

Feature Perceptual Loss for Variational Autoencoder

Xianxu Hou

University of Nottingham, Ningbo, China

xianxu.hou@nottingham.edu.cn

Linlin Shen

Shenzhen University, Shenzhen, China

llshen@szu.edu.cn

Ke Sun

University of Nottingham, Ningbo, China

ke.sun@nottingham.edu.cn

Guoping Qiu

University of Nottingham, Ningbo, China

guoping.qiu@nottingham.edu.cn

Abstract

We consider unsupervised learning problem to generate images like Variational Autoencoder (VAE) and Generative Adversarial Network (GAN), which are two popular generative models around this problem. Recent works on style transfer have shown that higher quality images can be generated by optimizing feature perceptual loss, which is based on pretrained deep convolutional neural network (CNN). We propose to train VAE by using feature perceptual loss to measure the similarity between the input and generated images instead of pixel-by-pixel loss. Testing on face image dataset, our model can produce better qualitative results than other models. Moreover, our experiments demonstrate that the learned latent representation in our model has powerful capability to capture the conceptual and semantic information of natural images, and achieve state-of-the-art performance in facial attribute prediction.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have been used to achieve state-of-the-art performances in many supervised computer vision tasks such as image classification [13, 28], retrieval [1], detection [5, ?], and captioning [9, ?]. Deep CNNs-based generative models, a branch of unsupervised learning techniques in machine learning, have become a hot research topic in computer vision area in recent years. A generative model trained with a given dataset can be used to generate data like the samples in the dataset, learn the internal essence of the dataset and "store" all the information in the limited parameters that are significantly smaller than the training dataset.

Variational Autoencoder (VAE) [12, 24] has become a popular generative model, allowing us to formalize this problem in the framework of probabilistic graphical models with latent variables. By default, pixel-by-pixel mea-

surement like L2 loss, or logistic regression loss is used to measure the difference between reconstructed and original images. Such measurements are easily implemented and effective for deep neural network training. However, the generated images are not clear and tend to be very blurry when compared to natural images. This is because the pixel-by-pixel loss is not good enough to capture the visual perceptual difference between two images and it is not the way how humans look at the world. For example, the same image offsetted by a few pixels has little visual perceptual difference for humans, but it could have very high pixel-by-pixel loss.

In this paper, we try to improve the standard (plain) VAE by replacing the pixel-by-pixel loss with feature perceptual loss which is the difference between high level features of images extracted from hidden layer in pretrained deep convolutional neural networks such as AlexNet [13] and VGGNet [28] trained on ImageNet [26]. The high-level feature-based loss has been successfully applied to deep neural network visualization [27, 31], texture synthesis and style transfer [4, 3], demonstrating superiority over pixel-by-pixel loss. We also explore the conceptual representation capability of the learned latent space, and use it for facial attribute prediction.

2. Related Work

Variational Autoencoder (VAE). A VAE [12] helps us to do two things. Firstly it allows us to encode an image x to a small dimension latent vector $z = \text{Encoder}(x) \sim q(z|x)$ with an encoder network, and then an decoder network is used to decode the latent vector z back to an image that will be as similar as the original image $\bar{x} = \text{Decoder}(z) \sim p(x|z)$. That is to say, we need to maximize marginal log-likelihood of each observation (pixel) in x , and the VAE reconstruction loss \mathcal{L}_{rec} is negative expected log-likelihood of observations in x . Another important property of VAE is able to control the distribution of latent vector z , which

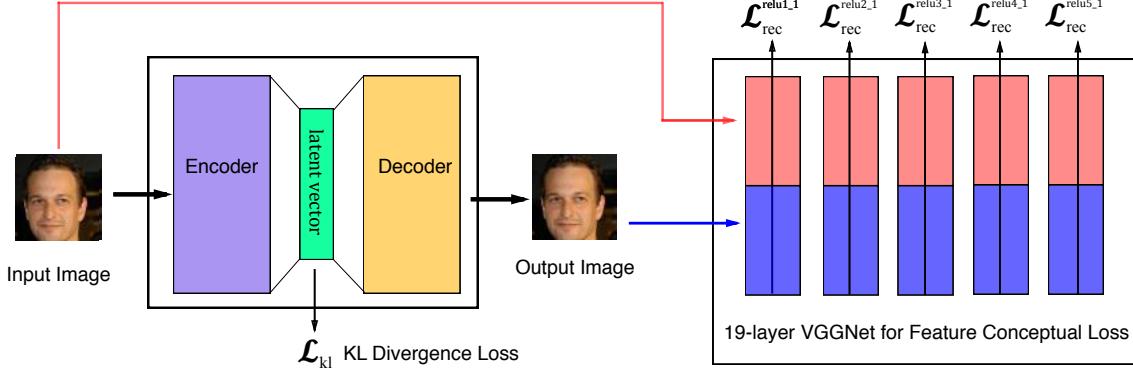


Figure 1. Model Overview. The left is a deep CNN-based Variational Autoencoder, and the right is a pretrained deep CNN used to compute feature perceptual loss.

has characteristic of being independent unit Gaussian random variables, 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 using gradient descent algorithm [12]. Therefore, VAE models can be trained by optimizing the sum of the reconstruction loss (\mathcal{L}_{rec}) and KL divergence loss (\mathcal{L}_{kl}) using gradient descent.

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

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

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

Several methods have been proposed to improve the performance of VAE. [11] extends the variational auto-encoders to semi-supervised learning with class labels, [30] proposes a variety of attribute-conditioned deep variational auto-encoders, and demonstrates that they are capable of generating realistic faces with diverse appearance, Deep Recurrent Attentive Writer (DRAW) [7] combines spatial attention mechanism with a sequential variational auto-encoding framework that allows iterative generation of images. Considering the shortcoming of pixel-by-pixel loss, [25] replaces pixel-by-pixel loss with multi-scale structural-similarity score (MS-SSIM) and demonstrates that it can better measure human perceptual judgments of image quality. [15] proposes to enhance the objective function with discriminative regularization. Another approach [16] tries to combine VAE and generative adversarial network (GAN) [23, 6], and use the learned feature representation in the GAN discriminator as basis for the VAE reconstruction objective.

high-level feature perceptual loss. Several recent papers successfully generate images by optimizing perceptual

loss, which is based on the high-level features extracted from pretrained deep convolutional neural networks. Neural style transfer [4] and texture synthesis [3] tries to jointly minimize high-level feature reconstruction loss and style reconstruction loss by optimization. Additionally images can be also generated by maximizing classification scores or individual features [27, 31]. Other works try to train a feed-forward network for real-time style transfer [8, 29, 17] and super-resolution [8] based on feature perceptual loss. In this paper, we train a deep convolutional variational autoencoder (CVAE) for image generation by replacing pixel-by-pixel reconstruction loss with high-level feature perceptual loss based on pre-trained network.

3. Method

Our system consists two main components as shown in Figure 1: an autoencoder network including an encoder network($E(x)$) and a decoder network($D(z)$), and a loss network (Φ) that is a pretrained deep convolutional neural network to define feature perceptual loss. An input image x is encoded as a latent vector $z = E(x)$, which will be decoded ($\bar{x} = D(z)$) back to image space. After training, new image can be generated by decoder network with a given vector z . In order to train a VAE, we need two loss functions, one is KL divergence loss ($\mathcal{L}_{kl} = D_{kl}(q(z|x)||p(z))$) [12] which is used to make sure that the latent vector z is an independent unit Gaussian random variable. The other is feature reconstruction loss. Instead of direct comparing the input image and the generated image in the pixel space, we pass both of them to a pre-trained deep convolutional neural network Φ respectively and then measure the difference between hidden layer representation, i.e., $\mathcal{L}_{rec} = \mathcal{L}^1 + \mathcal{L}^2 + \dots + \mathcal{L}^l$, where \mathcal{L}^t represents the feature loss at the t^{th} hidden layer. Thus, we use the high-level

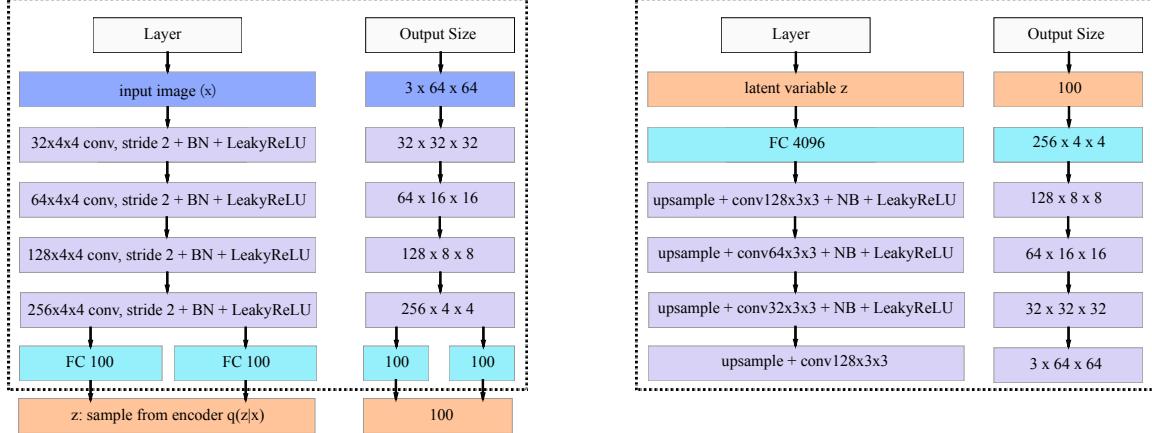


Figure 2. Autoencoder network architecture. The left is encoder network, and the right is decoder network.

feature loss to better measure perceptual and semantic differences between the two images, this is because the pre-trained network on image classification has already incorporated perceptual and semantic information we desire for. During the training, the pretrained loss network is fixed and just for high-level feature extraction, and KL divergence loss \mathcal{L}_{kl} is just used to update encoder network while the reconstruction feature loss \mathcal{L}_{rec} is responsible for updating parameters of both encoder and decoder.

3.1. Variational Autoencoder Network Architecture

Both encoder and decoder network are based on deep convolutional neural network (CNN) like AlexNet [13] and VGGNet [28]. We construct 4 convolutional layers in encoder network with 4 x 4 kernels, and the stride is fixed to be 2 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. Then two fully-connected output layers (for mean and variance) are added to encoder, and will be used to compute the KL divergence loss and sample latent variable z (see [12, ?] for details). For decoder, we use 4 convolutional layers with 3 x 3 kernels and set stride to be 1, and replace standard zero-padding with replication padding, i.e., feature map of an input is padded with the replication of the input boundary. For upsampling we use nearest neighbor method by scale of 2 instead of fractional-strided convolutions used by other works [19, 23]. We also use batch normalization to help stabilize training and use LeakyReLU as activation function. The details of autoencoder network architecture is shown in Figure 2.

3.2. Feature Perceptual Loss

Feature perceptual loss of two images is defined as the difference between the hidden features in a pretrained deep

convolutional neural network Φ . Similar to [4], we use VGGNet [28] as the loss network in our experiment, which is trained for classification problem on ImageNet dataset. The core idea of feature perceptual loss is to seek the similarity between the hidden representation of two images, and the input images tend to be similar from perceptual and semantic aspect if the difference of hidden representation is small. Specifically, let $\Phi(x)^l$ denote the representation of a l^{th} hidden layer when input image x is fed to network Φ . Mathematically $\Phi(x)^l$ is a 3D volume block array of shape $[C^l \times W^l \times H^l]$, where C^l is the number of filters, W^l and H^l represent the width and height of each feature map for the l^{th} layer. The feature perceptual loss for one layer (\mathcal{L}_{rec}^l) between two images x and \bar{x} can be simply defined by squared euclidean distance. Actually it is quite like pixel-by-pixel loss for images except that the color channel is not 3 any more.

$$\mathcal{L}_{rec}^l = \frac{1}{2C^l W^l H^l} \sum_{c=1}^{C^l} \sum_{w=1}^{W^l} \sum_{h=1}^{H^l} (\Phi(x)_c^l - \Phi(\bar{x})_c^l)^2$$

By optimization to reconstruct images from noise, [4, 8] show that reconstruction from lower layers is almost perfect. While using higher layers, pixel information such as color and shape are changed although overall spatial structures can be preserved. In our paper, our reconstruction loss is defined as the total loss at different layers of VGG Network, i.e., $\mathcal{L}_{rec} = \sum_l \mathcal{L}_{rec}^l$. Additionally we adopt the KL divergence loss \mathcal{L}_{kl} [12] to regularize the encoder network to control the distribution of latent variable z . To train VAE, we jointly minimize the KL divergence loss \mathcal{L}_{kl} and feature perceptual loss \mathcal{L}_{rec}^l for different layers, i.e.,

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{kl} + \beta \sum_i^l (\mathcal{L}_{rec}^l)$$

where α and β are weighted parameters for KL Diver-

gence and image reconstruction. It is quite similar to style transfer [4] if we treat KL Divergence as style reconstruction.

4. Experiments

In this paper, we perform experiments on face images to test our method. Specifically we compare the performance of our model trained by high-level feature perceptual loss with other generative models. Furthermore, we also investigate the latent space to seek semantic relationship between different latent representation and apply it to facial attribute prediction.

4.1. Training Details

Our model is trained on CelebFaces Attributes (CelebA) Dataset [18]. CelebA is a large-scale face attributes dataset with 202,599 number of face images, and 5 landmark locations, 40 binary attributes annotations per image. We build the training dataset by cropping and scaling the alignment images to 64 x 64 pixels like [16, 23]. We train our model with a batch size of 64 for 5 epochs over the training dataset and use Adam method for optimization [10] with initial learning rate of 0.0005, which is decreased by 0.5 for the following epochs. The 19-layer VGGNet [28] is chosen as loss network Φ to construct feature perceptual loss for image reconstruction. We experiment with different layer combinations to construct feature perceptual loss and report the results by using layers relu1_2 , relu2_1 , relu3_1 . In addition, the dimension of latent vector z is set to be 100, and the loss weighted parameters α and β are 1 and 0.8 respectively. Our implementation is built on deep learning framework Torch [2] and style transfer implementation [?].

4.2. Qualitative Results for Image Generation

In this paper, we also train additional two generative models for comparison. One is the plain Variational Autoencoder (PVAE), which has the same architecture as our proposed model, but trained with pixel-by-pixel loss in the image space. The other is Deep Convolutional Generative Adversarial Networks (DCGAN) consisting of a generator and a discriminator network [23], which has shown the ability to generate high quality images from a noise vector. DCGAN is trained with open source code [23] in Torch. The comparison is divided into two parts: arbitrary face images generated by decoder based on latent vector z drawn from $\mathcal{N}(0, 1)$, and face image reconstruction.

In the first part, random face images (shown in Figure 3) are generated by three models from latent vector z drawn from $\mathcal{N}(0, 1)$. We can see that the generated face images by plain VAE tend to be very blurry, even though overall spatial face structure can be preserved. It is very hard for plain VAE to generate clear facial parts such as eyes and noses,

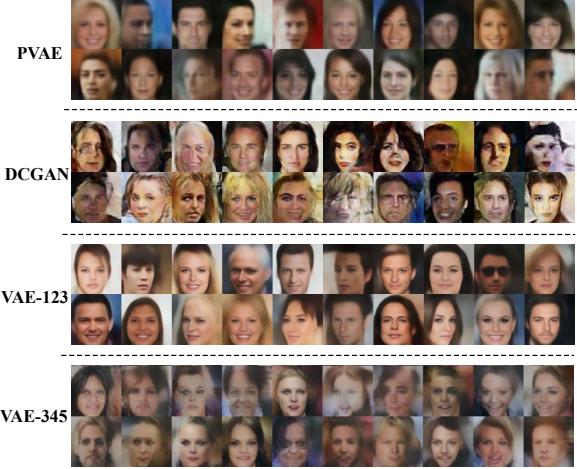


Figure 3. Generated fake face images from 100-dimension latent vector $z \sim \mathcal{N}(0, 1)$ from different models. The first part is generated from decoder network of plain variational autoencoder (PVAE) trained with pixel-based loss [12], the second part is generated from generator network of DCGAN [23], and the third part is our method trained with feature perceptual loss.

this is because it tries to minimize the reconstruction difference between two images with pixel-by-pixel loss. The pixel-based loss is problematic due to no semantic and perceptual information contained. DCGAN can generate clean and sharp face images containing clearer facial textures, however it has the facial distortion problem and sometimes generates weird faces. Our method based on feature perceptual loss can achieve better results, generating faces of different genders, ages and races with clear noses and eyes. What's more, face images with sunglasses and white clean teeth can be also randomly generated. One problem found in our method is that the generated hair tends to be blurry in most samples, and we think it is because of the subtle texture of human hair.

We also compare the reconstruction results (shown in Figure 4) between plain VAE and our method, and DCGAN is not compared because of no input image in their model. We can get similar conclusion as above between two methods. Even though the reconstruction is not perfect and the generated face images tend to be blurry when compared to input images, our method is much better than plain VAE.

4.3. Investigating Learned Latent Space

4.3.1 Linear interpolation of latent space

In order to get a better understanding of what our model has learned, we investigate the property of the z representation in the latent space from our encoder network, and the relationship between the different learned latent vectors.

As shown in Figure 5, we investigate the generated images from two latent vectors denoted as z_{left} and z_{right} .



Figure 4. Image reconstruction from different models. The first row is input image, the second row is generated from decoder network of plain variational autoencoder (PVAE) trained with pixel-based loss [12], and the last row is our method trained with feature perceptual loss.

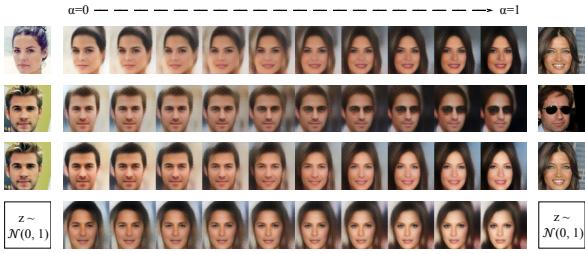


Figure 5. Linear interpolation for latent vector. Each row is the interpolation from left latent vector z_{left} to right latent vector z_{right} . e.g. $(1 - \alpha)z_{left} + \alpha z_{right}$. The first row is transitions from a non-smiling woman to a smiling woman, the second row is transitions from a man without sunglass to a man with sunglass, the third row is transitions from a man to a woman, and the last row is transitions between two fake faces decoded from $z \sim \mathcal{N}(0, 1)$.

The interpolation is defined by linear transformation $z = (1 - \alpha)z_{left} + \alpha z_{right}$, where $\alpha = 0, 0.1, \dots, 1$, and then z is fed to decoder network to generate new face images. In this paper, we provide three examples for latent vector z encoded from input images and one example for z randomly drawn from $\mathcal{N}(0, 1)$. From the first row in Figure 5, we can see the smooth transitions between *vector*("Woman without smiling and short hair") and *vector*("Woman with smiling and long hair"). Little by little the hair become longer, the distance between lips become larger and teeth is shown in the end for smiling, and pose turns from looking slightly left to looking front. Additionally we provide examples of transitions between *vector*("Man without sunglass") and *vector*("Man with sunglass"), and *vector*("Man") and *vector*("Woman").



Figure 6. Vector arithmetic for visual attributes. Each row is the generated faces from latent vector z_{left} by adding or subtracting an attribute-specific vector. e.g. $z_{left} + \alpha z_{smiling}$, where $\alpha = 0, 0.1, \dots, 1$. The first row is the transitions by adding a smiling vector with a linear factor α from left to right, the second row is the transitions by subtracting a smiling vector, the third and fourth row are the results by adding a sunglass vector to latent representation for a man and women, and the last row shows results by the subtracting a sunglass vector.

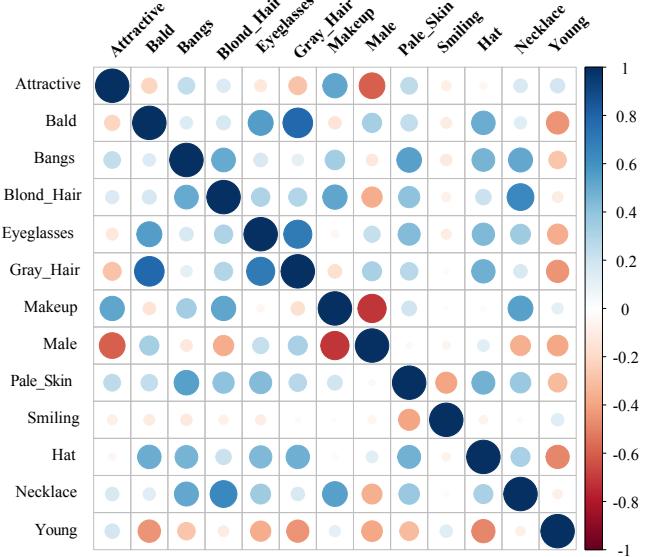


Figure 7. Diagram for the correlation between selected facial attribute-specific vectors. The blue indicates positive correlation, while red represents negative correlation, and the color shades and sizes of the circle represent the strength the correlation.

4.3.2 Facial attributes manipulation

The experiments above demonstrate interesting smooth transition's property between two learned latent vectors. In this part, instead of manipulating the overall face images, we seek to find a way to control a specific attribute of face images. In previous works, [21] shows that *vector*("King") - *vector*("Man") + *vector*("Woman") generates a vector whose nearest neighbor was the *vector*("Queen") when evaluating learned representation of words. [23] demon-

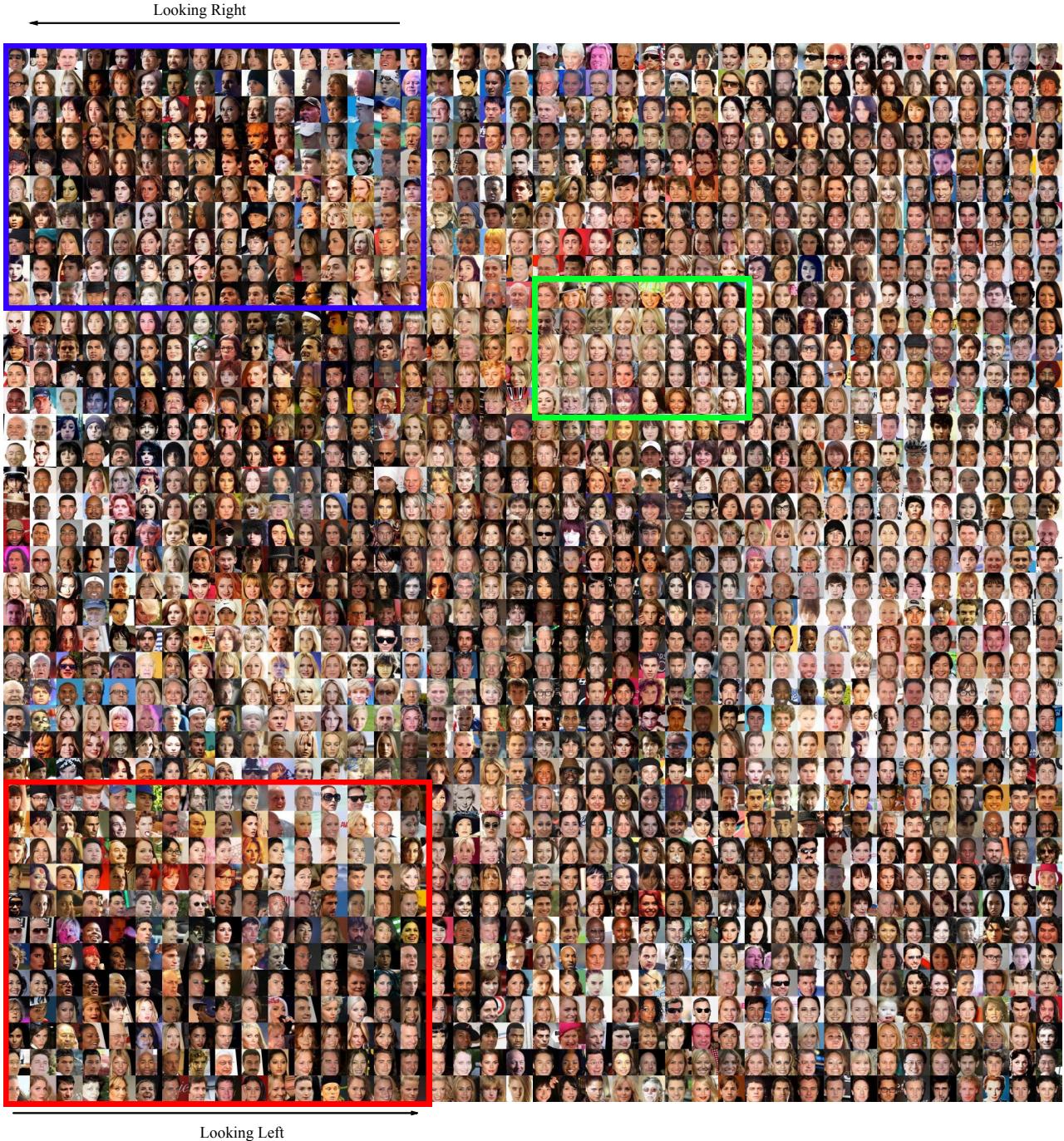


Figure 8. Visualization of 400 x 400 face images by latent vectors with t-SNE algorithm [20]

strates that visual concepts such as face pose and gender could be manipulated by simple vector arithmetic. In this paper, we investigate two facial attributes wearing sunglass and smiling. We randomly choose 1000 face images with sunglass and 1000 without sunglass respectively from the CelebA dataset [18], finally the two type of images are

fed to our encoder network to compute the latent vectors, and the mean latent vectors are calculated for each type respectively, denoted as $z_{pos_sunglass}$ and $z_{neg_sunglass}$. We then define the difference $z_{pos_sunglass} - z_{neg_sunglass}$ as sunglass-specific latent vector $z_{sunglass}$. In the same way, we calculate the smiling-specific latent vector $z_{smiling}$.

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	85	76	78	76	89	88	64	74	70	80	81	60	80	86	88	98	93	90	85	84	91
PANDA-w	82	73	77	71	92	89	61	70	74	81	77	69	76	82	85	94	86	88	84	80	93
PANDA-l	88	78	81	79	96	92	67	75	85	93	86	77	86	86	88	98	93	94	90	86	97
LNets+ANet	91	79	81	79	98	95	68	78	88	95	84	80	90	91	92	99	95	97	90	87	98
VAE-Z	89	77	75	81	98	91	76	79	83	92	95	80	90	91	92	99	95	97	90	87	98
VGG-FC	83	71	68	73	97	81	51	77	78	88	94	67	81	93	93	95	93	94	79	64	84
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	Young	Average		
FaceTracer	87	91	82	90	64	83	68	76	84	94	89	63	73	73	89	89	68	86	80	81.13	
PANDA-w	82	83	79	87	62	84	65	82	81	90	89	67	76	72	91	88	67	88	77	79.85	
PANDA-l	93	93	84	93	65	91	71	85	87	93	92	69	77	78	96	93	67	91	84	85.43	
LNets+ANet	92	95	81	95	66	91	72	89	90	96	92	73	80	82	99	93	71	93	87	87.30	
VAE-Z	80	96	89	88	73	96	73	92	94	95	87	79	74	82	96	88	88	93	81	86.95	
VGG-FC	60	93	87	84	66	96	58	86	93	85	65	68	70	49	98	82	87	89	74	79.85	

Table 1. Performance comparison of 40 facial attributes prediction. The accuracies of FaceTracer [14], PANDA-w [32], PANDA-l [32], and LNets+ANet [18] are collected from [18]. PANDA-l, VAE-Z and VGG-FC use the truth landmarks to get the face part.

Then we apply the two attribute-specific vectors to different latent vectors z by simple vector arithmetic, for instance, $z + \alpha z_{smiling}$. From Figure 6, by adding a smiling vector to the latent vector of a non-smiling man, we can observe the smooth transitions from non-smiling face to smiling face (the first row). What’s more, the smiling appearance becomes more obvious when the factor α is bigger, while other facial attributes are able to remain unchanged. The other way round, when the latent vector of smiling woman is subtracted by the smiling vector, the smiling face can be translated to not smiling by only changing the shape of mouth (the second row in Figure 6). Moreover, we could add or wipe out a sunglass by playing with the calculated sunglass vector.

4.3.3 Correlation between attribute-specific vectors

Considering the conceptual relationship between different facial attributes in natural images, for instance, bald and gray hair are often related old people, we selected 13 of 40 attributes from CelebA dataset and calculate the attribute-specific vector respectively (the calculation is the same as calculating sunglass-specific vector above). We then visualize the correlation as shown in Figure 7, and the results are well consistent with human interpretation. We can see that *Attractive* has a strong positive correlation with *Makeup*, and a negative correlation with *Male* and *Gray Hair*. It makes sense that female is generally considered more attractive than male and uses a lot of makeup. Similarly, *Bald* has a positive correlation with *Gray Hair*

and *Eyeglasses*, and a negative correlation with *Young*. Additionally, *Smiling* seems to have no correlation with most of other attributes and only have a weak negative correlation with *Pale Skin*. It could be explained that *Smiling* is a very common human facial expression and it could have a good match with many other attributes.

4.3.4 Visualization of latent vectors

Considering that the latent vectors are nothing but the encoding representation of the natural face images, we think that it may be interesting to visualize the natural images based on the similarity of the latent representation in an unsupervised way. Specifically we randomly choose 1600 face images from CelebA dataset and extract the corresponding 100-dimensional latent vectors, which are then reduced to 2-dimensional embedding by using t-SNE algorithm [20]. t-SNE can arrange images that have a similar high-dimensional code (L2 distance) nearby in the embedding space. The visualization of 400 x 400 images is shown in Figure 8, and we can discover that images with similar background (black or white) tend to be clustered as a group, and female with smiling can be clustered together (green rectangle in Figure 8). What’s more, the face pose information can also be captured even no pose annotations in the dataset. The face images in the upper left (blue rectangle) tend to look left and samples in the lower left (red rectangle) tend to look right, while in other area tend to look front.

4.3.5 Facial attribute prediction

In the end, we evaluate our model by applying latent vector to facial attribute prediction, which is a very challenging problem due to complex face variations. Similar to [18], 20,000 images from CelebA dataset are selected for testing and the rest for training. Firstly we use ground truth landmark points to crop out the face parts of the original images like PANDA-1 [32], and the cropped face images are fed to our encoder network to extract latent vectors, which are then used to train standard Linear SVM [22] classifiers. As a result, we train 40 binary classifiers for each attribute in CelebA dataset respectively. As a baseline, we also train different Linear SVM classifiers with 4096-dimensional deep features extracted from the last fully connected layer of pretrained VGGNet [28]. We then compare our method with other state-of-the-art methods. The average of prediction accuracies of FaceTracer [14], PANDA-w [32], PANDA-1 [32], and LNets+ANet [18] are 81.13, 79.85, 85.43 and 87.30 percent respectively. Our method with latent vector of VAE (VAE-Z) and VGG last layer features (VGG-FC) are 86.95 and 79.85 respectively. From Table 1, we can see that our method is comparable to the LNets+ANet and outperforms other methods. Our method can do a better job to predict *Wearing_Necklace*, *Receding_Hairline* and *Pale_Skin*. In addition, we notice that all the methods can achieve a good performance to predict *Bald*, *Wearing_Hat* and *Eyeglasses*, while they are very difficult to correctly predict attributes like *Big_Lips* and *Oval_Face*. The reason we think is that attributes like whether wearing hat and eyeglasses or not are much more obvious in natural face images, than attributes whether having big lips and Oval face or not, and the extracted features are not able to capture such subtle differences. Future work is needed to find a way to extract better features which can also capture tiny variation of facial attributes.

4.4. Discussion

For (variational) autoencoder models, one essential part is to define a reconstruction loss to measure the similar between input image and generated image. The plain VAE adopts the pixel-by-pixel distance, which is problematic and the generated images tend to be very blurry. Inspired by the state-of-the-art works on style transfer and texture synthesis [4, 8, 29], we measure the reconstruction loss in VAE by feature perceptual loss based on pretrained deep convolutional neural networks (CNNs). Our experiments above have shown that feature perceptual loss can be used to improve the performance of VAE to generate high quality images. One explanation is that the hidden representation in a pretrained deep CNN could capture conceptual and semantic information of a given image since it has the ability to do classification, which is a human understanding task. An-

other benefit of using deep CNNs is that we can combine different level of hidden representation, which can provide more constraints for the reconstruction. Actually we could explore different combinations even add weights to different level representation to generate weird but interesting images. However, the feature perceptual loss is not perfect, the trained model fails to generate clear hair texture in our experiments even though it can do a good job for eyes, noses and mouths generation. For further work, trying to construct better reconstruction loss to measure the similarity of the output images and ground-truth images is essential for this problem. One possibility is to combine feature perceptual loss with generative adversarial networks(GAN).

The more interesting part of VAE is the linear structure in the learned latent space. Different images generated by decoder can be smoothly transformed to each other by simply linear combination of their latent vectors. Additionally attribute-specific latent vectors 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, what's more, the correlation between attribute-specific vectors is well consistent with human understanding. Our experiments shows that the learned latent space of VAE can learn powerful representation of conceptual and semantic information of natural images, and it could be used for other applications like face attribute prediction.

5. Conclusion

In this paper, we try to improve the performance of image generation of VAE by combining feature perceptual loss based on pretrained deep CNNs to measure the similar of two images. We apply our model on face images and achieve comparable and better performance compared to different generative models (plain VAE and GAN). In addition, we fully explore the learned latent representation in our model and demonstrates it has powerful capability to capture the conceptual and semantic information of natural images. We also achieved state-of-the-art performance of facial attribute prediction based on the learned latent representation.

References

- [1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV 2014*, pages 584–599. Springer, 2014.
- [2] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.
- [3] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

- [4] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [7] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.
- [8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.
- [9] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [11] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
- [12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [14] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *European conference on computer vision*, pages 340–353. Springer, 2008.
- [15] A. Lamb, V. Dumoulin, and A. Courville. Discriminative regularization for generative models. *arXiv preprint arXiv:1602.03220*, 2016.
- [16] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.
- [17] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. *arXiv preprint arXiv:1601.04589*, 2016.
- [18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.
- [19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [20] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [21] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
- [22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [24] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286, 2014.
- [25] K. Ridgeway, J. Snell, B. Roads, R. Zemel, and M. Mozer. Learning to generate images with perceptual similarity metrics. *arXiv preprint arXiv:1511.06409*, 2015.
- [26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [27] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [29] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016.
- [30] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.
- [31] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.
- [32] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1644, 2014.