Multi-Adversarial Variational Autoencoder Nets for Simultaneous Image Generation and Classification

Abdullah-Al-Zubaer Imran and Demetri Terzopoulos

Abstract Discriminative deep-learning models are often reliant on copious labeled training data. By contrast, from relatively small corpora of training data, deep generative models can learn to generate realistic images approximating real-world distributions. In particular, the proper training of Generative Adversarial Networks (GANs) and Variational AutoEncoders (VAEs) enables them to perform semi-supervised image classification. Combining the power of these two models, we introduce Multi-Adversarial Variational autoEncoder Networks (MAVENs), a novel deep generative model that incorporates an ensemble of discriminators in a VAE-GAN network in order to perform simultaneous adversarial learning and variational inference. We apply MAVENs to the generation of synthetic images and propose a new distribution measure to quantify the quality of these images. Our experimental results with only 10% labeled training data from the computer vision and medical imaging domains demonstrate performance competitive to state-of-the-art semi-supervised models in simultaneous image generation and classification tasks.

1 Introduction

Training deep neural networks usually requires copious data, yet obtaining large, accurately labeled datasets for image classification and other tasks remains a fundamental challenge [36]. Although there has been explosive progress in the production of vast quantities of high resolution images, large collections of labeled data required for supervised learning remain scarce. Especially in domains such as medical imaging, datasets are often limited in size due to privacy issues, and annotation by medical experts is expensive, time-consuming, and prone to human subjectivity,

A.-A.-Z. Imran (✉) · D. Terzopoulos
University of California, Los Angeles, CA 90095, USA
e-mail: aimran@cs.ucla.edu

D. Terzopoulos
e-mail: dt@cs.ucla.edu

M. A. Wani et al. (eds.), *Deep Learning Applications, Volume 2*,
Advances in Intelligent Systems and Computing 1232,
https://doi.org/10.1007/978-981-15-6759-9_11

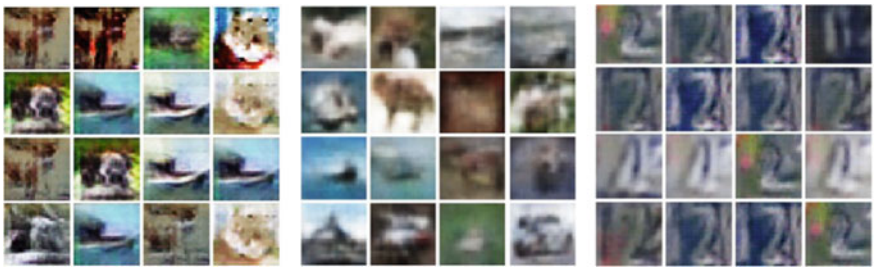


Fig. 1 Image generation based on the CIFAR-10 dataset [19]: **a** Relatively good images generated by a GAN. **b** Blurry images generated by a VAE. Based on the SVHN dataset [24]: **c** mode collapsed images generated by a GAN

inconsistency, and error. Even when large labeled datasets become available, they are often highly imbalanced and non-uniformly distributed. In an imbalanced medical dataset there will be an over-representation of common medical problems and an under-representation of rarer conditions. Such biases make the training of neural networks across multiple classes with consistent effectiveness very challenging.

The small-training-data problem is traditionally mitigated through simplistic and cumbersome data augmentation, often by creating new training examples through translation, rotation, flipping, etc. The missing or mismatched label problem may be addressed by evaluating similarity measures over the training examples. This is not always robust and its effectiveness depends largely on the performance of the similarity measuring algorithms.

With the advent of deep generative models such as Variational AutoEncoders (VAEs) [18] and Generative Adversarial Networks (GANs) [9], the ability to learn underlying data distributions from training samples has become practical in common scenarios where there is an abundance of unlabeled data. With minimal annotation, efficient semi-supervised learning could be the preferred approach [16]. More specifically, based on small quantities of annotation, realistic new training images may be generated by models that have learned real-world data distributions (Fig. 1a). Both VAEs and GANs may be employed for this purpose.

VAEs can learn dimensionality-reduced representations of training data and, with an explicit density estimation, can generate new samples. Although VAEs can perform fast variational inference, VAE-generated samples are usually blurry (Fig. 1b). On the other hand, despite their successes in generating images and semi-supervised classifications, GAN frameworks remain difficult to train and there are challenges in using GAN models, such as non-convergence due to unstable training, diminished gradient issues, overfitting, sensitivity to hyper-parameters, and mode collapsed image generation (Fig. 1c).

Despite the recent progress in high-quality image generation with GANs and VAEs, accuracy and image quality are usually not ensured by the same model, especially in multiclass image classification tasks. To tackle this shortcoming, we propose

a novel method that can simultaneously learn image generation and multiclass image classification. Specifically, our work makes the following contributions:

1. The Multi-Adversarial Variational autoEncoder Network, or MAVEN, a novel multiclass image classification model incorporating an ensemble of discriminators in a combined VAE-GAN network. An ensemble layer combines the feedback from multiple discriminators at the end of each batch. With the inclusion of ensemble learning at the end of a VAE-GAN, both generated image quality and classification accuracy are improved simultaneously.
2. A simplified version of the Descriptive Distribution Distance (DDD) [14] for evaluating generative models, which better represents the distribution of the generated data and measures its closeness to the real data.
3. Extensive experimental results utilizing two computer vision and two medical imaging datasets.[1] These confirm that our MAVEN model improves upon the simultaneous image generation and classification performance of a GAN and of a VAE-GAN with the same set of hyper-parameters.

2 Related Work

Several techniques have been proposed to stabilize GAN training and avoid mode collapse. Nguyen et al. [26] proposed a model where a single generator is used alongside dual discriminators. Durugkar et al. [7] proposed a model with a single generator and feedback aggregated over several discriminators, considering either the average loss over all discriminators or only the discriminator with the maximum loss in relation to the generator's output. Neyshabur et al. [25] proposed a framework in which a single generator simultaneously trains against an array of discriminators, each of which operates on a different low-dimensional projection of the data. Moridido et al. [23], arguing that all the previous approaches restrict the discriminator's architecture thereby compromising extensibility, proposed the Dropout-GAN, where a single generator is trained against a dynamically changing ensemble of discriminators. However, there is a risk of dropping out all the discriminators. Feature matching and minibatch discrimination techniques have been proposed [32] for eliminating mode collapse and preventing overfitting in GAN training.

Realistic image generation helps address problems due to the scarcity of labeled data. Various architectures of GANs and their variants have been applied in ongoing efforts to improve the accuracy and effectiveness of image classification. The GAN framework has been utilized as a generic approach to generating realistic training images that synthetically augment datasets in order to combat overfitting; e.g., for synthetic data augmentation in liver lesions [8], retinal fundi [10], histopathology [13], and chest X-rays [16, 31]. Calimeri et al. [3] employed a LAPGAN [6] and Han et al. [11] used a WGAN [1] to generate synthetic brain MR images. Bermudez

[1] This chapter significantly expands upon our ICMLA 2019 publication [15], which excluded our experiments on medical imaging datasets.

et al. [2] used a DCGAN [29] to generate 2D brain MR images followed by an autoencoder for image denoising. Chuquicusma et al. [4] utilized a DCGAN to generate lung nodules and then conducted a Turing test to evaluate the quality of the generated samples. GAN frameworks have also been shown to improve accuracy of image classification via the generation of new synthetic training images. Frid et al. [8] used a DCGAN and an ACGAN [27] to generate images of three liver lesion classes to synthetically augment the limited dataset and improve the performance of a Convolutional Neural Net (CNN) in liver lesion classification. Similarly, Salehinejad et al. [31] employed a DCGAN to artificially simulate pathology across five classes of chest X-rays in order to augment the original imbalanced dataset and improve the performance of a CNN in chest pathology classification.

The GAN framework has also been utilized in semi-supervised learning architectures to leverage unlabeled data alongside limited labeled data. The following efforts demonstrate how incorporating unlabeled data in the GAN framework has led to significant improvements in the accuracy of image-level classification. Madani et al. [20] used an order of magnitude less labeled data with a DCGAN in semi-supervised learning yet showed comparable performance to a traditional supervised CNN classifier and furthermore demonstrated reduced domain overfitting by simply supplying unlabeled test domain images. Springenberg et al. [33] combined a WGAN and CatGAN [35] for unsupervised and semi-supervised learning of feature representation of dermoscopy images.

Despite the aforecited successes, GAN frameworks remain challenging to train, as we discussed above. Our MAVEN framework mitigates the difficulties of training GANs by enabling training on a limited quantity of labeled data, preventing overfitting to a specific data domain source, and preventing mode collapse, while supporting multiclass image classification.

3 The MAVEN Architecture

Figure 2 illustrates the models that serve as precursors to our MAVEN architecture.

The VAE is an explicit generative model that uses two neural nets, an encoder E and decoder D'. Network E learns an efficient compression of real data x into a lower dimensional latent representation space $z(x)$; i.e., $q_\lambda(z|x)$. With neural network likelihoods, computing the gradient becomes intractable; however, via differentiable, non-centered re-parameterization, sampling is performed from an approximate function $q_\lambda(z|x) = N(z; \mu_\lambda, \sigma^2_\lambda)$, where $z = \mu_\lambda + \sigma_\lambda \odot \hat{\varepsilon}$ with $\hat{\varepsilon} \sim N(0, 1)$. Encoder E yields μ and σ, and with the re-parameterization trick, z is sampled from a Gaussian distribution. Then, with D', new samples are generated or real data samples are reconstructed; i.e., D' provides parameters for the real data distribution $p_\lambda(x|z)$. Subsequently, a sample drawn from $p_\phi(x|z)$ may be used to reconstruct the real data by marginalizing out z.

The GAN is an implicit generative model where a generator G and a discriminator D compete in a minimax game over the training data in order to improve their perfor-

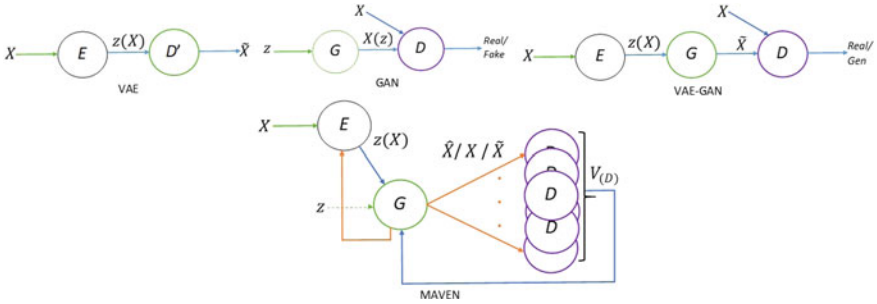


Fig. 2 Our MAVEN architecture compared to those of the VAE, GAN, and VAE-GAN. In the MAVEN, inputs to D can be real data X, or generated data $\hat{X}$ or $\tilde{X}$. An ensemble ensures the combined feedback from the discriminators to the generator

mance. Generator G tries to approximate the underlying distribution of the training data and generates synthetic samples, while discriminator D learns to discriminate synthetic samples from real samples. The GAN model is trained on the following objectives:

$$\max_D V(D) = E_{x\sim p_data(x)}[\log D(x)] + E_{x\sim p_g(z)}[\log(1 - D(G(z))]; \quad (1)$$

$$\min_G V(G) = E_{x\sim p_z(z)}[\log(1 - D(G(z))]. \quad (2)$$

G takes a noise sample $z \sim p_g(z)$ and learns to map it into image space as if it comes from the original data distribution $p_{\text{data}}(x)$, while D takes as input either real image data or generated image data and provides feedback to G as to whether that input is real or generated. On the one hand, D wants to maximize the likelihood for real samples and minimize the likelihood of generated samples; on the other hand, G wants D to maximize the likelihood of generated samples. A Nash equilibrium results when D can no longer distinguish real and generated samples, meaning that the model distribution matches the data distribution.

Makhzani et al. [21] proposed the adversarial training of VAEs; i.e., VAE-GANs. Although they kept both D' and G, one can merge these networks since both can generate data samples from the noise samples of the representation z. In this case, D receives real data samples x and generated samples $\tilde{x}$ or $\hat{x}$ via G. Although G and D compete against each other, the feedback from D eventually becomes predictable for G and it keeps generating samples from the same class, at which point the generated samples lack heterogeneity. Figure 1c shows an example where all the generated images are of the same class. Durugkar et al. [7] proposed that using multiple discriminators in a GAN model helps improve performance, especially for resolving this mode collapse. Moreover, a dynamic ensemble of multiple discriminators has recently been proposed to address the issue [23] (Fig. 3).

As in a VAE-GAN, our MAVEN has three components, E, G, and D; all are CNNs with convolutional or transposed convolutional layers. First, E takes real samples

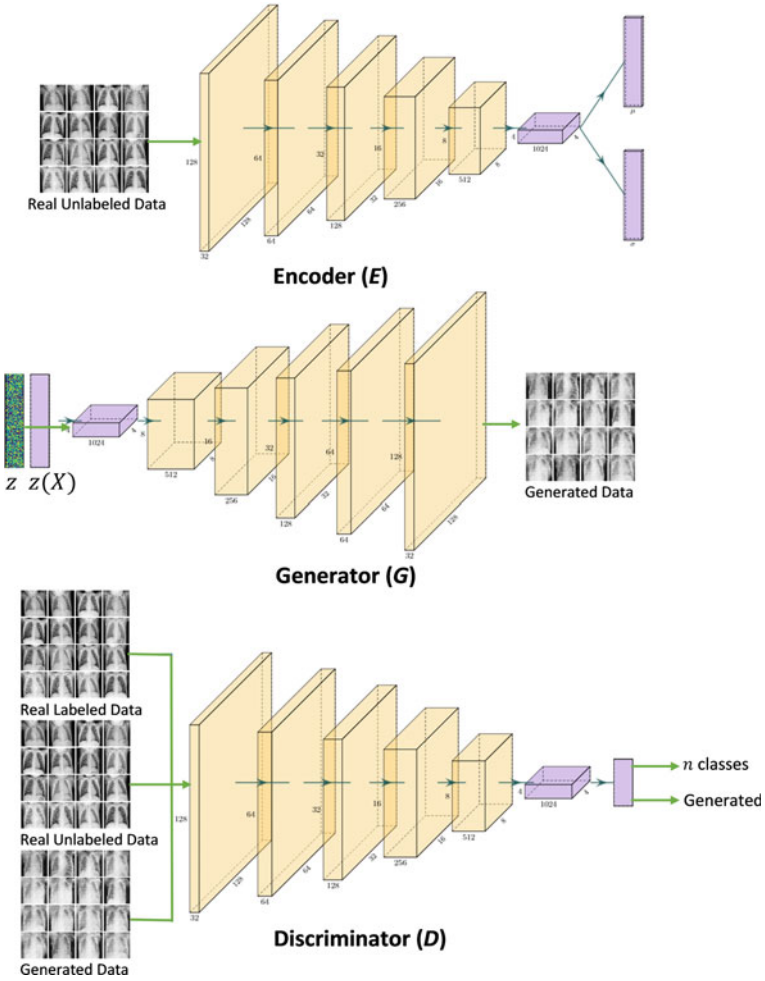


Fig. 3 The three convolutional neural networks, E, G, and D, in the MAVEN

x and generates a dimensionality-reduced representation $z(x)$. Second, G can input samples from noise distribution $z \sim p_g(z)$ or sampled noise $z(x) \sim q_\lambda(x)$ and it produces generated samples. Third, D takes inputs from distributions of real labeled data, real unlabeled data, and generated data. Fractionally strided convolutions are performed in G to obtain the image dimension from the latent code. The goal of an autoencoder is to maximize the Evidence Lower Bound (ELBO). The intuition here is to show the network more real data. The greater the quantity of real data that it sees, the more evidence is available to it and, as a result, the ELBO can be maximized faster.

In our MAVEN architecture (Fig. 2), the VAE-GAN combination is extended to include multiple discriminators aggregated in an ensemble layer. K discriminators are collected and the combined feedback

$$V(D) = \frac{1}{K} \sum_{k=1}^{K} w_k D_k \tag{3}$$

is passed to G. In order to randomize the feedback from the multiple discriminators, a single discriminator is randomly selected.

4 Semi-Supervised Learning

Algorithm 1 presents the overall training procedure of our MAVEN model. In the forward pass, different real samples x into E and noise samples z into G provide different inputs for each of the multiple discriminators. In the backward pass, the combined feedback from the discriminators is computed and passed to G and E.

In the conventional image generator GAN, D works as a binary classifier—it classifies the input image as real or generated. To facilitate the training for an n-class classifier, D assumes the role of an $(n + 1)$-classifier. For multiple logit generation, the sigmoid function is replaced by a softmax function. Now, it can receive an image x as input and output an $(n + 1)$-dimensional vector of logits $\{l_1, \ldots, l_n, l_{n+1}\}$, which are finally transformed into class probabilities for the n labels in the real data while class $(n + 1)$ denotes the generated data. The probability that x is real and belongs to class $1 \leq i \leq n$ is

$$p(y = i \mid x) = \frac{\exp(l_i)}{\sum_{j=1}^{n+1} \exp(l_j)} \tag{4}$$

while the probability that x is generated corresponds to $i = n + 1$ in (4). As a semi-supervised classifier, the model takes labels only for a small portion of the training data. It is trained via supervised learning from the labeled data, while it learns in an unsupervised manner from the unlabeled data. The advantage comes from generating new samples. The model learns the classifier by generating samples from different classes.

4.1 Losses

Three networks, E, G, and D, are trained on different objectives. E is trained on maximizing the ELBO, G is trained on generating realistic samples, and D is trained to learn a classifier that classifies generated samples or particular classes for the real data samples.

Algorithm 1 MAVEN Training procedure.
m is the number of samples; B is the minibatch-size; and K is the number of discriminators.

steps $\leftarrow \frac{m}{B}$
for each **epoch do**
  for each step in *steps* **do**
    for $k = 1$ to K **do**
      Sample minibatch $z^{(1)}, \ldots, z^{(m)}$ from $p_g(z)$
      Sample minibatch $x^{(1)}, \ldots, x^{(m)}$ from $p_{\text{data}}(x)$
      Update D_k by ascending along its gradient:

$$\nabla_{D_k} \frac{1}{m} \sum_{i=1}^{m} \big[\log D_k(x_i) + \log(1 - D_k(G(z_i)))\big]$$

    end for
    Sample minibatch $z_k^{(1)}, \ldots, z_k^{(m)}$ from $p_g(z)$
    if ensemble is 'mean' **then**
      Assign weights w_k to the D_k
      Determine the mean discriminator

$$D_\mu = \frac{1}{K} \sum_{k}^{K} w_k D_k$$

    end if
    Update G by descending along its gradient from the ensemble of D_μ:

$$\nabla_G \frac{1}{m} \sum_{i=1}^{m} \big[\log(1 - D_\mu(G(z_i)))\big]$$

    Sample minibatch $x^{(1)}, \ldots, x^{(m)}$ from $p_{\text{data}}(x)$
    Update E along its expectation function:

$$\nabla_{E_{q_\lambda}} \left[\log \frac{p(z)}{q_\lambda(z \mid x)}\right]$$

  end for
end for

4.1.1 D Loss

Since the model is trained on both labeled and unlabeled training data, the loss function of D includes both supervised and unsupervised losses. When the model receives real labeled data, it is the standard supervised learning loss

$$L_{D_{\text{supervised}}} = -\mathbb{E}_{x,y \sim p_{\text{data}}} \log[p(y = i \mid x)], \quad i < n + 1. \tag{5}$$

When it receives unlabeled data from three different sources, the unsupervised loss contains the original GAN loss for real and generated data from two different sources:

syn$G$ directly from G and syn$E$ from E via G. The three losses,

$$L_{D_{\text{real}}} = -\mathbb{E}_{x\sim p_{\text{data}}} \log[1 - p(y = n + 1 \mid x)], \tag{6}$$

$$L_{D_{\text{syn}G}} = -\mathbb{E}_{\hat{x}\sim G} \log[p(y = n + 1 \mid \hat{x})], \tag{7}$$

$$L_{D_{\text{syn}E}} = -\mathbb{E}_{\tilde{x}\sim G} \log[p(y = n + 1 \mid \tilde{x})], \tag{8}$$

are combined as the unsupervised loss in D:

$$L_{D_{\text{unsupervised}}} = L_{D_{\text{real}}} + L_{D_{\text{syn}G}} + L_{D_{\text{syn}E}}. \tag{9}$$

4.1.2 G Loss

For G, the feature loss is used along with the original GAN loss. Activation $f(x)$ from an intermediate layer of D is used to match the feature between real and generated samples. Feature matching has shown much potential in semi-supervised learning [32]. The goal of feature matching is to encourage G to generate data that matches real data statistics. It is natural for D to find the most discriminative features in real data relative to data generated by the model:

$$L_{G_{\text{feature}}} = \left\| \mathbb{E}_{x\sim p_{\text{data}}} f(x) - \mathbb{E}_{\hat{x}\sim G} f(\hat{x}) \right\|_2^2. \tag{10}$$

The total G loss becomes the combined feature loss (10) plus the cost of maximizing the log-probability of D making a mistake on the generated data (syn$G$ / syn$E$); i.e.,

$$L_G = L_{G_{\text{feature}}} + L_{G_{\text{syn}G}} + L_{G_{\text{syn}E}}, \tag{11}$$

where

$$L_{G_{\text{syn}G}} = -\mathbb{E}_{\hat{x}\sim G} \log[1 - p(y = n + 1 \mid \hat{x})], \tag{12}$$

and

$$L_{G_{\text{syn}E}} = -\mathbb{E}_{\tilde{x}\sim G} \log[1 - p(y = n + 1 \mid \tilde{x})]. \tag{13}$$

4.1.3 E Loss

In the encoder E, the maximization of ELBO is equivalent to minimizing the KL-divergence, allowing approximate posterior inferences. Therefore the loss function includes the KL-divergence and also a feature loss to match the features in the syn$E$ data with the real data distribution. The loss for the encoder is

$$L_E = L_{E_{\text{KL}}} + L_{E_{\text{feature}}}, \tag{14}$$

where

$$L_{E_{\mathrm{KL}}} = -\,\mathrm{KL}\left[q_\lambda(z \mid x) \parallel p(z)\right] = \mathbb{E}_{q_\lambda(z|x)}\left[\log \frac{p(z)}{q_\lambda(z \mid x)}\right] \approx \mathbb{E}_{q_\lambda(z|x)} \tag{15}$$

and

$$L_{E_{\mathrm{feature}}} = \left\| \mathbb{E}_{x \sim p_{\mathrm{data}}} f(x) - \mathbb{E}_{\tilde{x} \sim G} f(\tilde{x}) \right\|_2^2 . \tag{16}$$

5 Experiments

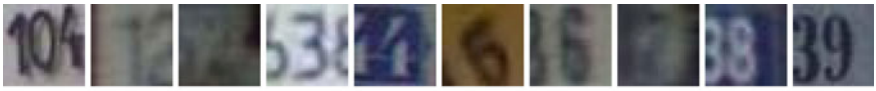
Applying semi-supervised learning using training data that is only partially labeled, we evaluated our MAVEN model in image generation and classification tasks in a number of experiments. For all our experiments, we used 10% labeled and 90% unlabeled training data.

5.1 Data

We employed the following four image datasets:

1. The Street View House Numbers (SVHN) dataset [24] (Fig. 4a). There are 73,257 digit images for training and 26,032 digit images for testing. Out of two versions of the images, we used the version which has MNIST-like 32 × 32 pixel RGB color images centered around a single digit. Each image is labeled as belonging to one of 10 classes: digits 0–9.
2. The CIFAR-10 dataset [19] (Fig. 4b). It consists of 60,000 32 × 32 pixel RGB color images in 10 classes. There are 50,000 training images and 10,000 test images. Each image is labeled as belonging to one of 10 classes: plane, auto, bird, cat, deer, dog, frog, horse, ship, and truck.
3. The anterior-posterior Chest X-Ray (CXR) dataset [17] (Fig. 4c). The dataset contains 5,216 training and 624 test images. Each image is labeled as belonging to one of three classes: normal, bacterial pneumonia (b-pneumonia), and viral pneumonia (v-pneumonia).
4. The skin lesion classification (SLC) dataset (Fig. 4d). We employed 2,000 RGB skin images from the ISIC 2017 dermoscopy image dataset [5]; of which we used 1,600 for training and 400 for testing. Each image is labeled as belonging to one of two classes: non-melanoma and melanoma.

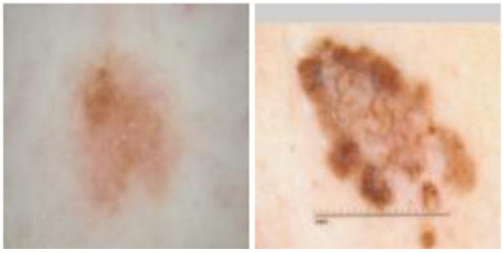
For the SVHN and CIFAR-10 datasets, the images were normalized and provided to the models in their original (32 × 32 × 3) pixel sizes. For the CXR dataset, the images were normalized and resized to 128 × 128 × 1 pixels. For the SLC dataset, the images were resized to 128 × 128 × 3 pixels.



SVHN (from left, classes 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

CIFAR-10 (from left, classes plane, auto, bird, cat, deer, dog, frog, horse, ship, truck)

CXR (from left, classes normal, bacterial, viral)



SLC (from left, classes non-melanoma, melanoma)

Fig. 4 Example images of each class in the four datasets

5.2 Implementation Details

To compare the image generation and multiclass classification performance of our MAVEN model, we used two baselines, the Deep Convolutional GAN (DC-GAN) [29] and the VAE-GAN. The same generator and discriminator architectures were used for DC-GAN and MAVEN models and the same encoder was used for the VAE-GAN and MAVEN models. For our MAVENs, we experimented with 2, 3, and 5 discriminators. In addition to using the mean feedback of the multiple discriminators, we also experimented with feedback from a randomly selected discriminator. The six MAVEN variants are therefore denoted MAVEN-m2D, MAVEN-m3D, MAVEN-m5D, MAVEN-r2D, MAVEN-r3D, and MAVEN-r5D, where "m" indicates mean feedback while "r" indicates random feedback.

All the models were implemented in TensorFlow and run on a single Nvidia Titan GTX (12 GB) GPU. For the discriminator, after every convolutional layer, a dropout layer was added with a dropout rate of 0.4. For all the models, we consistently used the Adam optimizer with a learning rate of 2.0^{-4} for G and D, and 1.0^{-5} for E, with a momentum of 0.9. All the convolutional layers were followed by batch normalizations. Leaky ReLU activations were used with $\alpha = 0.2$.

5.3 Evaluation

5.3.1 Image Generation Performance Metrics

There are no perfect performance metrics for measuring the quality of generated samples. However, to assess the quality of the generated images, we employed the widely used Fréchet Inception Distance (FID) [12] and a simplified version of the Descriptive Distribution Distance (DDD) [14]. To measure the Fréchet distance between two multivariate Gaussians, the generated samples and real data samples are compared through their distribution statistics:

$$\text{FID} = \left\| \mu_{\text{data}} - \mu_{\text{syn}} \right\|^2 + \text{Tr}\left(\Sigma_{\text{data}} + \Sigma_{\text{syn}} - 2\sqrt{\Sigma_{\text{data}}\,\Sigma_{\text{syn}}} \right). \tag{17}$$

Two distribution samples, $X_{\text{data}} \sim \mathcal{N}(\mu_{\text{data}}, \Sigma_{\text{data}})$ and $X_{\text{syn}} \sim \mathcal{N}(\mu_{\text{syn}}, \Sigma_{\text{syn}})$, for real and model data, respectively, are calculated from the 2,048-dimensional activations of the pool3 layer of Inception-v3 [32]. DDD measures the closeness of a generated data distribution to a real data distribution by comparing descriptive parameters from the two distributions. We propose a simplified version based on the first four moments of the distributions, computed as the weighted sum of normalized differences of moments, as follows:

$$\text{DDD} = -\sum_{i=1}^{4} \log w_i \left| \mu_{\text{data}_i} - \mu_{\text{syn}_i} \right|, \tag{18}$$

where the μ_{data_i} are the moments of the data distribution, the μ_{syn_i} are the moments of the model distribution, and the w_i are the corresponding weights found in an exhaustive search. The higher order moments are weighted more in order to emphasize the stability of a distribution. For both the FID and DDD, lower scores are better.

5.3.2 Image Classification Performance Metrics

To evaluate model performance in classification, we used two measures, image-level classification accuracy and class-wise F1 scoring. The F1 score is

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{19}$$

with

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{20}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively.

5.4 Results

We measured the image classification performances of the models with cross-validation and in the following sections report the average scores from running each model 10 times.

5.4.1 SVHN

For the SVHN dataset, we randomly selected 7,326 labeled images and they along with the remaining 65,931 unlabeled images were provided to the network as training data. All the models were trained for 300 epochs and then evaluated. We generated new images equal in number to the training set size. Figure 5 presents a visual comparison of a random selection of images generated by the DC-GAN, VAE-GAN, and MAVEN models and real training images. Figure 6 compares the image intensity histograms of 10K randomly sampled real images and equally many images sampled from among those generated by each of the different models.

Generally speaking, our MAVEN models generate images that are more realistic than those generated by the DC-GAN and VAE-GAN models. This was further corroborated by randomly sampling 10K generated images and 10K real images. The generated image quality measurement was performed for the eight different models. Table 1 reports the resulting FID and DDD scores. For the FID score calculation, the score is reported after running the pre-trained Inception-v3 network for 20 epochs for each model. The MAVEN-r3D model achieved the best FID score and the best DDD score was achieved by the MAVEN-m5D model.

Table 2 compares the classification performance of all the models for the SVHN dataset. The MAVEN model consistently outperformed the DC-GAN and VAE-GAN classifiers both in classification accuracy and class-wise F1 scores. Among all the models, our MAVEN-m2D and MAVEN-m3D models were the most accurate.

Fig. 5 Visual comparison of image samples from the SVHN dataset against those generated by the different models

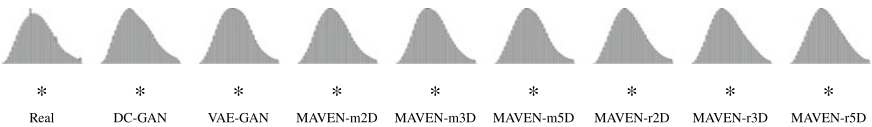


Fig. 6 Histograms of the real SVHN training data, and of the data generated by the DC-GAN and VAE-GAN models and by our MAVEN models with mean and random feedback from 2, 3, to 5 discriminators

Table 1 Minimum FID and DDD scores achieved by the DC-GAN, VAE-GAN, and MAVEN models for the CIFAR-10, SVHN, CXR, and SLC datasets

CIFAR-10			SVHN			CXR			SLC		
Model	FID	DDD	Model	FID	DDD	Model	FID	DDD	Model	FID	DDD
DC-GAN	61.293±0.209	0.265	DC-GAN	16.789±0.303	0.343	DC-GAN	152.511±0.370	0.145	DC-GAN	1.828±0.370	0.795
VAE-GAN	15.511±0.125	0.224	VAE-GAN	13.252±0.001	0.329	VAE-GAN	141.422±0.580	0.107	VAE-GAN	1.828±0.580	0.795
MAVEN-m2D	12.743±0.242	0.223	MAVEN-m2D	11.675±0.001	0.309	MAVEN-m2D	141.339±0.420	0.138	MAVEN-m2D	1.874±0.270	0.802
MAVEN-m3D	**11.316±0.808**	**0.190**	MAVEN-m3D	11.515±0.065	0.300	MAVEN-m3D	**140.865±0.983**	**0.018**	MAVEN-m3D	**0.304±0.018**	**0.249**
MAVEN-m5D	12.123±0.140	0.207	MAVEN-m5D	10.909±0.001	**0.294**	MAVEN-m5D	147.316±1.169	0.100	MAVEN-m5D	1.518±0.190	0.793
MAVEN-r2D	12.820±0.584	0.194	MAVEN-r2D	11.384±0.001	0.316	MAVEN-r2D	154.501±0.345	0.038	MAVEN-r2D	1.505±0.130	0.789
MAVEN-r3D	12.620±0.001	0.202	MAVEN-r3D	**10.791±0.029**	0.357	MAVEN-r3D	158.749±0.297	0.179	MAVEN-r3D	0.336±0.080	0.783
MAVEN-r5D	18.509±0.001	0.215	MAVEN-r5D	11.052±0.751	0.323	MAVEN-r5D	152.778±1.254	0.180	MAVEN-r5D	1.812±0.014	0.795
DO-GAN [23]	88.60± 0.08	–									
TTUR [12]	36.9	–									
C-GAN [34]	27.300	–									
AIQN [28]	49.500	–									
SN-GAN [22]	21.700	–									
LM [30]	18.9	–									

Table 2 Average cross-validation accuracy and class-wise F1 scores in the semi-supervised classification performance comparison of the DC-GAN, VAE-GAN, and MAVEN models using the SVHN dataset

Model	Accuracy	F1 scores									
		0	1	2	3	4	5	6	7	8	9
DC-GAN	0.876	0.860	0.920	0.890	0.840	0.890	0.870	0.830	0.890	0.820	0.840
VAE-GAN	0.901	0.900	0.940	0.930	0.860	0.920	0.900	0.860	0.910	0.840	0.850
MAVEN-m2D	**0.909**	0.890	0.930	0.940	0.890	0.930	0.900	0.870	0.910	0.870	0.890
MAVEN-m3D	**0.909**	0.910	0.940	0.940	0.870	0.920	0.890	0.870	0.920	0.870	0.860
MAVEN-m5D	0.905	0.910	0.930	0.930	0.870	0.930	0.900	0.860	0.910	0.860	0.870
MAVEN-r2D	0.905	0.910	0.930	0.940	0.870	0.930	0.890	0.860	0.920	0.850	0.860
MAVEN-r3D	0.907	0.890	0.910	0.920	0.870	0.900	0.870	0.860	0.900	0.870	0.890
MAVEN-r5D	0.903	0.910	0.930	0.940	0.860	0.910	0.890	0.870	0.920	0.850	0.870

5.4.2 CIFAR-10

For the CIFAR-10 dataset, we used 50 K training images, only 5 K of them labeled. All the models were trained for 300 epochs and then evaluated. We generated new images equal in number to the training set size. Figure 7 visually compares a random selection of images generated by the DC-GAN, VAE-GAN, and MAVEN models and real training images. Figure 8 compares the image intensity histograms of 10K randomly sampled real images and equally many images sampled from among those generated by each of the different models. Table 1 reports the FID and DDD scores. As the tabulated results suggest, our MAVEN models achieved better FID scores than some of the recently published models. Note that those models were implemented in different settings.

As for the visual comparison, the FID and DDD scores confirmed more realistic image generation by our MAVEN models compared to the DC-GAN and VAE-GAN models. The MAVEN models have smaller FID scores, except for MAVEN-r5D. MAVEN-m3D has the smallest FID and DDD scores among all the models.

Table 3 compares the classification performance of all the models with the CIFAR-10 dataset. All the MAVEN models performed better than the DC-GAN and VAE-GAN models. In particular, MAVEN-m5D achieved the best classification accuracy and F1 scores.

5.4.3 CXR

With the CXR dataset, we used 522 labeled images and 4,694 unlabeled images. All the models were trained for 150 epochs and then evaluated. We generated an equal number of new images as the training set size. Figure 9 presents a visual comparison of a random selection of generated and real images. The FID and DDD measurements were performed for the distributions of generated and real training samples, indicating that more realistic images were generated by the MAVEN models than by the GAN

Fig. 7 Visual comparison of image samples from the CIFAR-10 dataset against those generated by the different models

Fig. 8 Histograms of the real CIFAR-10 training data, and of the data generated by the DC-GAN and VAE-GAN models and by our MAVEN models with mean and random feedback from 2, 3, to 5 discriminators

Table 3 Average cross-validation accuracy and class-wise F1 scores in the semi-supervised classification performance comparison of the DC-GAN, VAE-GAN, and MAVEN models using the CIFAR-10 dataset

Model	Accuracy	F1 scores									
		Plane	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
DC-GAN	0.713	0.760	0.840	0.560	0.510	0.660	0.590	0.780	0.780	0.810	0.810
VAE-GAN	0.743	0.770	0.850	0.640	0.560	0.690	0.620	0.820	0.770	0.860	0.830
MAVEN-m2D	0.761	0.800	0.860	0.650	0.590	0.750	0.680	0.810	0.780	0.850	0.850
MAVEN-m3D	0.759	0.770	0.860	0.670	0.580	0.700	0.690	0.800	0.810	0.870	0.830
MAVEN-m5D	**0.771**	0.800	0.860	0.650	0.610	0.710	0.640	0.810	0.790	0.880	0.820
MAVEN-r2D	0.757	0.780	0.860	0.650	0.530	0.720	0.650	0.810	0.800	0.870	0.860
MAVEN-r3D	0.756	0.780	0.860	0.640	0.580	0.720	0.650	0.830	0.800	0.870	0.830
MAVEN-r5D	0.762	0.810	0.850	0.680	0.600	0.720	0.660	0.840	0.800	0.850	0.820

and VAE-GAN models. The FID and DDD scores presented in Table 1 show that the mean MAVEN-m3D model has the smallest FID and DDD scores.

The classification performance reported in Table 4 suggests that our MAVEN model-based classifiers are more accurate than the baseline GAN and VAE-GAN classifiers. Among all the models, the MAVEN-m3D classifier was the most accurate.

5.4.4 SLC

For the SLC dataset, we used 160 labeled images and 1,440 unlabeled images. All the models were trained for 150 epochs and then evaluated. We generated new images equal in number to the training set size. Figure 10 presents a visual comparison of randomly selected generated and real image samples.

The FID and DDD measurements for the distributions of generated and real training samples indicate that more realistic images were generated by the MAVEN models than by the GAN and VAE-GAN models. The FID and DDD scores presented in Table 1 show that the mean MAVEN-m3D model has the smallest FID and DDD scores.

The classification performance reported in Table 5 suggests that our MAVEN model-based classifiers are more accurate than the baseline GAN and VAE-GAN classifiers. Among all the models, MAVEN-r3D is the most accurate in discriminating between non-melanoma and melanoma lesion images.

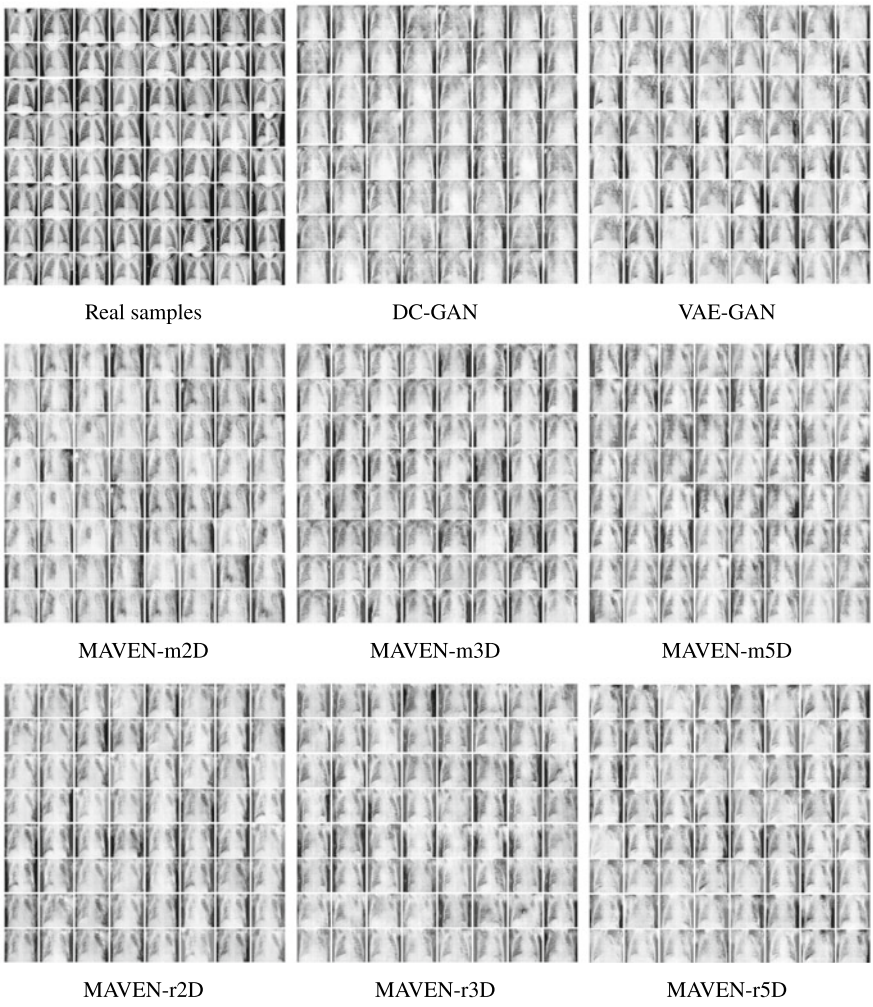


Fig. 9 Visual comparison of image samples from the CXR dataset against those generated by the different models

Table 4 Average cross-validation accuracy and class-wise F1 scores for the semi-supervised classification performance comparison of the DC-GAN, VAE-GAN, and MAVEN models using the CXR dataset

Model	Accuracy	F1 scores		
		Normal	B-Pneumonia	V-Pneumonia
DC-GAN	0.461	0.300	0.520	0.480
VAE-GAN	0.467	0.220	0.640	0.300
MAVEN-m2D	0.469	0.310	0.620	0.260
MAVEN-m3D	**0.525**	0.640	0.480	0.480
MAVEN-m5D	0.477	0.380	0.480	0.540
MAVEN-r2D	0.478	0.280	0.630	0.310
MAVEN-r3D	0.506	0.440	0.630	0.220
MAVEN-r5D	0.483	0.170	0.640	0.240

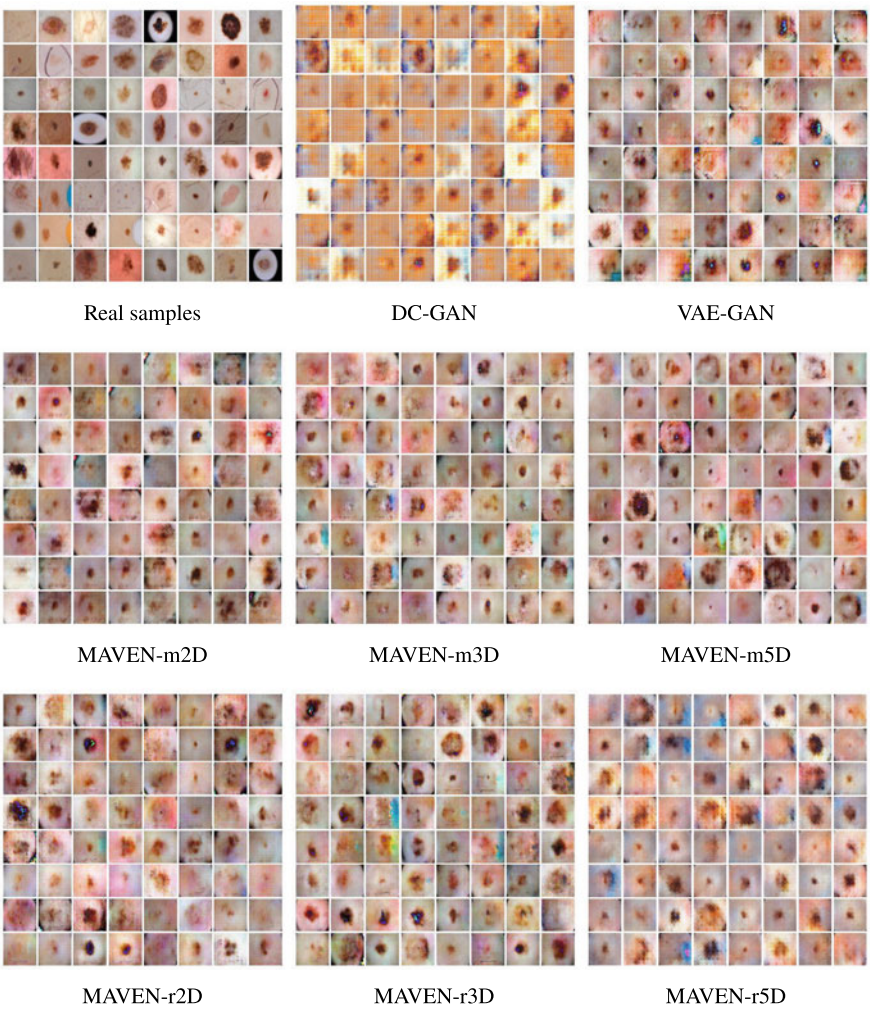


Fig. 10 Visual comparison of image samples from the SLC dataset against those generated by the different models

Table 5 Average cross-validation accuracy and class-wise F1 scores for the semi-supervised classification performance comparison of the DC-GAN, VAE-GAN, and MAVEN models using the SLC dataset

Model	Accuracy	F1 scores	
		Non-melanoma	Melanoma
DC-GAN	0.802	0.890	0.120
VAE-GAN	0.810	0.890	0.012
MAVEN-m2D	0.815	0.900	0.016
MAVEN-m3D	0.814	0.900	0.110
MAVEN-m5D	0.812	0.900	0.140
MAVEN-r2D	0.808	0.890	0.260
MAVEN-r3D	**0.821**	0.900	0.020
MAVEN-r5D	0.797	0.890	0.040

6 Conclusions

We have introduced a novel generative modeling approach, called Multi-Adversarial Variational autoEncoder Networks, or MAVENs, which demonstrates the advantage of an ensemble of discriminators in the adversarial learning of variational autoencoders. We have shown that training our MAVEN models on small, labeled datasets and allowing them to leverage large numbers of unlabeled training examples enables them to achieve superior performance relative to prior GAN and VAE-GAN-based classifiers, suggesting that MAVENs can be very effective in simultaneously generating high-quality realistic images and improving multiclass image classification performance. Furthermore, unlike conventional GAN-based semi-supervised classification, improvements in the classification of natural and medical images do not compromise the quality of the generated images. Future work with MAVENs should explore more complex image analysis tasks beyond classification and include more extensive experimentation spanning additional domains.

References

1. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN (2017). arXiv preprint arXiv:1701.07875
2. C. Bermudez, A.J. Plassard, L.T. Davis, A.T. Newton, S.M. Resnick, B.A. Landman, Learning implicit brain MRI manifolds with deep learning, in *Medical Imaging 2018: Image Processing*, vol. 10574 (2018), p. 105741L
3. F. Calimeri, A. Marzullo, C. Stamile, G. Terracina, Biomedical data augmentation using generative adversarial neural networks, in *International Conference on Artificial Neural Networks* (2017), pp. 626–634
4. M.J. Chuquicusma, S. Hussein, J. Burt, U. Bagci, How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis, in *IEEE International Symposium on Biomedical Imaging (ISBI)* (2018), pp. 240–244

5. N.C. Codella, D. Gutman, M.E. Celebi, B. Helba, M.A. Marchetti, S.W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler et al., Skin lesion analysis toward melanoma detection: a challenge at the 2017 ISBI, hosted by ISIC, in *IEEE International Symposium on Biomedical Imaging (ISBI 2018)* (2018), pp. 168–172
6. E.L. Denton, S. Chintala, A. Szlam, R. Fergus, Deep generative image models using a Laplacian pyramid of adversarial networks, in *Advances in Neural Information Processing Systems (NeurIPS)* (2015)
7. I. Durugkar, I. Gemp, S. Mahadevan, Generative multi-adversarial networks (2016). arXiv preprint arXiv:1611.01673
8. M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification (2018). arXiv preprint arXiv:1803.01229
9. 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 (NeurIPS)* (2014), pp. 2672–2680
10. J.T. Guibas, T.S. Virdi, P.S. Li, Synthetic medical images from dual generative adversarial networks (2017). arXiv preprint arXiv:1709.01872
11. C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, H. Nakayama, GAN-based synthetic brain MR image generation, in *IEEE International Symposium on Biomedical Imaging (ISBI)* (2018), pp. 734–738
12. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in *Advances in Neural Information Processing Systems (NeurIPS)* (2017), pp. 6626–6637
13. L. Hou, A. Agarwal, D. Samaras, T.M. Kurc, R.R. Gupta, J.H. Saltz, Unsupervised histopathology image synthesis (2017). arXiv preprint arXiv:1712.05021
14. A.A.Z. Imran, P.R. Bakic, A.D. Maidment, D.D. Pokrajac, Optimization of the simulation parameters for improving realism in anthropomorphic breast phantoms, in *Proceedings of the SPIE*, vol. 10132 (2017)
15. A.A.Z. Imran, D. Terzopoulos, Multi-adversarial variational autoencoder networks, in *IEEE International Conference on Machine Learning and Applications (ICMLA)* (oca Raton, FL, 2019), pp. 777–782
16. A.A.Z. Imran, D. Terzopoulos, Semi-supervised multi-task learning with chest X-ray images (2019). arXiv preprint arXiv:1908.03693
17. D.S. Kermany, M. Goldbaum, W. Cai, C.C. Valentim, H. Liang, S.L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan et al., Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell **172**(5), 1122–1131 (2018)
18. D.P. Kingma, M. Welling, Auto-encoding variational Bayes (2013). arXiv preprint arXiv:1312.6114
19. A. Krizhevsky, Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, Dept. of Computer Science (2009)
20. A. Madani, M. Moradi, A. Karargyris, T. Syeda-Mahmood, Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation, in *IEEE International Symposium on Biomedical Imaging (ISBI)* (2018), pp. 1038–1042
21. A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders (2015). arXiv preprint arXiv:1511.05644
22. T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks (2018). arXiv preprint arXiv:1802.05957
23. G. Mordido, H. Yang, C. Meinel, Dropout-GAN: learning from a dynamic ensemble of discriminators (2018). arXiv preprint arXiv:1807.11346
24. Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, vol. 2011 (2011), pp. 1–9
25. B. Neyshabur, S. Bhojanapalli, A. Chakrabarti, Stabilizing GAN training with multiple random projections (2017). arXiv preprint arXiv:1705.07831

26. T. Nguyen, T. Le, H. Vu, D. Phung, Dual discriminator generative adversarial nets, in *Advances in Neural Information Processing Systems (NeurIPS)* (2017), pp. 2670–2680
27. A. dena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs (2017). arXiv preprint arXiv:1610.09585
28. G. Ostrovski, W. Dabney, R. Munos, Autoregressive quantile networks for generative modeling (2018). arXiv preprint arXiv:1806.05575
29. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks (2015). arXiv preprint arXiv:1511.06434
30. S. Ravuri, S. Mohamed, M. Rosca, O. Vinyals, Learning implicit generative models with the method of learned moments (2018). arXiv preprint arXiv:1806.11006
31. H. Salehinejad, S. Valaee, T. Dowdell, E. Colak, J. Barfett, Generalization of deep neural networks for chest pathology classification in X-rays using generative adversarial networks, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 990–994
32. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, in *Advances in Neural Information Processing Systems (NeurIPS)* (2016), pp. 2234–2242
33. J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks (2015). arXiv preprint arXiv:1511.06390
34. T. Unterthiner, B. Nessler, C. Seward, G. Klambauer, M. Heusel, H. Ramsauer, S. Hochreiter, Coulomb GANs: provably optimal nash equilibria via potential fields (2017). arXiv preprint arXiv:1708.08819
35. S. Wang, L. Zhang, CatGAN: coupled adversarial transfer for domain generation (2017). arXiv preprint arXiv:1711.08904
36. M.A. Wani, F.A. Bhat, S. Afzal, A.I. Khan (eds.), *Advances in Deep Learning* (Springer, 2020)