CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training

Jianmin Bao[1],   Dong Chen[2],   Fang Wen[2],   Houqiang Li[1],   Gang Hua[2]

[1]University of Science and Technology of China   [2]Microsoft Research

jmbao@mail.ustc.edu.cn   {doch, fangwen, ganghua}@microsoft.com   lihq@ustc.edu.cn

Abstract

We present variational generative adversarial networks, a general learning framework that combines a variational auto-encoder with a generative adversarial network, for synthesizing images in fine-grained categories, such as faces of a specific person or objects in a category. Our approach models an image as a composition of label and latent attributes in a probabilistic model. By varying the fine-grained category label fed into the resulting generative model, we can generate images in a specific category with randomly drawn values on a latent attribute vector. Our approach has two novel aspects. First, we adopt a cross entropy loss for the discriminative and classifier network, but a mean discrepancy objective for the generative network. This kind of asymmetric loss function makes the GAN training more stable. Second, we adopt an encoder network to learn the relationship between the latent space and the real image space, and use pairwise feature matching to keep the structure of generated images. We experiment with natural images of faces, flowers, and birds, and demonstrate that the proposed models are capable of generating realistic and diverse samples with fine-grained category labels. We further show that our models can be applied to other tasks, such as image inpainting, super-resolution, and data augmentation for training better face recognition models.

1. Introduction

Building effective generative models of natural images is one of the key problems in computer vision. It aims to generate diverse realistic images by varying some latent parameters according to the underlying natural image distributions. Therefore, a desired generative model is necessitated to capture the underlying data distribution. This is often a very difficult task, since a collection of image samples may lie on a very complex manifold. Nevertheless, recent advances in deep convolutional neural networks have spawned a series of deep generative models [14, 12, 8, 31, 29, 34, 15, 4, 33, 6] that have made tremendous progress, largely due to the capability of deep networks in learning representations.

Building on top of the success of these recent works, we want to go one step further to generate images of
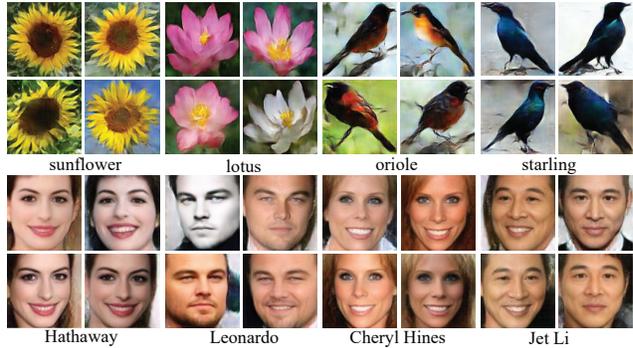

Figure 1. **Synthesized images** using our CVAE-GAN model at high resolution (128×128) for different classes. The generated samples are realistic and diverse within a class.

fine-grained object categories. For example, we want to be able to synthesize images for a specific identity (Figure 1), or produce a new image of a specified species of flowers or birds, and so on. Inspired by CVAE [34] and VAE/GAN [15], we propose a general learning framework that combines a variational auto-encoder with a generative adversarial network under a conditioned generative process to tackle this problem.

However, we find this naïve combination insufficient in practice. The results from VAE are usually blurry. The discriminator can easily classify them as "fake", even though they sometimes look remarkably good for face images, the gradient vanishing problem still exists. Thus, the generated images are very similar to the results from using VAE alone.

In this paper, we propose a new objective for the generator. Instead of using the same cross entropy loss as the discriminator network, the new objective requires the generator to generate data that minimize the ℓ_2 distance of the mean feature to the real data. For multi-class image generation, the generated samples of one category also need to match the average feature of real data of that category, since the feature distance and the separability are positively correlated. It solves the gradient vanishing problem to a certain extent. This kind of asymmetric loss function can partially help prevent the mode collapse problem that all outputs moving toward a single point, making the training of GAN more stable.

Although using mean feature matching will reduce the chance of mode collapse, it does not completely solve the

problem. Once mode collapse occurs, the gradient descent is unable to separate identical outputs. To keep the diversity of generated samples, we take advantage of the combination of VAE and GAN. We use an encoder network to map the real image to the latent vector. Then the generator is required to reconstruct the raw pixels and match the feature of original images with a given latent vector. In this way, we explicitly set up the relationship between the latent space and real image space. Because of the existence of these anchor points, the generator is enforced to emit diverse samples. Moreover, the pixel reconstruction loss is also helpful for maintaining the structure, such as a straight line or a facial structure in an image.

As shown in Figure 2 (g), our framework consists of four parts: 1) The encoder network E, which maps the data sample x to a latent representation z. 2) The generative network G, which generates image x' given a latent vector. 3) The discriminative network D, which distinguishes real/fake images. 4) The classifier network C, which measures the class probability of the data. These four parts are seamlessly cascaded together, and the whole pipeline is trained end-to-end. We call our approach CVAE-GAN.

Once the CVAE-GAN is trained, it can be used in different applications, *e.g.*, image generation, image inpainting, and attributes morphing. Our approach estimates a good representation of the input image, and the generated image appears to be more realistic. We show that it outperforms CVAE, CGAN, and other state-of-the-art methods. Compared with GAN, the proposed framework is much easier to train and converges faster and more stable in the training stage. In our experiments, we further show that the images synthesized from our models can be applied to other tasks, such as data augmentation for training better face recognition models.

2. Related work

Conventional wisdom and early research of generative models, including Principle Component Analysis (PCA) [40], Independent Component Analysis (ICA) [10],and the Gaussian Mixture Model (GMM) [46, 27, 37], all assume a simple formation of data. They have difficulty modeling complex patterns of irregular distributions. Later works, such as the Hidden Markov Model (HMM) [35], Markov Random Field (MRF) [19], and restricted Boltzmann machines (RBMs) [9, 32], discriminatively train generative models [39], limiting their results on texture patches, digital numbers or well aligned faces, due to a lack of effective feature representations.

There have been many recent developments of deep generative models [14, 12, 8, 31, 29, 15, 4, 33, 6]. Since deep hierarchical architectures allow them to capture complex structures in the data, all these methods show promising results in generating natural images that are far more realis-
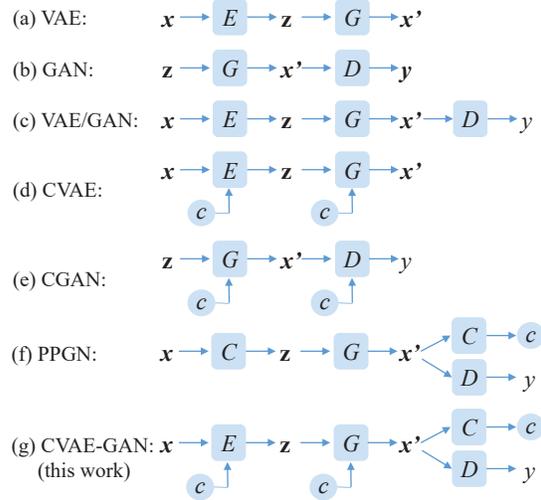

Figure 2. Illustration of the structure of VAE [12, 31], GAN [8], VAE/GAN [15], CVAE [34], CGAN [18], PPGN [23] and the proposed CVAE-GAN. Where x and x' are input and generated image. E,G,C,D are encoder, generative, classification, and discriminative network, respectively. z is the latent vector. y is a binary output which represents real/synthesized image. c is the condition, such as attribute or class label.

tic than conventional generative models. Among them are three main themes: Variational Auto-encoder (VAE) [12, 31], Generative Adversarial Network (GAN) [8, 29, 33], and Autoregression [14].

VAE [12, 31] pairs a differentiable encoder network with a decoder/generative network. A disadvantage of VAE is that, because of the injected noise and imperfect element-wise measures such as the squared error, the generated samples are often blurry.

Generative Adversarial Network (GAN) [8, 29, 33] is another popular generative model. It simultaneously trains two models: a generative model to synthesize samples, and a discriminative model to differentiate between natural and synthesized samples. However, the GAN model is hard to converge in the training stage and the samples generated from GAN are often far from natural. Recently, many works have tried to improve the quality of the generated samples. For example, the Wasserstein GAN (WGAN) [2] uses Earth Mover Distance as an objective for training GANs, and McGAN [20] uses mean and covariance feature matching. They need to limit the range of the parameters of the discriminator which will decrease discriminative power. Loss-Sensitive GAN [28] learns a loss function which can quantify the quality of generated samples and uses this loss function to generate high-quality images. There are also methods which tried to combine GAN and VAE, *e.g.*, VAE/GAN [15] and adversarial autoencoders [17]. They are closely related to and partly inspired our work.

VAEs and GANs can also be trained to conduct conditional generation, *e.g.*, CVAE [34] and CGAN [18].

By introducing additional conditionality, they can handle probabilistic one-to-many mapping problems. Recently there have been many interesting works based on CVAE and CGAN, including conditional face generation [7], Attribute2Image [47], text to image synthesis [30], forecasting from static images [42], and conditional image synthesis [25]. All of them achieve impressive results.

Generative ConvNet [44], demonstrates that a generative model can be derived from the commonly used discriminative ConvNet. Dosovitskiy et al. [5] and Nguyen et al. [22] introduce a method that generates high quality images from features extracted from a trained classification model. PPGN [23] performs exceptionally well in generating samples by using a gradient ascent and prior to the latent space of a generator.

Autoregression [14] follows a different idea. It uses autoregressive connections to model images pixel by pixel. Its two variants, PixelRNN [41] and PixelCNN [26], also produce excellent samples.

Our model differs from all these models. As illustrated in Figure 2, we compare the structure of the proposed CVAE-GAN with all these models. Besides the difference in the structure, more importantly, we take advantages of both statistic and pairwise feature matching to make the training process converge faster and more stable.

3. Our Formulation of CVAE-GAN

In this section, we introduce the proposed CVAE-GAN networks. As shown in Figure 3, our proposed method contains four parts: 1) the encoder network E; 2) the generative network G; 3) the discriminative network D; and 4) the classification network C.

The function of networks E and G is the same as that in conditional variational auto-encoder (CVAE) [34]. The encoder network E maps the data sample $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$ through a learned distribution $P(\boldsymbol{z}|\boldsymbol{x}, c)$, where c is the category of the data. The generative network G generates image $\boldsymbol{x}'$ by sampling from a learned distribution $P(\boldsymbol{x}|\boldsymbol{z}, c)$. The function of network G and D is the same as that in the generative adversarial network (GAN) [8]. The network G tries to learn the real data distribution by the gradients given by the discriminative network D which learns to distinguish between "real" and "fake" samples. The function of network C is to measure the posterior $P(c|\boldsymbol{x})$.

However, the naïve combination of VAE and GAN is insufficient. Recent work [1] shows that the training of GAN will suffer from a gradient vanishing or instability problem with network G. Therefore, we only keep the training process of networks E, D, and C the same as the original VAE [12] and GAN [8], and propose a new mean feature matching objective for the generative network G to improve the stability of the original GAN.

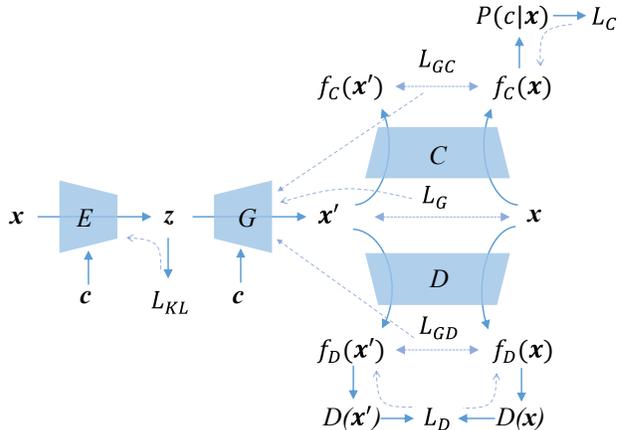Even with the mean feature matching objective, there is

Figure 3. Illustration of our network structure. Our model contains four parts: 1) The encoder network E; 2) The generative network G; 3) The classification network C; and 4) The discriminative network D. Please refer to Section 3 for details.

still some risk of mode collapse. So we use the encoder network E and the generative network G to obtain a mapping from real samples $\boldsymbol{x}$ to the synthesized samples $\boldsymbol{x}'$. By using the pixel-wise ℓ_2 loss and pair-wise feature matching, the generative model is enforced to emit diverse samples and generate structure-preserving samples.

In the following sections, we begin by describing the method of mean feature matching based GAN (Section 3.1). Then we show that the mean feature matching can also be used in conditional image generation tasks (Section 3.2). After that, we introduce pair-wise feature matching by using an additional encoder network (Section 3.3). Finally, we analyse the objective of the proposed method and provide the implementation details in the training pipeline (Section 3.4).

3.1. Mean feature matching based GAN

In traditional GANs, the generator G and a discriminator D compete in a two-player minimax game. The discriminator tries to distinguish real training data from synthesized data; and the generator tries to fool the discriminator. Concretely, the network D tries to minimize the loss function

$$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x} \sim P_r}[\log D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim P_z}[\log(1 - D(G(\boldsymbol{z})))], \tag{1}$$

while network G tries to minimize

$$\mathcal{L}'_{GD} = -\mathbb{E}_{\boldsymbol{z} \sim P_z}[\log D(G(\boldsymbol{z}))].$$

In practice, the distributions of "real" and "fake" images may not overlap with each other, especially at the early stage of the training process. Hence, the discriminative network D can separate them perfectly. That is, we always have $D(\boldsymbol{x}) \to 1$ and $D(\boldsymbol{x}') \to 0$, where $\boldsymbol{x}' = G(\boldsymbol{z})$ is the generated image. Therefore, when updating network G, the gradient $\partial \mathcal{L}'_{GD} / \partial D(\boldsymbol{x}') \to -\infty$. So the training process of

network G will be unstable. Recent works [1, 2] also theoretically show that training GAN often has to deal with the unstable gradient of G.

To address this problem, we propose using a mean feature matching objective for the generator. The objective requires the center of the features of the synthesized samples to match the center of the features of the real samples. Let $f_D(\boldsymbol{x})$ denote features on an intermediate layer of the discriminator, G then tries to minimize the loss function

$$\mathcal{L}_{GD} = \frac{1}{2}||\mathbb{E}_{\boldsymbol{x}\sim P_r}f_D(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_z}f_D(G(\boldsymbol{z}))||_2^2. \quad (2)$$

In our experiment, for simplicity, we choose the input of the last Fully Connected (FC) layer of network D as the feature f_D. Combining the features of multiple layers could marginally improve the converging speed. In the training stage, we estimate the mean feature using the data in a minibatch. We also use moving historical averages to make it more stable.

Therefore, in the training stage, we update network D using Eq. 1, and update network G using Eq. 2. Using this asymmetrical loss for training GAN has the following three advantages: 1) since Eq. 2 increases with the separability, the ℓ_2 loss on the feature center solves the gradient vanishing problem; 2) when the generated images are good enough, the mean feature matching loss becomes zero, making the training more stable; 3) compared with WGAN [2], we do not need to clip the parameters. The discriminative power of network D can be kept.

3.2. Mean Feature Matching for Conditional Image Generation

In this section, we introduce mean feature matching for conditional image generation. Supposing we have a set of data belonging to K categories, we use the network C to measure whether an image belongs to a specific fine-grained category. Here we use a standard method for classification. The network C takes in $\boldsymbol{x}$ as input and outputs a K-dimensional vector, which then turns into class probabilities using a softmax function. The output of each entry represents the posterior probability $P(c|\boldsymbol{x})$. In the training stage, the network C tries to minimize the softmax loss

$$\mathcal{L}_C = -\mathbb{E}_{\boldsymbol{x}\sim P_r}[\log P(c|\boldsymbol{x})]. \quad (3)$$

For the network G, if we still use the similar softmax loss function as in Eqn. 3, it will suffer from the same gradient instability problem as described in [1].

Therefore, we propose using the mean feature matching objective for generative network G. Let $f_C(\boldsymbol{x})$ denote features on an intermediate layer of the classification, then G tries to minimize:

$$\mathcal{L}_{GC} = \frac{1}{2}\sum_c ||\mathbb{E}_{\boldsymbol{x}\sim P_r}f_C(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim P_z}f_C(G(\boldsymbol{z}, c))||_2^2. \quad (4)$$

Here, we choose the input of the last FC layer of network C as the feature for simplicity. We also try to combine features of multiple layers, it only marginally improves the ability to preserve the identity of network G. Since there are only a few samples belonging to the same category in a minibatch, it is necessary to use moving averages of features for both real and generated samples.

3.3. Pairwise Feature Matching

Although, using mean feature matching could prevent all outputs from moving toward a single point, thus reducing the likelihood of mode collapse, it does not completely solve this problem. Once mode collapse occurs, the generative network outputs the same images for different latent vectors, thus the gradient descent will not be able to separate these identical outputs. Moreover, despite the generated samples and real samples having the same feature center, they may have different distributions.

In order to generate diverse samples, DCGAN [29] uses Batch Normalization, McGAN [20] uses both mean and covariance feature statistics, and Salimans *et al.* [33] use minibatch discrimination. They are all based on using multiple generated examples. Different from these methods, we add an encoder network E to obtain a mapping from the real image $\boldsymbol{x}$ to the latent space $\boldsymbol{z}$. Therefore, we explicitly set up the relationship between the latent space and real image space.

Similar to VAE, for each sample, the encoder network outputs the mean and covariance of the latent vector, i.e., μ and ϵ. We use the KL loss to reduces the gap between the prior $P(\boldsymbol{z})$ and the proposal distributions, i.e.,

$$\mathcal{L}_{KL} = \frac{1}{2}\left(\boldsymbol{\mu}^T\boldsymbol{\mu} + sum(\exp(\boldsymbol{\epsilon}) - \boldsymbol{\epsilon} - 1)\right). \quad (5)$$

We can then sample the latent vector $\boldsymbol{z} = \mu + \boldsymbol{r} \odot exp(\epsilon)$, where $\boldsymbol{r} \sim N(\boldsymbol{0}, \boldsymbol{I})$ is a random vector and $\odot$ represents the element-wise multiplication. After obtaining a mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$, we obtain the generated image $\boldsymbol{x}'$ with network G. Then, we add a ℓ_2 reconstruction loss and pair-wise feature matching loss between x and x',

$$\mathcal{L}_G = \frac{1}{2}(||\boldsymbol{x} - \boldsymbol{x}'||_2^2 + ||f_D(\boldsymbol{x}) - f_D(\boldsymbol{x}')||_2^2 \\ + ||f_C(\boldsymbol{x}) - f_C(\boldsymbol{x}')||_2^2), \quad (6)$$

where, f_D and f_C are the features of an intermediate layer of discriminative network D and classification network C, respectively.

3.4. Objective of CVAE-GAN

The goal of our approach is to minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_C + \lambda_1\mathcal{L}_{KL} + \lambda_2\mathcal{L}_G + \lambda_3\mathcal{L}_{GD} + \lambda_4\mathcal{L}_{GC}, \quad (7)$$
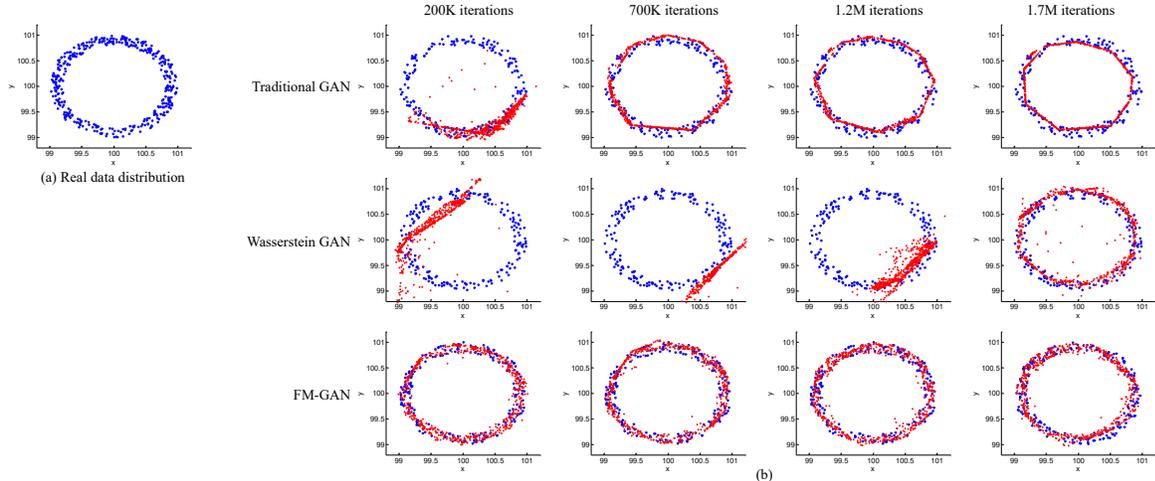
Figure 4. Results on a toy example for different generative models. The blue dots are the real points, the red dots are the generated points. a) The real data distribution which is like a "ring". b) The generated points by traditional GAN, WGAN and mean feature matching GAN at different iterations.

where the exact forms of each of the terms are presented in Eqns. 1∼6. Every term of the above formula is meaningful. $\mathcal{L}_{KL}$ is only related to the encoder network E. It represents whether the distribution of the latent vector is under expectation. $\mathcal{L}_G$, $\mathcal{L}_{GD}$ and $\mathcal{L}_{GC}$ are related to the generative network G. They represent whether the synthesized image is similar to the input training sample, the real image, and other samples within the same category, respectively. $\mathcal{L}_C$ is related to the classification network C, which represents the capability of the network to classify images from different categories, and $\mathcal{L}_D$ is related to the discriminative network, which represents how good the network is at distinguishing between real/synthesized images. All these objectives are complementary to each other, and ultimately enable our algorithm to obtain superior results. The whole training procedure is described in Algorithm 1. In our experiments. we empirically set $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 10^{-3}$ and $\lambda_4 = 10^{-3}$.

4. Analysis of Toy Example

In this section, we present and demonstrate the benefits of the mean feature matching based GAN with a toy example. We assume that we have a real data distribution which is a "ring" as shown in Figure 4(a). The center of the ring is set to $(100, 100)$, such that it is far from the generated distribution at the beginning. We compare the traditional GAN, WGAN, and the mean feature matching based GAN introduced in Section 3.1 to learn the real data distribution.

The three compared models share the same settings. Generator G is an MLP with 3 hidden layers with 32, 64, and 64 units, respectively. Discriminator D is also an MLP with 3 hidden layers with 32, 64, and 64 units, respectively. We use RMSProp and a fixed learning rate of 0.00005 for all methods. We trained each model for $2M$ iterations until they all converge. The generated samples of each model

at different iterations are plotted in Figure 4. From the results we can observe that: 1) For traditional GAN (first row in Figure 4(b)), the generated samples only lie in a limited area of the real data distribution, which is known as the mode collapse problem. This problem always exists during the training process. 2) For WGAN (second row in Figure 4(b)), it cannot learn the real data distribution at early iterations, we think this problem is caused by the clamping weights trick, which influence D's ability to distinguishing between real and fake samples. We also tried to vary the clamp values to accelerate the training process, and find that if the value is too small, it will cause a gradient vanishing problem. If too large, the network will diverge. 3) The third row shows the results of the proposed feature matching based GAN. It correctly learns the real data distribution the fastest.

5. Experiments

In this section, we use experiments to validate the effectiveness of the proposed method. We evaluate our model on three datasets: the FaceScrub [21],the 102 Category Flower [24], and CUB-200 [43] datasets. These three datasets contain three completely different objects, which are human faces, birds, and flowers, respectively.

The sizes of input and synthesized images are 128×128 for all experiments. For the FaceScrub dataset, we first detect the facial region with the JDA face detector [3], and then locate five facial landmarks (two eyes, nose tip and two mouth corners) with SDM [45]. After that, we use similarity transformation based on the facial landmarks to align faces to a canonical position. Finally, we crop a 128×128 face region centered around the nose tip. For the 102 Category Flower dataset, we tightly crop a rectangle region based on the ground-truth mask which contains the flower, and then resize it into 128×128. For the CUB-200 dataset, we just
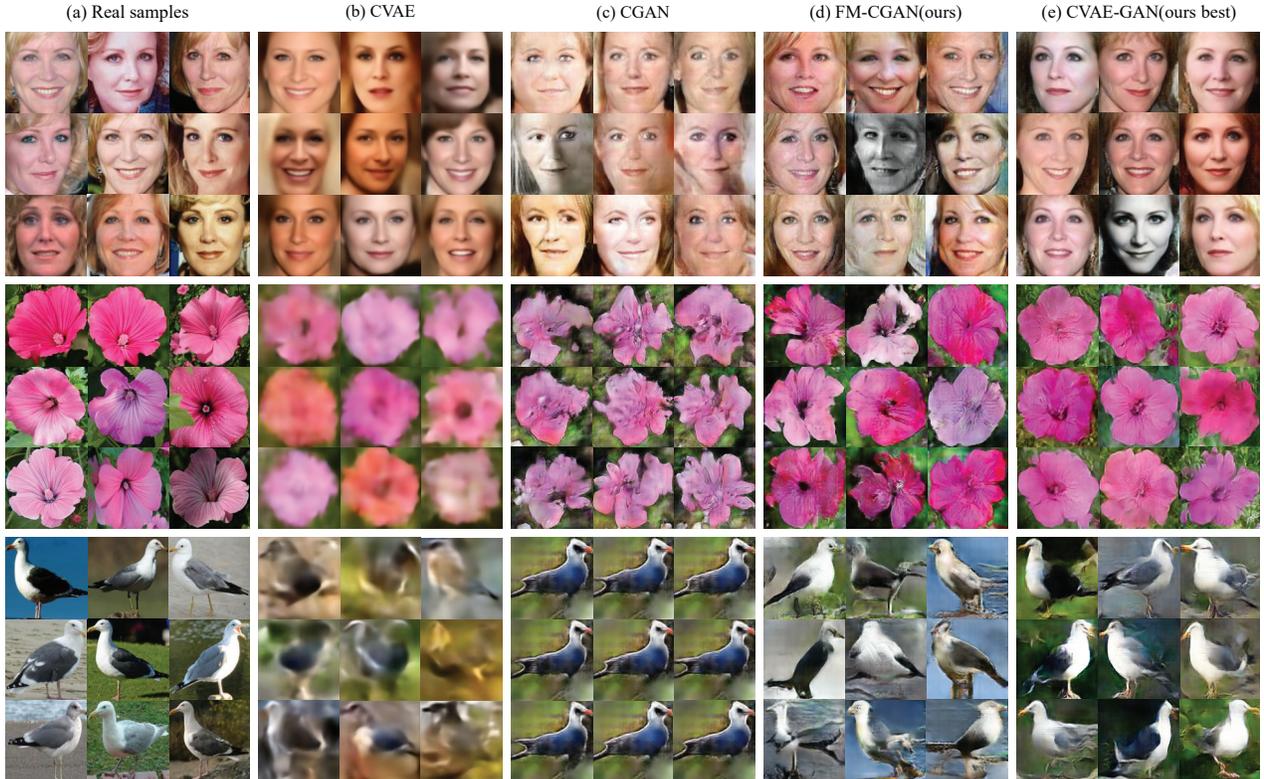
5

Figure 5. Comparison of randomly generated samples from different methods on FaceScrub [21], 102 Category Flower datasets [24] and CUB-200 [43] datasets. a) 9 random real images from one category. b) Results from CVAE, which is blurry and cannot preserve the category identity, c) Results from traditional CGAN, it loses diversity and structure info. d) Results from our mean feature matching CGAN, showing diverse results, but also lose of structure info. e) Results from our CVAE-GAN, which shows realistic, diversity and category-keeping results.

use the original images from the dataset.

In our experiments, the encoder network E is a GoogleNet [36], The category information and the image is merged at the last FC layer of the E network. The G network consists of 2 fully-connected layers, followed by 6 deconv layers with 2-by-2 upsampling. The convolution layers have 256, 256, 128, 92, 64 and 3 channels with filter size of 3×3, 3×3, 5×5, 5×5, 5×5, 5×5. For the D network we use the same D network as the DCGAN [29].For the C network, we use an Alexnet [13] structure, and change the input to 128×128. We fix the latent vector dimension to be 256 and find this configuration sufficient for generating images. The batch normalization layer [11] is also applied after each convolution layer. The model is implemented using the deep learning toolbox Torch.

5.1. Visualization comparison with other models

In this experiment, we compare the proposed mean feature matching based CGAN introduced in Section 3.2 (FM-CGAN), and CVAE-GAN model with other generative models for image synthesis of fine-grained images.

In order to fairly compare each method, we use the same network structure and same training data for all methods.

All networks are trained from scratch. In the testing stage, the network architectures are the same. All three methods only use network G to generate images. Therefore, although our approach has more parameters in the training stage, we believe this comparison is fair.

We conduct experiments on three datasets: FaceScrub, 102 Category Flower and CUB-200 dataset. We perform category conditioned image generation for all methods. For each dataset, all methods are trained with all the data in that dataset. In the test stage, we first randomly chose a category c, and then randomly generate samples of that category by sampling the latent vector $z \sim N(0, I)$. For evaluation, we visualize the samples generated from all methods.

The comparison results are presented in Figure 5. All images are randomly selected without any personal bias. We observe that images generated by CVAE are often blurry. For traditional CGAN, the variation within a category is very small, which is because of the mode collapse. For FM-CGAN, we observe clear images with well preserved identities, but some images lose the structure of an object, such as the shape of the face. On the other hand, images generated by the proposed CVAE-GAN models look realistic and clear, and are non-trivially different from each other,

Algorithm 1 The training pipeline of the proposed CVAE-GAN algorithm.

Require: m, the batch size. n, class number. θ_E, initial E network parameters. θ_G, initial G network parameters. θ_D, initial D network parameters. θ_C, initial C network parameters, $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 10^{-3}$ and $\lambda_4 = 10^{-3}$.

1: **while** θ_G has not converged **do**
2:      Sample $\{x_r, c_r\} \sim P_r$ a batch from the real data;
3:      $\mathcal{L}_C \leftarrow -\log(P(c_r|x_r))$
4:      $z \leftarrow E(x_r, c_r)$
5:      $\mathcal{L}_{KL} \leftarrow KL(q(z|x_r, c_r)||P_z)$
6:      $x_f \leftarrow G(z, c_r)$
7:      Sample $z_p \sim P_z$ a batch of random noise, sample c_p a batch of random classes;
8:      $x_p \leftarrow G(z_p, c_p)$
9:      $\mathcal{L}_D \leftarrow -(\log(D(x_r)) + \log(1 - D(x_f)) + \log(1 - D(x_p)))$
10:      Calculate x_r feature center $\frac{1}{m}\sum_i^m f_D(x_r)$ and x_p feature center $\frac{1}{m}\sum_i^m f_D(x_p)$;
11:      $\mathcal{L}_{GD} \leftarrow \frac{1}{2}||\frac{1}{m}\sum_i^m f_D(x_r) - \frac{1}{m}\sum_i^m f_D(x_p)||_2^2$
12:      Calculate each class c_i feature center $f_C^{c_i}(x_r)$ for x_r and $f_C^{c_i}(x_p)$ for x_p using moving average method;
13:      $\mathcal{L}_{GC} \leftarrow \frac{1}{2}\sum_{c_i}||f_C^{c_i}(x_r) - f_C^{c_i}(x_p)||_2^2$
14:      $\mathcal{L}_G \leftarrow \frac{1}{2}(||x_r - x_f||_2^2 + ||f_D(x_r) - f_D(x_f)||_2^2 + ||f_C(x_r) - f_C(x_f)||_2^2)$
15:      $\theta_C \xleftarrow{+} -\nabla_{\theta_C}(\mathcal{L}_C)$
16:      $\theta_D \xleftarrow{+} -\nabla_{\theta_D}(\mathcal{L}_D)$
17:      $\theta_G \xleftarrow{+} -\nabla_{\theta_G}(\lambda_2\mathcal{L}_G + \lambda_3\mathcal{L}_{GD} + \lambda_4\mathcal{L}_{GC})$
18:      $\theta_E \xleftarrow{+} -\nabla_{\theta_E}(\lambda_1\mathcal{L}_{KL} + \lambda_2\mathcal{L}_G)$
19: **end while**

Figure 6. Results of attributes morphing.

	Real data	CVAE	CGAN	FM-CGAN	CVAE-GAN
Top-1 acc	99.61%	8.09%	61.97%	79.76%	97.78%
Realism	20.85	10.29	15.79	19.40	19.03

Table 1. Quantitative result of generated image quality of different methods. Please refer to Section 5.2 for details.

and adopt $exp(\mathbb{E}_x KL(p(y|x)||p(y)))$ as the metric to measure the realism and diversity of the generative models, where $p(y|x)$ represents the posterior probability of each class of generated samples. Images that contain meaningful objects should have a conditional label distribution $p(y|x)$ with low entropy. Moreover, if the model generate diverse images, the marginal $p(y) = \int p(y|G(z))dz$ should have high entropy. A larger score means the generator can produce more realistic and diverse images. The results are shown in Table 1. Our proposed CVAE-GAN and FM-CGAN achieve better scores than the other models, which are also very close to the real data.

5.3. Attributes Morphing

In this part, we validate that the attribute in the generated images will continuously change with the latent vector. We call this phenomenon attribute morphing. We also test our model on the FaceScrub, CUB-200 and 102 Category Flower datasets. We first select a pair of images x_1 and x_2 in the same category, and then extract the latent vector z_1 and z_2 using the encoder network E. Finally, we obtain a series of latent vectors z by linear interpolation, i.e., $z = \alpha z_1 + (1-\alpha)z_2, \alpha \in [0, 1]$. Figure 6 shows the results of attribute morphing. In each row, the attribute, such as pose, emotion, color, or flower number, gradually changes from left to right.

5.4. Image Inpainting

In this part, we show that our model can also be applied to image inpainting. We first randomly corrupt a 50×50 patch of an original 128×128 image x (Fig.7b), and then feed it to the E network to obtain a latent vector z, then we can synthesize an image x' by $G(z, c)$ where c is the class label, then we update the image by the following equation, i.e.,

$$x = M \odot x' + (1 - M) \odot x, \qquad (8)$$

where M is the binary mask for the corrupted patch, and $\odot$ denotes the element-wise product. So $(1 - M) \odot x$ is the uncorrupted area in the original image. The inpainting

especially for view-point and background color. Our model is also able to keep the identity information. It shows the strength of the proposed CVAE-GAN method.

5.2. Quantitative Comparison

Evaluating the quality of a synthesized image is challenging due to the variety of probabilistic criteria [38]. We attempt to measure the generative model on three criteria: discriminability, diversity and realism.

We use face images for this experiment. First, we randomly generate $53k$ samples (100 for each class) from CVAE, CGAN, FM-CGAN, and CVAE-GAN models for evaluation.

To measure discriminability, we use a pre-trained face classification network on the real data. Here we use GoogleNet [36]. With this trained model, we evaluate the top-1 accuracy of the generated samples from each method. The results are shown in Table 1. Our model achieves the best top-1 accuracy with a big gap to other generative models. This demonstrates the effectiveness of the proposed method.

Following the method in [33], we use the *Inception Score* to evaluate the realism and diversity of generated samples. We train a classification model on the CASIA [48] datasets,
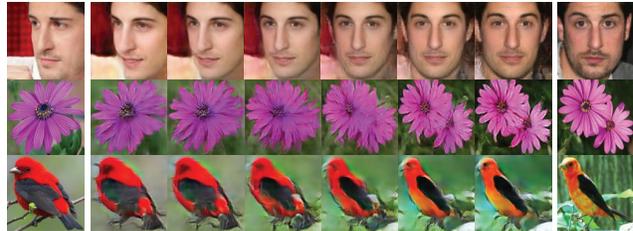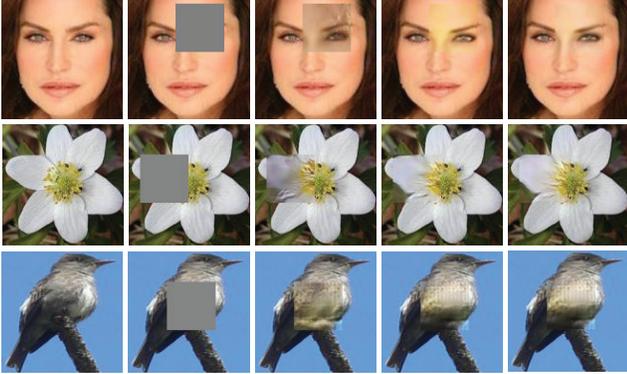
a) Original images b) Masked images c) CVAE-GAN-1 d) CVAE-GAN-5 e) CVAE-GAN-10

Figure 7. Result of image inpainting using our proposed model CVAE-GAN-1 $\sim$ 10 shows the results of iteration $1 \sim 10$.

results are shown in Figure 7 (c). We should emphasize that all input images are downloaded from websites, with none of them belonging to the training data. We can iteratively feed the resulting images into the model to obtain a better results, as shown in Figure 7 (d,e).

5.5. Comparing Different Combination of Losses

In our model, we propose using pairwise feature matching at the image pixel level, the feature level in the classification network C and the discriminative network D to update the network G. To understand the effects of each loss component, we separate the $\mathcal{L}_G + \mathcal{L}_{GD} + \mathcal{L}_{GC}$ to three parts: $\mathcal{L}_G(img) + \mathcal{L}_G(D) + \mathcal{L}_G(C)$, where $\mathcal{L}_G(img)$ is the ℓ_2 distance at the pixel level of the image, $\mathcal{L}_G(D)$ is the ℓ_2 distance at the feature level in the discriminative network D, $\mathcal{L}_G(C)$ is the ℓ_2 distance at the feature level in the classification network C.

We repeat the training of the CVAE-GAN model with the same settings but using different combination of losses in $\mathcal{L}_G(img)$, $\mathcal{L}_G(D)$, and $\mathcal{L}_G(C)$, and compared the quality of the reconstructed samples. As shown in Fig. 8, we find that removing the adversarial loss $\mathcal{L}_G(D)$ will cause the model to generate blurry images. Removing the pixel level reconstruction loss $\mathcal{L}_G(img)$ causes images to lose details. Lastly, if we remove the feature level loss $\mathcal{L}_G(C)$ in the classification network C, the generated samples will lose category info. Despite this, our model produces best results.

5.6. CVAE-GAN for Data Augmentation

We further show that the images synthesized from our model can be used for data augmentation for training better face recognition model. We use the FaceScrub dataset as training data, and test using the LFW [16] dataset.

We experiment with two data augmentation strategies: 1) generating more images for existing identities in the training datasets; 2) generating new identities by mixing different identities. We test these two kinds of data augmentation methods. For 1), we randomly generate about 200 images
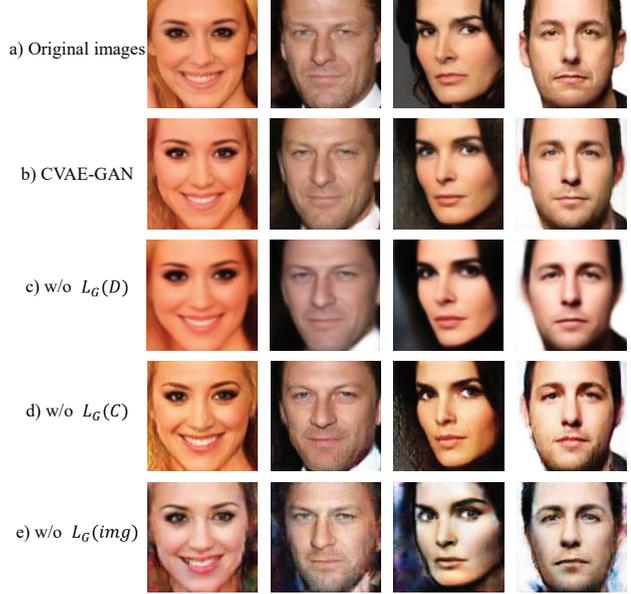

a) Original images
b) CVAE-GAN
c) w/o $L_G(D)$
d) w/o $L_G(C)$
e) w/o $L_G(img)$

Figure 8. Visualization comparison between different generator G, each trained with different combination of losses.

Method	Training Data	Accuracy
no data augmentation	$80K$	91.87%
existing identities augmentation	$80K + 100K$	92.77%
$5k$ new identities augmentation	$80K + 500K$	92.98%

Table 2. Results of face data augmentation.

per person, totaling $100k$ images. For 2), we create $5k$ new identities by randomly mixing the label of three different existing identities, and generate 100 images for each new identity. For both strategies, the generated images are combined with the Facescrub dataset to train a face recognition model.

In the testing stage, we directly use the cosine similarity of features to measure the similarity between two faces. In Table 2, we compare face verification accuracy on the LFW dataset with and without additional synthesized faces. With the data augmentation of new identities, we achieve about 1.0% improvement in accuracy compared with no augmentation. This demonstrates that our generative network has a certain extrapolation ability.

6. Conclusion

In this paper, we propose a new CVAE-GAN model for fine-grained category image generation. The superior performance on three different datasets demonstrates the ability to generate various kinds of objects. The proposed method can support a wide variety of applications, including image generation, attribute morphing, image inpainting, and data augmentation for training better face recognition models. Our future work will explore how to generate samples of an unknown category, such as face images of a person that do not exist in the training dataset.

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer, 2014.

[4] 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*, pages 1486–1494, 2015.

[5] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[6] A. Dosovitskiy, J. Springenberg, M. Tatarchenko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. 2016.

[7] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014, 2014.

[8] 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.

[9] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[10] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[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] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In *AISTATS*, volume 1, page 2, 2011.

[15] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[16] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016.

[17] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[18] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[19] V. Mnih, G. E. Hinton, et al. Generating more realistic images using gated mrf's. In *Advances in Neural Information Processing Systems*, pages 2002–2010, 2010.

[20] Y. Mroueh, T. Sercu, and V. Goel. Mcgan: Mean and covariance feature matching gan. *arXiv preprint arXiv:1702.08398*, 2017.

[21] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 343–347. IEEE, 2014.

[22] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[23] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

[24] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[25] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

[26] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

[27] H. Permuter, J. Francos, and I. H. Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–569. IEEE, 2003.

[28] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[29] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[31] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[32] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, volume 1, page 3, 2009.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[34] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[35] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov mod-

els. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[37] L. Theis, R. Hosseini, and M. Bethge. Mixtures of conditional gaussian scale mixtures applied to multiscale image representations. *PloS one*, 7(7):e39857, 2012.

[38] L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[39] Z. Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[40] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[41] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[42] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[43] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[44] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu. A theory of generative convnet. *arXiv preprint arXiv:1602.03264*, 2016.

[45] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.

[46] L. Xu and M. I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[47] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.

[48] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.