From left to right: The Variational autoencoder (VAE), the VGGNet used for feature extraction and WGAN discriminator. Note that the inputs fed to the discriminator come from the first convolutional layer of the VGGNet.

3. Method

3.1. Overview

As shown in Figure 1, our model consists of three components: a variational autoencoder including an encoder network $E(x)$ and a decoder network $D(z)$, a pre-trained VGGNet $\Phi(x)$ for feature extraction and a classifier network used as discriminator $Dis(x)$. Both the encoder and the decoder are deep residual convolutional neural networks with a 100-dimensional latent vector. The encoder processes the input image into the latent vector which is then decoded to an output image. In order to train a VAE, we need two losses, one is KL divergence loss $\mathcal{L}_{kl} = D_{kl}(q(z|x)||p(z))$ [16], which is used to make sure that the latent vector z is an independent unit Gaussian random variable. The other is a feature reconstruction loss, which is based on the features extracted from VGGNet. Specifically we feed both of the input and output images to the pre-trained network Φ respectively and then measure the difference between the hidden layer representations, i.e., $\mathcal{L}_{rec} = \mathcal{L}^1 + \mathcal{L}^2 + ... + \mathcal{L}^l$, where $\mathcal{L}^l$ represents the feature reconstruction loss at the l^{th} hidden layer. Furthermore, the VAE also serves as the generator and works with the discriminator to play the GAN game. Instead of feeding the pixels to the discriminator, we propose to use the first layer's output of the VGGNet as the input of the discriminator. The purpose is to enable more stable training as well as use as much low level image information as possible. It is worth noting that our architecture is different from that of [42]. Whilst they use the hidden layer features of the

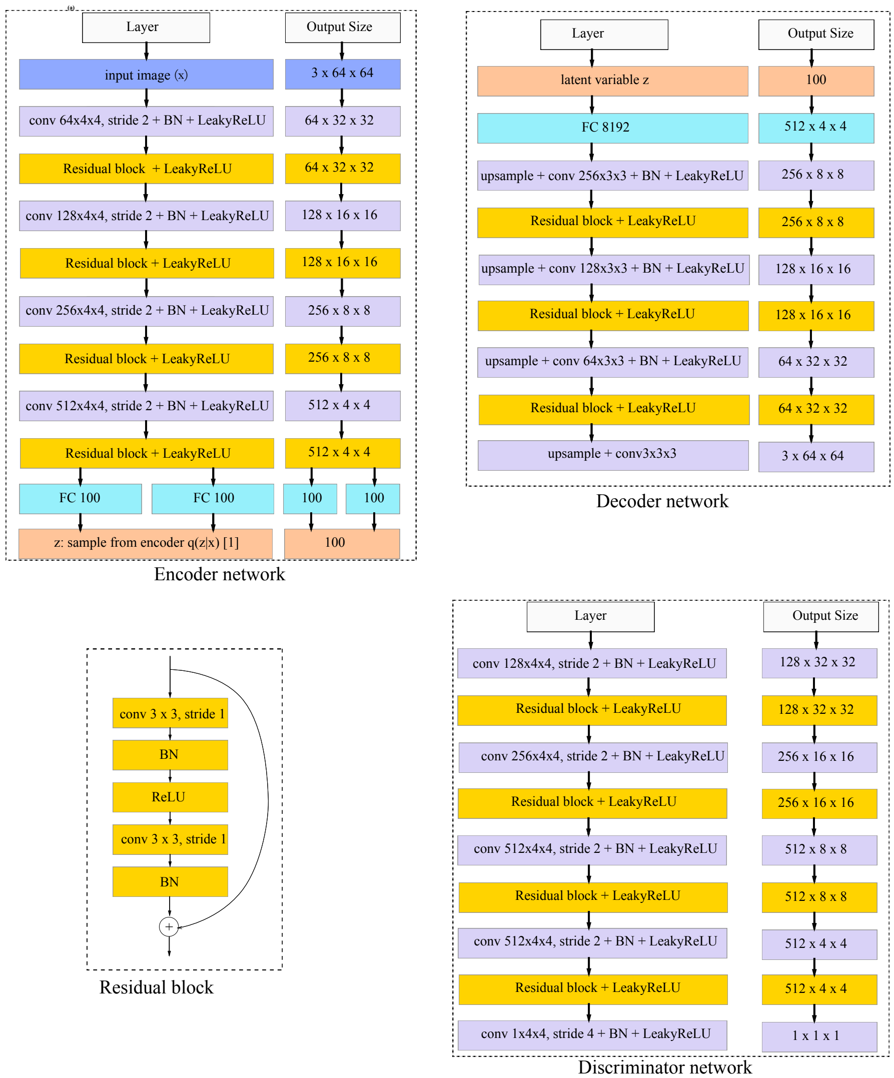

Figure 2: The architecture of the autoencoder and discriminator network.

GAN discriminator to compute the image reconstruction loss, we adopt a pre-trained VGGNet. What's more, the pre-trained VGGNet is fixed during training and it still allows feed-forward and back-propagation computation. As a result, our model can be trained end-to-end.

3.2. Neural network architecture

As shown in Figure 2, both of the autoencoder and discriminator network are deep residual convolutional neural networks based on [4, 32]. We construct 4 convolutional layers in the encoder network with 4 $\times$ 4 kernel and 2 $\times$ 2 stride to achieve spatial downsampling instead of using deterministic spatial functions such as maxpooling. Each convolutional layer is followed by a batch normalization layer and a LeakyReLU activation layer. In addition, a residual block is added after each convolutional layer and all the residual blocks contain two 3 $\times$ 3 kernel convolutional layers with the same number of filters. Lastly two fully-connected output layers (for mean and variance) are added to the encoder and will be used to calculate the KL divergence loss and sample latent variable z (see [16] for details).

For the decoder, we use 4 convolutional layers with 3 $\times$ 3 kernels and 1 $\times$ 1 stride. We also propose to replace standard zero-padding with replication padding, i.e., feature map of an input is padded with the replication of the input boundary. Similar to the encoder, each convolutional layer is also followed by a residual layer except the last one. For upsampling we use nearest neighbor method by a scale of 2 instead of fractional-strided convolutions used by other works [51, 32]. We also use batch normalization to help stabilize the whole training and use LeakyReLU as the activation function.

The design of the discriminator follows the architectural innovations of DCGAN [32]. We use convolutional layers with 4 $\times$ 4 kernel and 2 $\times$ 2 stride to achieve spatial downsampling and add a residual block after each convolutional layer except the last layer. Like WGAN [19], the sigmoid layer is removed in the last layer and use a 4 $\times$ 4 stride convolution layer to produce a single output, and the gradients of discriminator is clipped between -0.01 to 0.01.

3.3. Feature reconstruction loss

Feature reconstruction loss of two images is defined as the difference between the hidden features in a pretrained deep convolutional neural network Φ. Similar to [43], we use VGGNet [3] as the loss network in our experiment. The core idea of feature reconstruction loss is to seek consistency between two images in the learned feature space. As the hidden representations can capture important perceptual quality features such as spatial correlation, a smaller difference of hidden representations indicates a better consistency of spatial correlations between the input and the output, as a result, we can get a better visual quality of the output image. Specifically, let $\Phi_l(x)$ denotes the representation of the l^{th} hidden layer when input image x is fed to network Φ. Mathematically $\Phi_l(x)$ is a 3D volume block array of shape [C_l x W_l x H_l], where C_l is the number of filters, W_l and H_l denote the width and height of each feature map for the l^{th} layer. The feature reconstruction loss for one layer ($\mathcal{L}_l$) between two images x and $\bar{x}$ can be simply defined by squared Euclidean distance. Actually it is quite like the per-pixel loss for images except that the number of color channels is not 3 anymore.

$$\mathcal{L}_l = \frac{1}{2C_l W_l H_l} \sum_{c=1}^{C_l} \sum_{w=1}^{W_l} \sum_{h=1}^{H_l} (\Phi_l(x)_{c,w,h} - \Phi_l(\bar{x})_{c,w,h})^2 \tag{5}$$

Instead of only using a single layer features, we leverage visual features in different layers and combine the outputs of the five convolutional layers of the VGGNet. The final reconstruction loss is defined as:

$$\mathcal{L}_{rec} = \sum_{l=1}^{L} \frac{100}{C_l^2} \mathcal{L}_l \tag{6}$$

where $\mathcal{L}_l$ and C_l are the feature loss and the number of filters at l^{th} layer respectively, L is total convolutional layers in the pretrained network.

Additionally we adopt the KL divergence loss $\mathcal{L}_{kl}$ [16] to regularize the encoder network to control the distribution of the latent variable z. To train VAE, we jointly minimize the KL divergence loss $\mathcal{L}_{kl}$ and the feature reconstruction loss $\mathcal{L}_{rec}$ for different layers as follows:

$$\mathcal{L}_{vae} = \alpha \mathcal{L}_{kl} + \beta \mathcal{L}_{rec} \tag{7}$$

where α and β are the weighting parameters for KL Divergence loss and feature reconstruction loss. It is worth noting that the pre-trained VGGNet is used for feature extraction only and is fixed during the training. The latent representation of the image refers to the latent variable of the autoencoder in our paper.

3.4. Adversarial loss

In addition to the feature reconstruction loss described above, we also incorporate variational autoencoder in the framework of generative adversarial network to encourage the VAE to produce outputs that reside on the manifold of natural images. Our adversarial training is based on WGAN [19]. In order to further improve the training stability, instead of directly feeding the real images and generated images to a discriminator, we first extract the first layer features of the pretrained VGGNet and feed them to the discriminator network. It is because we would like to push the reconstructed image similar to natural images in terms of low-level information, which can be often obtained from lower layers of deep networks. In addition, we propose another technique to further relax the constraint on the output of the discriminator network. WGAN [19] proposes to remove the last Sigmoid layer in the generator and use 1 and -1 as ground-truth label for real and generated images. In our experiments, we found that GAN training could collapse and the VAE training tends to dominate the training when using too small labels, e.g., 1 and -1. In addition, we also found that the adversarial loss would dominate the training by using too big labels like -100 and 100, which could lead to structural changes of the reconstructed images. Using empirical values 10 and -10 to represent ground-truth labels, we can effectively balance well between the VAE and GAN, and generate diverse synthesized results in a more natural and flexible manner.

Finally our entire deep model can be trained end-to-end with a combination of KL divergence loss, reconstruction loss and adversarial loss as Equation 8 and the training procedure is summarized in Algorithm 1.

$$\mathcal{L}_{vae} = \alpha\mathcal{L}_{kl} + \beta\mathcal{L}_{rec} + \mathcal{L}_{GAN} \tag{8}$$

4. Experiments

In this paper, we conduct experiments on CelebFaces Attributes (CelebA) [52] and CIFAR-10 [53] Dataset to evaluate our method on the performance of image generation. We also study how different layer features of the pre-trained VGGNet affects the

Algorithm 1 Training VAE-WGAN Model

Require: c, the clipping parameter; Φ, pretrained model

$W_{Encoder}, W_{Decoder}, W_{Discriminator} \leftarrow$ Initialize parameters

repeat

$X \leftarrow$ random mini-batch images from the dataset

$Z \leftarrow$ Encoder(X)

$\mathcal{L}_{kl} \leftarrow D_{KL}\left(q(Z|X)||p(Z)\right)$

$\hat{X} \leftarrow$ Decoder(Z)

$\mathcal{L}_{rec} \leftarrow ||\Phi(X) - \Phi(\hat{X})||^2$

$\mathcal{L}_{GAN} \leftarrow Discriminator(X) - Discriminator(\hat{X})$ // Wasserstein GAN

$W_{Encoder} \stackrel{+}{\leftarrow} -\nabla_{W_{\text{Encoder}}}\left(\mathcal{L}_{kl} + \mathcal{L}_{rec} - \mathcal{L}_{GAN}\right)$

$W_{Decoder} \stackrel{+}{\leftarrow} -\nabla_{W_{\text{Decoder}}}\left(\mathcal{L}_{rec} - \mathcal{L}_{GAN}\right)$

$W_{Discriminator} \stackrel{+}{\leftarrow} -\nabla_{W_{\text{Discriminator}}}\mathcal{L}_{GAN}$

$W_{Discriminator} \leftarrow clip\left(W_{Discriminator}, -c, c\right)$

until convergence of parameters

performances of image synthesis. Furthermore, we consider manipulating the facial attributes in the learned latent space. Finally we apply the learned representations to facial attribute recognition and show that we can achieve state of the art performances.

4.1. Training details

CelebA is a large-scale face attribute dataset with 202,599 face images, 5 landmark locations and 40 binary attributes annotations per image. We build the training dataset by cropping and scaling the aligned images to 64 $\times$ 64 pixels like [42, 32]. The CIFAR-10 dataset consists of 60,000 images of shape 32 $\times$ 32 in 10 classes. There are 50,000 training images and 10,000 test images. For both datasets, we train our model with a batch size of 64 for 5 epochs over the training dataset and use Adam method for optimization [54] with an initial learning rate of 0.0005, which is decreased by a factor of 0.5 for the following epochs. The 19-layer VGGNet [3] is chosen as loss network Φ to construct feature reconstruction loss for image reconstruction. The loss weighting parameters α and β are 1 and 0.5 respectively. Our implementation is built on deep

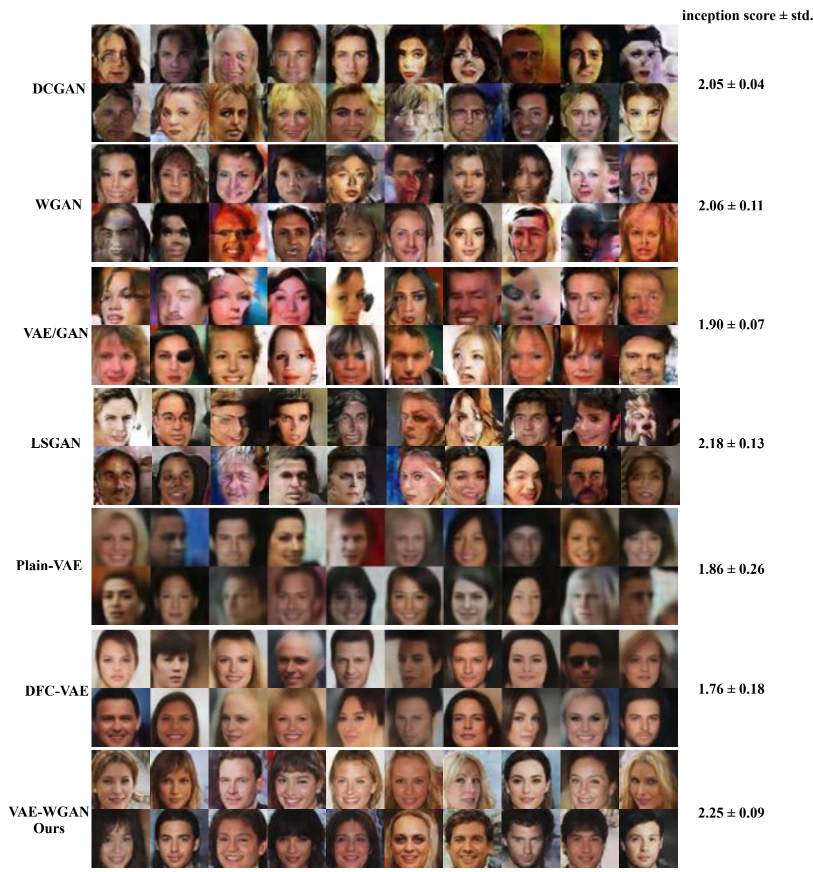

Figure 3: Face images generated from 100-dimension latent vector $z \sim \mathcal{N}(0, 1)$ by different models. We compare our VAE-WGAN with DCGAN[32], WGAN[19], VAE/GAN[42], LSGAN[41], Plain-VAE[16] and DFC-VAE[17].

learning framework Torch [55]. As for the computational time, it takes around 10 hours to train our models and 0.012 seconds to process an image of size 64×64 during testing. The training and testing time are both benchmarked on a single GTX 1080Ti GPU.

4.2. Qualitative results for image generation

The comparison is divided into two parts: one is arbitrary image generation decoded from vectors z randomly drawn from $\mathcal{N}(0, 1)$, the other is natural image reconstruction.

Figure 4: Cifar images generated from 100-dimension latent vector $z \sim \mathcal{N}(0, 1)$ by different models. We compare our VAE-WGAN with DCGAN[32], WGAN[19], VAE/GAN[42], LSGAN[41], Plain-VAE[16] and DFC-VAE[17].

4.2.1. Arbitrary image generation.

First, we compare the perceptual quality of the output face images for different generative models. As shown in Figure 3 and 4, we compare our model VAE-WGAN with Plain-VAE [16], DFC-VAE [17], DCGAN [32], WGAN [19], VAE/GAN [42] and LSGAN[41]. All the compared models are implemented with the public available code from the corresponding papers with default settings. The final output images are produced by feeding vectors randomly drawn from a given distribution $\mathcal{N}(0, 1)$ to either VAE decoder or GAN generator. We can see that DCGAN, WGAN as well as LSGAN can generate clean and sharp images, however the image details can be distorted, resulting in unsatisfactory outputs with weird appearance like unpleasing faces. It is be-

Figure 5: Face images reconstructed by different models. We compare our VAE-WGAN with Plain-VAE[16], VAE/GAN[42] and DFC-VAE[17].

Figure 6: CIFAR images reconstructed by different models. We compare our VAE-WGAN with Plain-VAE[16], VAE/GAN[42] and DFC-VAE[17].

cause there no input image information for pure GAN training. In contrast, the results produced by VAE decoder can better preserve the overall object structures. However, Plain-VAE tends to produce very blurry images because it tries to minimize the per-pixel loss between two images and each pixel is optimized independently. DFC-VAE can produce clear and sharp images because the feature reconstruction loss contains the perceptual and spatial correlation information in the learned feature space. VAE/GAN and our VAE-WGAN can achieve better results than all the other models, however VAE/GAN still suffers from observed distortions. Our method can generate more consistent and realistic human faces with much clearer noses, eyes, teeth, hair textures as well as reasonable backgrounds. Moreover, our method can achieve highest inception scores [34] on the two dataset as shown in Figure 3 and 4. The inception scores are calculated based on 2,000 images for each model.

4.2.2. Image reconstruction.

We also evaluate the reconstruction performance of our method (shown in Figure 5 and 6) by comparing with Plain-VAE, DFC-VAE [17] and VAE/GAN [42]. Pure GAN models are not involved because of no input images in their models. Similar to arbitrary images generated above, Plain-VAE reconstructs very blurry images because of the shortcomings of per-pixel loss. DFC-VAE can produce better images such as faces with clear eyes and mouths, however it still produces blurry background for CIFAR images and unrealistic hairs for face images. The results of VAE/GAN show that the images are reasonably sharp and clear, however details in the original images are missing. Again our model can produce much better reconstruction results than other models. Our model is better at preserving the original color and overall structures of the input images.

4.2.3. Impact of different level reconstruction loss

We also conduct experiments to investigate how features of different level convolutional layers of the loss network affect the quality of image generation. Figure 7 shows the randomly generated face images by our five models trained with feature reconstruction loss based on layers relu1_1, relu2_1 relu3_1, relu4_1 and relu5_1 respectively. It can be seen that all the generated images are able to keep the overall structures of faces. However as we reconstruct from lower level layers like relu1_1, the generated images are very blurry especially in the hair and background area. When using higher level layers, the generated face images are much sharper and can show reasonable hair textures,

Figure 7: Generated face images from 100-dimension latent vector $z \sim \mathcal{N}(0, 1)$ by 5 different models, which are trained with feature reconstruction loss based on layers relu1_1, relu2_1 relu3_1, relu4_1 and relu5_1 respectively.

but the exact structure of facial attributes cannot be preserved like eyes and mouths. One explanation for this is that the higher level features are corresponding to a coarser space area of the encoded image. The areas covered by relu4_1 and conv5_1 layers are too large to construct local facial attributes like mouth and eyes, but better for larger area textures like hair. Overall we can get better results when using reconstruction loss by combining different layers.

4.2.4. Impact of weighting parameters α and β

We further conduct experiments to look into the influences of weighting parameters α and β in Equation 7 in terms of image quality. Specifically we train two models with $\alpha = 1, \beta = 0.01$ and $\alpha = 0.01, \beta = 1$ respectively. As shown in Figure 8 and Figure 9, we can see that the images can be better reconstructed when using bigger β, however the randomly generated images look weird with unusual face shapes. In addition, the randomly generated images are similar to the reconstructed ones with bigger α while they usually suffer from the problems of poor quality and lack of diversity. It is clear that the α and β can be used to balance the trade-off between the latent variable distribution and image reconstruction in variational autoencoder. As shown in previous sections, our model works well with $\alpha = 1$ and $\beta = 0.5$.

In addition, we also conduct experiments without reconstruction loss. As shown α=0.01, β = 1

α=1, β = 0.01

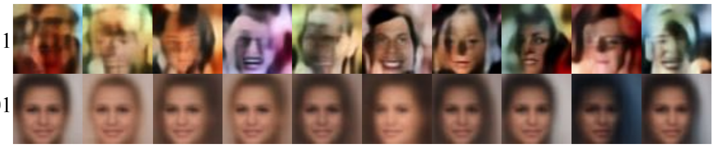

Figure 8: Randomly generated images by our method with different weighting parameters α and β.

Input

α=0.01, β = 1

α=1, β = 0.01

Figure 9: Reconstruction results by our method with different weighting parameters α and β.

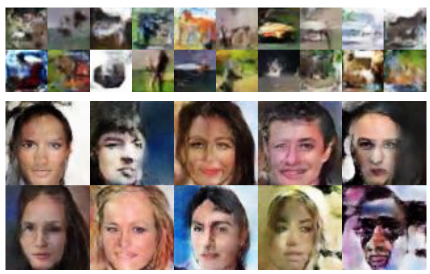

Figure 10: Generated images by our method without reconstruction loss.

in Figure 10, we can see that the results are similar to those trained with DCGAN and WGAN. This is because the latent vector distribution is similar to the pre-defined Gaussian distribution without reconstruction constraint. Thus the whole training processing is roughly equal to a pure GAN training.

4.3. The learned latent space

In order to get a better understanding of what our model has learned, we investigate the property of the learned representation in the latent space. What's more, we also conduct experiments to show the effectiveness of our model to learn meaningful feature representations beyond image generation. In particular, we visualize the latent representations based on the t-SNE embedding and also apply them to the facial

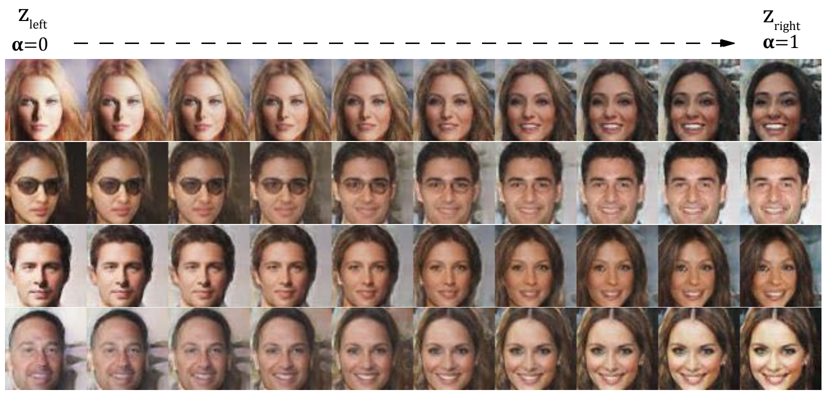

Figure 11: Linear interpolation of latent vector. Each row is the interpolation from left latent vector z_{left} to right latent vector z_{right}. e.g. $(1-\alpha)z_{left} + \alpha z_{right}$.

attribute recognition task.

4.3.1. Linear interpolation of latent space

As shown in Figure 11, we have studied the linear interpolation between the generated images from two latent vectors denoted as z_{left} and z_{right}. The interpolation is defined by a simple linear transformation $z = (1-\alpha)z_{left} + \alpha z_{right}$, where $\alpha = 0, 0.1, \ldots, 1$, and then z is fed to the decoder network to generate new face images. From the first row in Figure 11, we can see the smooth transitions between $vector$("Woman without smiling and blond hair") and $vector$("Woman with smiling and black hair"). Little by little the color of the hair becomes black, the distance between lips becomes larger and teeth are shown in the end as smiling, and pose turns from looking slightly front to looking right. Additionally we provide examples of transitions between $vector$("Man without eyeglass") and $vector$("Woman with eyeglass"), as well as $vector$("Man") and $vector$("Woman").

4.3.2. Facial attribute manipulation

The experiments above demonstrate interesting smooth transitional property between two latent vectors. In this section, instead of manipulating the overall face images, we seek to find a way to control a specific attribute of face images. In previous

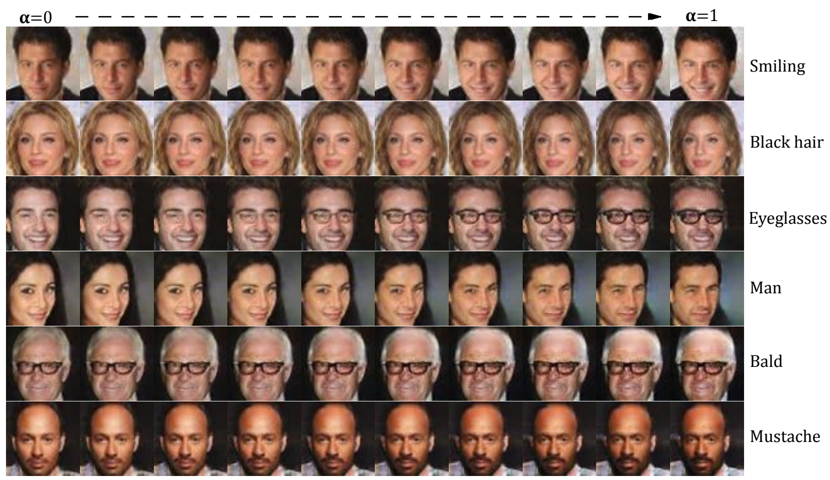

Figure 12: Vector arithmetic for visual attributes. Each row is the generated faces from latent vector z_{left} by adding or subtracting an attribute-specific vector, i.e., z_{left} + α $z_{smiling}$, where $\alpha = 0, 0.1, \ldots, 1$.

works, [56] shows that $vector$("King") - $vector$("Man") + $vector$("Woman") generates a vector whose nearest neighbor is the $vector$("Queen") when evaluating learned representation of words. [32] demonstrates that visual concepts such as face poses and gender could be manipulated by simple vector arithmetics.

In this paper, we conduct experiments to manipulate the facial attributes in the learned latent space of VAE-WGAN. For a given attribute such as *smiling*, 2,000 smiling face samples are fed into the trained encoder to generate 2,000 latent vectors. The average of these vectors forms the latent representation $z_{smiling+}$. Similarly, we use 2,000 non-smiling face samples to generate a non-smiling latent vector $z_{smiling-}$. Finally the difference $z_{smiling} = z_{smiling+} - z_{smiling-}$, which in effect takes away any non-smiling attributes from the smiling images, is used as the semantic representation for the attribute *smiling*. Similarly, we use the same approach to constructing other semantic attribute latent reconstructions for *Bald*, *Black hair*, *Eyeglass*, *Male* and *Mustache*. Thus, for a given image with latent vector z, we can manipulate the facial attribute with the corresponding attribute vector arithmetically, e.g. $z = z + \alpha z_{smiling}$. Figure 12 shows the results for the 6 attributes, i.e., *Bald*, *Black hair*, *Eyeglass*, *Male*, *Smiling*, and *Mustache*. As shown in Figure 12, by adding a smiling vector to the la-

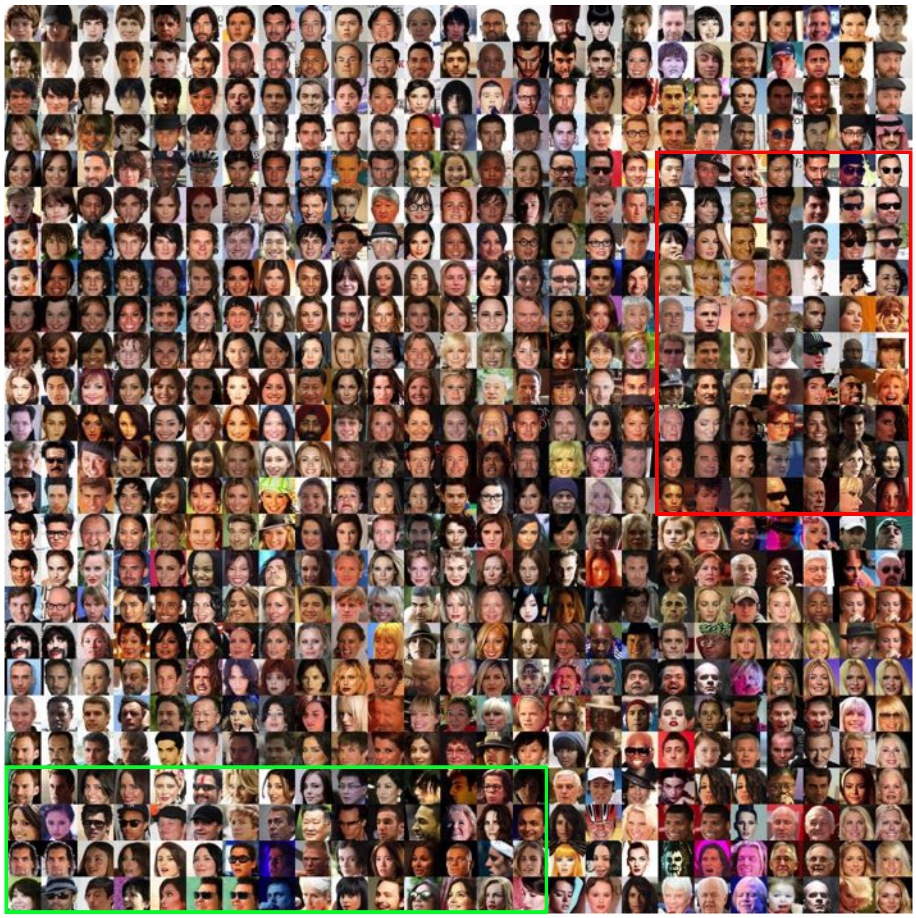

Figure 13: Visualization of 25 x 25 face images based on latent vectors by t-SNE algorithm [57].

tent representation of a non-smiling man, we can observe the smooth transitions from non-smiling face to smiling face (the first row). Furthermore, the smiling appearance becomes more obvious when the weighting factor α is bigger, while other facial attributes are able to remain unchanged. We can see that our method can achieve smooth image transitions for different facial attributes with high quality, demonstrating that the face attributes can be modeled linearly in the learned latent space.

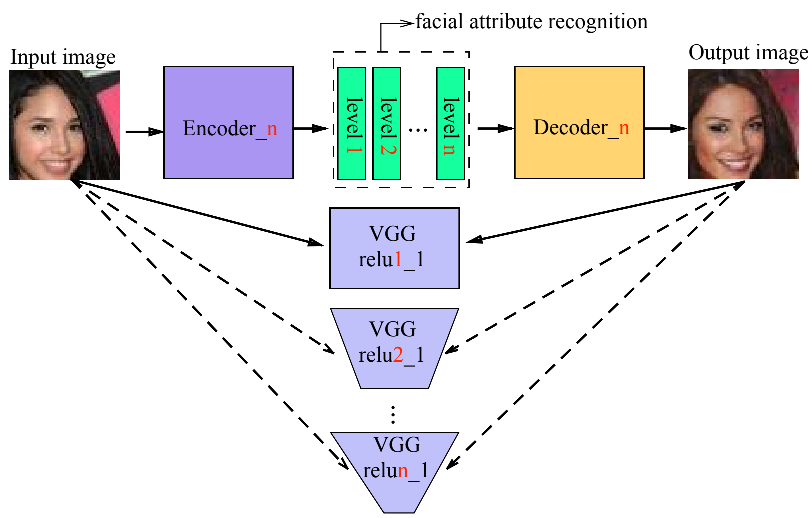

Figure 14: Multi-view feature extraction. 5 VAE-WGAN models are trained with feature reconstruction loss based on layers relu1_1, relu2_1 relu3_1, relu4_1 and relu5_1 respectively. The latent vectors for all the 5 models are concatenated as the final extracted features for facial attribute recognition.

4.3.3. Visualization of latent vectors

Considering that the latent vectors are nothing but the encoding representation of the natural face images, it would be interesting to visualize the natural face images based on the similarity of their latent representations. Specifically we randomly choose 625 face images from CelebA dataset and extract the corresponding 100-dimensional latent vectors, which are then reduced to 2-dimensional embedding by t-SNE algorithm [57]. t-SNE can arrange images that have similar high-dimensional vectors (L_2 distance) to be nearby each other in the embedding space. The visualization of 25×25 images is shown in Figure 13. We can see that images with a similar background (black or white) tend to be clustered together. Furthermore, the face pose information can be also captured even no pose annotations in the dataset. The face images in the upper right (red rectangle) are those looking to the left and samples in the bottom left (green rectangle) are those looking to the right.

Table 1: Performance comparison of 40 facial attributes recognition.

Method	5 Shadow	Arch. Eyebrows	Attractive	Bags Un. Eyes	Bald	Bangs	Big Lips	Big Nose	Black Hair	Blond Hair	Blurry	Brown Hair	Bushy Eyebrows	Chubby	Double Chin	Eyeglasses	Goatee	Gray Hair	Heavy Makeup	H. Cheekbones	Male
FaceTracer [58]	85	76	78	76	89	88	64	74	70	80	81	60	80	86	88	98	93	90	85	84	91
PANDA-w [59]	82	73	77	71	92	89	61	70	74	81	77	69	76	82	85	94	86	88	84	80	93
PANDA-l [59]	88	78	**81**	79	96	92	67	75	85	93	86	77	86	86	88	98	93	94	**90**	86	97
LNets+ANet [52]	**91**	79	**81**	79	**98**	**95**	68	78	**88**	**95**	84	80	**90**	91	92	**99**	**95**	**97**	**90**	**87**	**98**
VAE-123 [17]	89	77	75	81	**98**	91	76	79	83	92	**95**	80	87	94	95	96	94	96	85	81	90
VAE-345 [17]	89	**80**	78	**82**	**98**	**95**	**77**	**81**	85	93	**95**	80	88	94	**96**	**99**	**95**	**97**	89	85	95
VGG-FC[17]	83	71	68	73	97	81	51	77	78	88	94	67	81	93	93	95	93	94	79	64	84
VAE-WGAN (ours)	90	**80**	79	**82**	**98**	**95**	**77**	**81**	86	94	**95**	**82**	89	**95**	**96**	98	**95**	**97**	88	85	94

Method	Mouth S. O.	Mustache	Narrow Eyes	No Beard	Oval Face	Pale Skin	Pointy Nose	Reced. Hairline	Rosy Cheeks	Sideburns	Smiling	Straight Hair	Wavy Hair	Wear. Earrings	Wear. Hat	Wear. Lipstick	Wear. Necklace	Wear. Necktie	Young	Average
FaceTracer [58]	87	91	82	90	64	83	68	76	84	94	89	63	73	73	89	89	68	86	80	81.13
PANDA-w [59]	82	83	79	87	62	84	65	82	81	90	89	67	76	72	91	88	67	88	77	79.85
PANDA-l [59]	**93**	93	84	93	65	91	71	85	87	93	**92**	69	77	78	96	**93**	67	91	84	85.43
LNets+ANet [52]	92	95	81	**95**	66	91	72	89	90	**96**	**92**	73	**80**	82	**99**	**93**	71	**93**	**87**	87.30
VAE-123 [17]	80	**96**	**89**	88	73	96	73	**92**	**94**	95	87	79	74	82	96	88	**88**	**93**	81	86.95
VAE-345 [17]	88	**96**	**89**	91	**74**	96	**74**	**92**	**94**	**96**	91	**80**	79	84	98	91	**88**	**93**	84	88.73
VGG-FC[17]	60	93	87	84	66	96	58	86	93	85	65	68	70	49	98	82	87	89	74	79.85
VAE-WGAN (ours)	85	**96**	**89**	91	**74**	**97**	**74**	**92**	**94**	**96**	91	**80**	**80**	**85**	**99**	91	**88**	**93**	84	**88.88**

4.4. Facial attribute recognition

We further evaluate the quality of the learned latent representations of the VAE by applying them to facial attribute recognition, which is a very challenging problem. Like [52], 20,000 face images in the CelebA dataset [52] are used for testing while the remaining are used as training data. We proposed to use a multi-view strategy for feature extraction as shown in Figure 14. Specifically, 5 VAE-WGAN models are trained independently, each uses a different convolutional layer of the VGGNet to calculate the feature reconstruction loss. The latent vectors for all the 5 models are concatenated as the final extracted features which are used to train standard linear SVM classifiers to predict the 40 facial attributes in the dataset. As a result, we train 40 binary classifiers for each attribute in CelebA dataset respectively.

We then compare our method with other state of the art methods, i.e., FaceTracer [58], PANDA-w [59], PANDA-l [59], LNets+ANet [52],VAE-123 [17], VAE-345 [17]. From Table 1, It is seen that our method can achieve the highest average prediction accuracies, which slightly beats the state of the art results. Additionally, we find that our method is not always the best for all the facial attributes. In particular, it does not work very well to predict attributes like "Mouth S. O" (mouth slightly open) and "Wear Lipstick" as shown in Table 1. One possible explanation of this is that these attributes are hard to detect in face images and difficult to reconstruct precisely in variational autoencoder model. As a result, the encoded latent vectors are not able to capture such subtle differences.

5. Discussion

For variational autoencoder model, one essential part is to define a metric to measure the inconsistency between the input and the reconstructed output. The plain VAE adopts the per-pixel measurement, leading to unacceptably blurry outputs because it essentially treats images as "unstructured" input and each pixel is independent with all the other pixels. Inspired by the recent works like image style transfer [43, 44, 45], we propose to improve the performance of VAE by measuring the inconsistency in the deep feature space instead of naive pixel space. The hidden representations from pre-trained deep CNN are able to capture essential visual quality factors such as spatial correlation because of convolutional operations. What's more, variational autoencoder can be seamlessly incorporated into the framework of generative adversarial network to enforce the output to resemble natural images. The adversarial loss can be regarded as "structured" measurement because the GAN training is essentially performing high level image classification, and each pixel is not treated independently at all.

Another benefit of using deep CNNs to construct loss function is that we can achieve multi-scale modeling implicitly. Due to the hierarchy architecture of deep convolutional neural networks, a higher layer is corresponding to a coarser spatial area of the encoded image. Thus, unlike traditional methods that try to directly use multi-scale images as input, we can achieve another kind of multi-scale modeling by constructing loss function with different layers.

Another interesting part of VAE is the linear property in the learned latent space. Different images generated by the decoder can be smoothly transformed to others by a simple linear combination of their latent vectors. Additionally attribute specific features could be also calculated by encoding the annotated images and used to manipulate the related attribute of a given image while keeping other attributes unchanged.

6. Conclusion

In this paper, we propose a more stable architecture and several effective techniques to incorporate variational autoencoder. In particular, we employ deep feature consistent principle to allow the output to have a better perceptual quality and use adversarial training to help produce images that reside on the manifold of natural images. Compared to previous approaches, our model can generate more consistent and realistic images with fine details and reasonable backgrounds. In addition, we further investigate the quality of the learned representation to manipulate facial attributes. Finally, we have shown that our method can be used to extract effective representations for facial attribute recognition and achieve state of the art performance.

References

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354 – 377.

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[3] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (ICLR), 2015.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[5] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 584–599.

[6] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[7] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.

[8] J. Yu, C. Hong, Y. Rui, D. Tao, Multitask autoencoder model for recovering human poses, IEEE Transactions on Industrial Electronics 65 (6) (2018) 5060–5068.

[9] C. Hong, X. Chen, X. Wang, C. Tang, Hypergraph regularized autoencoder for image-based 3d human pose recovery, Signal Processing 124 (2016) 132–140.

[10] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Transactions on Industrial Electronics 62 (6) (2015) 3742–3751.

[11] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Transactions on Information Forensics and Security 12 (5) (2017) 1005–1016.

[12] J. Zhang, J. Yu, D. Tao, Local deep-feature alignment for unsupervised dimension reduction, IEEE Transactions on Image Processing 27 (5) (2018) 2420–2432.

[13] A. Liu, Y. Laili, Balance gate controlled deep neural network, Neurocomputing 320 (2018) 183–194.

[14] J. J. de Mesquita Sá Junior, A. R. Backes, O. M. Bruno, Randomized neural network based descriptors for shape classification, Neurocomputing 312 (2018) 201 – 209.

[15] V. Osipov, M. Osipova, Space–time signal binding in recurrent neural networks with controlled elements, Neurocomputing 308 (2018) 194 – 204.

[16] D. P. Kingma, M. Welling, Auto-encoding variational bayes, International Conference on Learning Representations (ICLR), 2014.

[17] X. Hou, L. Shen, K. Sun, G. Qiu, Deep feature consistent variational autoencoder, in: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, IEEE, 2017, pp. 1133–1141.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[19] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 214–223.

[20] M. Makkie, H. Huang, Y. Zhao, A. V. Vasilakos, T. Liu, Fast and scalable distributed deep convolutional autoencoder for fmri big data analytics, Neurocomputing 325 (2019) 20–30.

[21] J. Chen, Z. Wu, J. Zhang, F. Li, W. Li, Z. Wu, Cross-covariance regularized autoencoders for nonredundant sparse feature representation, Neurocomputing 316 (2018) 49–58.

[22] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, Y. Jin, Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection, Neurocomputing.

[23] S. Feng, M. F. Duarte, Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation, Neurocomputing 312 (2018) 310 – 323.

[24] K. Sun, J. Zhang, C. Zhang, J. Hu, Generalized extreme learning machine autoencoder and a new deep neural network, Neurocomputing 230 (2017) 374–381.

[25] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Transactions on Image Processing 24 (12) (2015) 5659–5670.

[26] D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1278–1286.

[27] D. P. Kingma, S. Mohamed, D. J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.

[28] X. Yan, J. Yang, K. Sohn, H. Lee, Attribute2image: Conditional image generation from visual attributes, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, 2016, pp. 776–791.

[29] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra, DRAW: A recurrent neural network for image generation, in: ICML, Vol. 37 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015, pp. 1462–1471.

[30] K. Ridgeway, J. Snell, B. Roads, R. Zemel, M. Mozer, Learning to generate images with perceptual similarity metrics, arXiv preprint arXiv:1511.06409.

[31] E. L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks, in: Advances in neural information processing systems, 2015, pp. 1486–1494.

[32] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.

[33] D. J. Im, C. D. Kim, H. Jiang, R. Memisevic, Generating images with recurrent adversarial networks, arXiv preprint arXiv:1602.05110.

[34] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 2226–2234.

[35] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 2172–2180.

[36] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, International Conference on Learning Representations (ICLR), 2018.

[37] Z. Hu, Z. Yang, R. Salakhutdinov, E. P. Xing, On unifying deep generative models, International Conference on Learning Representations (ICLR), 2018.

[38] X. Dong, Y. Yan, W. Ouyang, Y. Yang, Style aggregated network for facial landmark detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[39] X. Dong, J. Huang, Y. Yang, S. Yan, More is less: A more complicated network with less inference complexity, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 1895–1903.

[40] C. Wan, T. Probst, L. V. Gool, A. Yao, Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation, in: 2017 IEEE Conference

on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 1196–1205.

[41] G. Qi, Loss-sensitive generative adversarial networks on lipschitz densities, CoRR abs/1701.06264. arXiv:1701.06264.

[42] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 1558–1566.

[43] L. A. Gatys, A. S. Ecker, M. Bethge, A neural algorithm of artistic style, arXiv preprint arXiv:1508.06576.

[44] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II, 2016, pp. 694–711.

[45] D. Ulyanov, V. Lebedev, A. Vedaldi, V. S. Lempitsky, Texture networks: Feed-forward synthesis of textures and stylized images, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 1349–1357.

[46] C. Li, M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 2479–2486.

[47] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034.

[48] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, arXiv preprint arXiv:1506.06579.

[49] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus, Intriguing properties of neural networks, International Conference on Learning Representations (ICLR), 2014.

[50] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 427–436.

[51] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[52] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.

[53] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Master's thesis, Department of Computer Science, University of Toronto.

[54] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR), 2015.

[55] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: A matlab-like environment for machine learning, in: BigLearn, NIPS Workshop, no. EPFL-CONF-192376, 2011.

[56] T. Mikolov, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, 2013.

[57] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

[58] N. Kumar, P. Belhumeur, S. Nayar, Facetracer: A search engine for large collections of images with faces, in: European conference on computer vision, Springer, 2008, pp. 340–353.

[59] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L. Bourdev, Panda: Pose aligned networks for deep attribute modeling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1637–1644.