This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

Semi-supervised learning based on GAN with mean and variance feature matching

Cong Hu, Xiao-jun Wu*, Josef Kittler

Abstract—Improved generative adversarial network (Improved GAN) is a successful method by using generative adversarial model to solve the problem of semi-supervised learning. Improved GAN learns a generator with the technique of mean feature matching which penalizes the discrepancy of the first order moment of the latent features. To better describe common attributes of a distribution, the paper proposes a novel semi-supervised learning method which incorporates the first order moment and the second order moment of the features in an intermediate layer of the discriminator, called mean and variance feature matching GAN(MVFM-GAN). To capture more precisely the data manifold, not only mean but also variance is used in the latent feature learning. Compared with improved gan and other traditional methods, MVFM-GAN shows its superior performance of semi-supervised classification and the stability of GAN training, particularly in the cases when the number of labelled samples is low. It shows the comparable performance with the state-of-the-art methods on several benchmark datasets. As a byproduct of the novel approach, MVFM-GAN can generate realistic images with good visual quality.

Index Terms—Semi-supervised learning, generative adversarial networks, feature matching, neural network, image recognition.

1 Introduction

It is well-known that supervised learning models for deep learning, such as convolutional neural networks(CNN) and long short term memory (LSTM) networks, have recently been advancing dramatically, enabling successful applications to many computational tasks including object recognition [1], [2], [3], [4], speech recognition [5], [6], [7], image caption generation [8], [9], [10], machine translation [11], [12], medical detection [13], video restoration [14], image synthesis [15] and so on. Some traditional methods also have achieved great performance on several visual tasks, such as image synthesis [16], [17], [18], [19], image representation [20], [21], landmark detection [22], [23] and so on. Despite these successes, these supervised learning methods have one common bottleneck, namely it is expensive and time-consuming to obtain enough labelled samples to train the deep model and capture the intrinsic structure of the data. Consequently, semi-supervised learning(SSL), which learns from a combination of unlabelled data and few labelled data for better performance than using the labelled data alone, has attracted considerable attention.

In this work we focus on generative models in SSL due to their ability to capture the salient properties and structure of data. Deep generative models are particularly appealing because they are capable of learning a latent manifold on which the data has high density. Learning this manifold allows smooth variations in the latent space to result in meaningful transformations in the original space, effectively traversing between high density modes through low density areas. The generative adversarial network approach [24] is a framework for training generative models, which we briefly review. The architecture of GAN is shown as figure 1. It consists of two networks pitted against one another in a two player game: A generative model, G, is trained to synthesize images resembling the data distribution and a discriminative model, D, is trained to distinguish between samples drawn from G and images drawn from the training data. The generator generates unlabeled realistic samples from the latent model to improve the discriminate ability of the discriminator. More representative estimates also are obtained by using additional unlabeled samples.

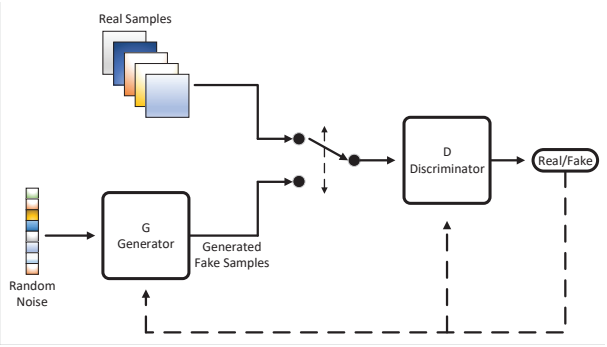


Fig. 1. The architecture of GAN.

The Improved GAN with mean feature matching(FM) achieved a state-of-the-art performance in semi-supervised learning. Distinguishing two distributions by finite samples is known as Two-Sample Test in statistics. Our proposed method MVFM-GAN, as a kind of generative methods, is used for semi-supervised learning based on this principle

- *Cong Hu and Xiao-Jun Wu are with the School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, Jiangsu Province, China. Cong Hu and Xiao-Jun Wu are also with Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China. E-mail: wxhucong@163.com; wu_xiaojun@jiangnan.edu.cn*
- *Josef Kittler is with Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK. E-mail: j.kittler@surrey.ac.uk*

This work is supported by the National Natural Science Foundation of China under Grant No.: 61373055, 61672265, and the 111 Project of Chinese Ministry of Education under Grant No.: B12018.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

of statistics. The hypothesis that the real and fake samples sampled from the same latent model is used. Real and fake samples linked by estimating the parameters of the latent model, such as mean and variance. To better describe the geometry of a probability distribution, the second moments variances are usually employed in practice. Inspired by MMD and mean feature matching, we propose to embed distributions in a finite dimensional feature space and to match them based on not only their means but also variances feature statistics, since incorporating first and second order statistics has a better chance to capture the various modes of the distribution. The two reasons why our method could improve the GAN can summarized as follow: 1) The additional unlabelled data generated from our methods MVFM-GAN improved the representative ability of the model. 2)Combining the first-order moment mean and the second-order moment variance can capture more precisely the data manifold and the geometry of the probability distribution. By comparison with various strong competitors including improved GAN, we show that our proposed method MVFM-GAN achieves the state-of-the-art results in semi-supervised learning on MNIST, SVHN, CIFAR-10 and STL-10 benchmark datasets.

The major contributions of this paper lie in:

(1)In our proposed MVFM-GAN, the variance discrepancy as a loss function is used for minimizing the distributional difference between the generated data and the real data.

(2)It is very easy to implement in the generative adversarial nets for the proposed loss function. The standard SGD and Adam can be directly used for optimizing our MVFM-GAN model.

(3)In terms of semi-supervised learning, we verify the superior performance of our proposed method than the improved GAN and other traditional methods.

2 Related Work

2.1 Literature Review

Originally, semi-supervised learning came to prominence in the 1970s. The earliest recorded approach is named self-learning, an iterative procedure where some unlabelled samples are labelled by the best predictions made by the supervised model, thereby providing more training samples for the supervised learning algorithm. Blum and Mitchell's co-training [25] offers an approach, where two models are trained on two separate subsets of the data features. The labels of samples predicted with confidence by one model are then used for supervised training of the other model.

In the late 1990s, transductive Support Vector Machine (TSVM) [26] was a popular technique. Similar to the regular SVM, TSVM aims to maximise the margin between the training samples and the decision boundary. Additionally, the distance of unlabelled data to the margin was maximised simultaneously. However, this kind of semi-supervised SVM optimisation problem is difficult to solve because it is non-convex. A graph-based approach was proposed as another concept for SSL. In the method, a graph is constructed by connecting samples by edges attributed by some measure of similarity. A label propagation mechamism [27] was then used to minimise the difference between the predicted labels for nodes with heavily weighted edges. By virtue of this process, label information propagates from the labelled samples to unlabelled samples in their neighbourhood.

Later, unsupervised learning evolved to deep learning methods such as autoencoders [28] [Hinton (2009)]. These methods are often used to extract features from the data in an unsupervised fashion. The labels provided are then used to train a classifier in the derived feature space. Since the unsupervised feature learning and the supervised learning stages were already decoupled, such deep learning approaches can naturally be extended to SSL, where one can simply train the predictor model with just the smaller, labelled data subset in the supervised stage. Because of this synergy, autoencoders were always involved in neural network based SSL. Pseudolabel [29] is an early SSL approach that made use of autoencoders and it is essentially a self-learning method. Prior to each iteration, the current most confident prediction of the model temporarily labelles the unlabelled samples and then the model is updated on the combined labelled and unlabelled data. Ladder Network [30] is another SSL approach offering competitive performance, whose starting point is essentially an autoencoder. Variational autoencoders (VAE) [31] is another autoencoders-based method used in some SSL related works. One successful example is the auxiliary deep generative model(ADGM) [32], which extends VAE with auxiliary variables to improve the variational approximation. The auxiliary variables leave the generative model unchanged but make the variational distribution more expressive. Virtual adversarial training(VAT) [33] is a regularization approach based on virtual adversarial loss, which measures the local smoothness of the output distribution. VAT aims to find the optimal adversarial perturbation of a real data input and maximizes the KL divergence between the output of the original samples and that of perturbed samples.

Recently, the Generative Adversarial Networks (GANs) [24] have been shown to exhibit promising performance in unsupervised learning and semi-supervised learning. GANs learn generative models based on the theory of Nash Equilibrium. The GANs learn two sub-networks: a generator and a discriminator. The generator transforms noise z to $x = G(z;\theta^{(G)})$ in order to generate data consistent with the real data distribution $p_{data}(x)$, fooling the discriminator into accepting the generated data as being real. The discriminator is trained to reveal whether a sample is generated or real. The original work showed that in GAN this objective is defined by the Jensen-Shannon divergence. Other φ-divergences were successfully used in [34]. the Max-margin deep generative model(MMCVA) [35] was proposed to improve the predictive performance of deep generative models with the discriminative principle of max-margin learning. On the other hand, the Adversarially Learned Inference(ALI) [36] architecture was designed to learn an inference model during the GAN training process. It also achieved competitive SSL results. Triple GAN [37] used the data produced by the GAN generator as additional training data for improving the SSL performance. CatGAN [38] provided an approach for training a classification model and GAN for SSL simultaneously by introducing a categorical cross-entropy loss term. The badGAN [39] generated the "bad" samples in low density regions where

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

the training data is rare based on the low density separtion assumption. Images with random patches removed are used as the input of generator of CC-GAN [40]. It achieved good performance in semi-supervised learning.

The Maximum Mean Discrepancy objective(MMD) for GAN training was proposed in [41], [42], named GMMN. MMD is a centerpiece of non-parametric two-sample test to determine the distance distribution. During the training of GAN, the generator is trained to test the hypothesis that the generated sample satisfies the MMD distance. Inspired by MMD, the Improved GAN [43] demonstrated an excellent performance in SSL, showing that one can train the GAN discriminator using the objective of vanilla GAN while training the generator using L_q mean feature matching. In the method, two vectors of average features are given in a latent layer, one from the real data and one from the fake data. The L_q norm of the difference between these vectors is then added as a cost to the generator. In this way, the generator gains an additional training signal that encourages it to produce images whose features, according to the discriminator, match those of real images.

2.2 Maximum Mean Discrepancy

Given two datasets $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$, we wish to consider the question of proving whether the generating distributions are the same, i.e. $P_X = P_Y$. The two-sample test [44] is a useful method for addressing this kind of problem. One of the most successful methods is an estimator known as the maximum mean discrepancy(MMD). MMD compares the means between the two sets of samples. If the means of the two datasets are similar then they are likely come from the same generating distribution. The objective of MMD is given by

$$L_{MMD} = \| E(\phi(x_i)) - E(\phi(y_j)) \|_q \tag{1}$$

where $\phi(\bullet)$ is a vector embedding function which maps sample $\bullet$ to a feature space. To match the mean statistics of the two distributions, any l_q norm $\| \bullet \|_q$ can be used. Taking ϕ to be the identity function leads to matching the sample mean. Minimizing MMD is equivalent to minimizing a distance between the two distributions. If and only if $P_X = P_Y$, MMD is equal to 0 [44], [45].

l_2 norm is adopted to measure the MMD. The mean squared error of the expectations of the two distributions P_X and P_Y is shown as:

$$\begin{aligned} L_{MMD^2} &= \| E(\phi(x_i)) - E(\phi(y_j)) \|_2^2 \\ &= \| \frac{1}{N}\sum_{i=1}^{N}\phi(x_i) - \frac{1}{M}\sum_{j=1}^{M}\phi(y_j) \|_2^2 \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\phi(x_i)^T\phi(x_{i'}) \\ &\quad - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\phi(x_i)^T\phi(y_j) \\ &\quad + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M}\phi(y_j)^T\phi(y_{j'}). \end{aligned} \tag{2}$$

3 Mean and variance feature matching GAN(MVFM-GAN)

3.1 Max variance discrepancy

MMD compares two distributions by comparing the means of their feature embedding. However, relying just on the first order statistics is a rather simplistic way to compare two distributions. Our aim is to extend the distribution discrepancy measure so as to reflect also second order statistics, i.e variance information of feature embeddings.

In order to make the two distributions to be as close as possible, the statistics that we should consider is not only the first order moment - mean, but also the second order moment - variance. Inspired by the MMD in equation(1), in this paper, we propose the Maximum variance discrepancy(MVD) as follows,

$$L_{MVD} = \| Var(\phi(x_i)) - Var(\phi(y_j)) \|_q \tag{3}$$

where, the vector of variances of the feature space embedding of the input data, $Var(\bullet)$, is defined as

$$Var(\phi(x_i)) = E[\phi(x_i) - E(\phi(x_i)]^2, \tag{4}$$

$$Var(\phi(y_j)) = E[\phi(y_j) - E(\phi(y_j)]^2. \tag{5}$$

3.2 Mean and variance feature matching

To improve the instability of GANs, feature matching(FM) aims to prevent the generator from over training on the discriminator by assigning a novel objective to the generator. Different from the objective of vanilla GAN by maximizing the output of the discriminator, this novel objective forces the generator to produce data which can match the statistics of the real data. And the statistics that are worth matching will be specified as the objective. One natural choice of statistics is mean value, matching the expected value of the features of the real data and that of generated data. In practise, mean statistic can not well effectively describe the discrepancy of different distributions. Adding second order information would enrich the discrimination power of the feature space. Here, we incorporate both the first and second order moment of every embedding dimensionality by using the linear kernel to match the mean and variance of features, named maximum mean and variance discrepancy. The mean squared difference of the mean and variance of the two sets of samples are used to train the generator. Letting $\phi(x)$ denote activations on an intermediate layer of the discriminator, our new objective for the generator is defined as:

$$\begin{aligned} L = &\| E(\phi(x_i)) - E(\phi(y_j)) \|_q \\ &+ \lambda \| Var(\phi(x_i)) - Var(\phi(y_j)) \|_q \end{aligned} \tag{6}$$

where $\| \bullet \|_q$ denotes the l_q norm, $E(\bullet)$ and $Var(\bullet)$ denote the means and variances of the feature embeddings of data.

3.3 semi-supervised classification

Suppose a sample x is classified into one of the K possible classes. Softmax can turn x into the class probabilities $p_{model}(y = j|x) = \frac{exp(x_j)}{\sum_{k=1}^{K} exp(x_k)}$. To train supervised model, the objective minimizes the cross-entropy between the predictive distribution $p_{model}(y|x)$ with the target labels. By

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

adding another target class $K+1$ into the classes, the generated samples that labelled with the new class $y = K+1$ can be mixed with the real data as the new dataset to train a semi-supervised model. Corresponding to the probability $1 - D(x)$ in the original GAN, $p_{model}(y = K+1|x)$ is used for representing the probability that x is fake. To maximize $logp_{model}(y \in \{1, ..., K\}|x)$, we train the model with unlabelled real data until these data correspond to the K classes. As the same as the training manner of traditional GAN, the detailed procedure of MVFM-GAN are carried out in two steps.

At the first step, the discriminator D is optimized by minimizing the objective $L(D)$,

$$D^* = \underset{D}{argmin}\, L(D). \tag{7}$$

The cost function to train the classifier is as follow:

$$\begin{aligned} L(D) =& - E_{x,y\sim p_{data}(x,y)}[logp_{model}(y|x)] \\ & - E_{x\sim G}[logp_{model}(y = K+1|x)] \\ =& L_{sup} + L_{unsup} \end{aligned} \tag{8}$$

where,

$$L_{sup} = -E_{x,y\sim p_{data}(x,y)} logp_{model}(y|x, y < K+1), \tag{9}$$

$$\begin{aligned} L_{unsup} =& - \{E_{x,y\sim p_{data}(x,y)} log[1 - p_{model}(y = K+1|x) \\ & + E_{x\sim G} log[p_{model}(y = K+1|x)]\}. \end{aligned} \tag{10}$$

The cost function consists of two parts, supervised loss L_{sup} and unsupervised L_{unsup}. L_{sup} is a standard supervised loss function where we train the real labelled data to fit the negative log probability of their corresponding labels. $p_{model}(y = K+1|x)$ denotes the probability that x is fake data. And L_{unsup} is in fact the standard loss of original GAN and $D(x)$ is represented with $1 - p_{model}(y = K+1|x)$, the unsupervised loss is presented as follows:

$$\begin{aligned} L_{unsup} =& - \{E_{x\sim p_{data}(x)} logD(x) \\ & + E_{z\sim noise} log(1 - D(G(z)))\}, \end{aligned} \tag{11}$$

$D(x)$ denotes the probability that x is real data. By minimizing L_{sup} and L_{unsup} jointly, we train the semi-supervised model to estimate this optimal solution. The L_{sup} and L_{unsup} are estimated by training samples as follows:

$$L_{sup} = -\frac{1}{S}\sum_{s=1}^{S} Z^l_{s,j=y_s} + \frac{1}{S}\sum_{s=1}^{S} f(Z^l_s), \tag{12}$$

$$\begin{aligned} L_{unsup} =& -\frac{1}{U}\sum_{u=1}^{U} f(Z^l_u) + \frac{1}{U}\sum_{u=1}^{U} log_e(1 + e^{f(Z^l_u)}) \\ & + \frac{1}{R}\sum_{r=1}^{R} log_e(1 + e^{f(Z^l_r)}), \end{aligned} \tag{13}$$

where,

$$f(Z^l_i) = max(Z^l_{ij}) + \sum_{j=1}^{K^l} e^{Z^l_{ij} - max(Z^l_{ij})} \tag{14}$$

where, Z^l_i denotes (the output of ith sample in the lth layer). S, U and R respectively denote the number of labelled, unlabelled and generated data. Z^l_{ij} denotes the value of jth unit of ith sample's output(Z_i) in lth layer. y_s is the label index of sth labelled sample. K^l denotes the dimension of the lth layer. We use the last layer as the target layer when training the discriminator D, and K^l is equal to the number the classes K.

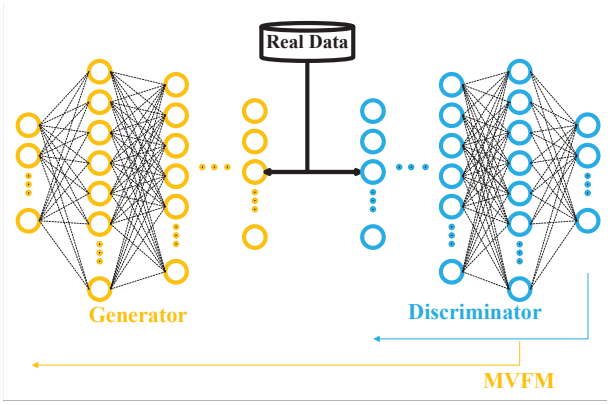


Fig. 2. The architecture of our proposed method MVFM-GAN.

At the second step, the generator G is then optimized by applying the optimized discriminator D to G. Here, we train the G using feature matching to force G to approximate the real data generating distribution, with regularizing the discrepancy of the statistics value of the embeddings. The following objective computes the mean squared difference of the means and variances of the activations of every dimension in a certain latent layer of the discriminator.

$$G^* = \underset{G}{argmin}\, L(G), \tag{15}$$

where,

$$\begin{aligned} L(G) =& \parallel E_{x\sim p_{data}(x)}(\phi(x)) - E_{z\sim noise}(\phi(G(z))) \parallel_q^2 \\ & + \lambda \parallel Var_{x\sim p_{data}(x)}(\phi(x)) - Var_{z\sim noise}(\phi(G(z))) \parallel_q^2, \end{aligned} \tag{16}$$

any L_q norm can be used to optimize the generator, in this paper, we use the L_2 norm. λ is the trade-off parameter. The $L(G)$ is estimated as follow:

$$\begin{aligned} L(G) =& \frac{1}{K^l}\sum_{j=1}^{K^l}(\frac{1}{U}\sum_{u=1}^{U} Z^l_{uj} - \frac{1}{R}\sum_{r=1}^{R} Z^l_{rj})^2 \\ & + \frac{\lambda}{K^l}\sum_{j=1}^{K^l}[\frac{1}{U}\sum_{u=1}^{U}(Z^l_{uj} - \frac{1}{U}\sum_{u=1}^{U} Z^l_{uj})^2 \\ & - \frac{1}{R}\sum_{r=1}^{R}(Z^l_{rj} - \frac{1}{R}\sum_{r=1}^{R} Z^l_{rj})^2], \end{aligned} \tag{17}$$

where, K^l denotes the dimension of lth layer. A certain layer of discriminator is used for the target layer when training the generator. U denotes the number of unlabelled real data, and R is the number of generated data. In all the experiments of this paper, we set $U = R$. The architecture of MVFM-GAN is shown in Fig.2. In Algorithm 1, we summarize the learning details of MVFM-GAN, and Adam [46] is used to optimize our model.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

Algorithm 1 The MVFM-GAN training algorithm

Input: Training dataset $X = \{X_{label}, X_{unlabel}\}$, $X_{label} = \{(x_1, y_1), (x_2, y_2), ..., (x_S, y_S)\}$, $X_{unlabel} = \{x_1, x_2, ..., x_U\}$. Initialize all the parameters $\Theta = \{\Theta_g, \Theta_d\}$ in generator and discriminator, trade-off hyperparameter λ and Adam hyperparameter α. The number of iteration t$\leftarrow$ 0.
Output: $\Theta = \{\Theta_g, \Theta_d\}$
1: **while** θ_g does not converge do.
2: t$\leftarrow$ t+1.
3: Compute the cost of $L^t(D)$ by $L^t(D) \leftarrow L^t_{sup}(X_{label}) + L^t_{unsup}(X)$ with equations(8), (12) and (13).
4: Compute the backpropagation error to optimize discriminator $\Theta^t_d \leftarrow Adam(\nabla_{\theta^t_d} L^t(D), \alpha)$.
5: Sample noisy data $z \sim p(z)$, Generated data $X_g = \{G(z_1), G(z_2), ..., G(z_R)\}$.
6: Compute the cost of mean and variance feature matching $L^t(G)$ with equations(17).
7: Fix the discriminator parameter Θ^t_d and compute the backpropagation error to optimize generator $\Theta^t_g \leftarrow Adam(\nabla_{\theta^t_g} L^t(G), \alpha)$.
8: **end while**

TABLE 1
Network architecture used for MNIST

Stage	Layer	Layer Type
Generator	0	Input Noise 100 units
	1	Dense layer 500 units+softPlus+BatchNorm
	2	Dense layer 500 units+softPlus+BatchNorm
	3	Output 784 units+sigmoid+Scaling
Discriminator	4	Input layer 784 units+Gaussian noise $\sigma = 0.3$
	5	Dense Layer 1000 units+ReLU+Gaussian noise $\sigma = 0.5$
	6	Dense Layer 500 units+ReLU+Gaussian noise $\sigma = 0.5$
	7	Dense Layer 250 units+ReLU+Gaussian noise $\sigma = 0.5$
	8	Dense Layer 250 units+ReLU+Gaussian noise $\sigma = 0.5$
	9	Dense Layer 250 units+ReLU+Gaussian noise $\sigma = 0.5$
	10	Output Layer (K classes)units+Softmax

4 Experiments

We perform semi-supervised experiments and sample generation experiments on four benchmark datasets including MNIST [47], CIFAR-10 [48], SVHN [49] and STL-10 [50].

4.1 MNIST

MNIST is a well-known handwritten digits dataset. In the first experiment, we train the proposed method on this standard benchmark dataset. This dataset comrpises samples of digits 0 to 9(10-classes), in the form of 28×28 black and white images. There are 60,000 training images and 10,000 test images. Before we input these images to our model, the pixel values are scaled to the [0,1] range. Two fully connected networks are used for the discriminator and the generator in the original Improved GAN [43] paper. A batch normalization and Gaussian noise are added to the output of each layer. An overview of the networks is shown in Table 1.

To evaluate the performance of our method in semi-supervised learning, we consider four sets labelled samples of size 50, 100, 200 and 1000, respectively. The classification results are averaged over 10 runs with labelled subsets chosen at random, having a balanced number of samples from each class. The remaining samples are used for trained without labels. Fig.3 compares the classification error of different versions of training obtained with Improved GAN(FM) and our method MVFM-GAN. From the results, we can easily see that our method achieves a better rate of semi-supervised classification and a greater stability of the GAN training, especially when only a few labelled data is available. This suggests that constraining both the mean and variance of the features in the latent space can more easily model the manifold of the data and regularize the distribution. We also compare our method with other

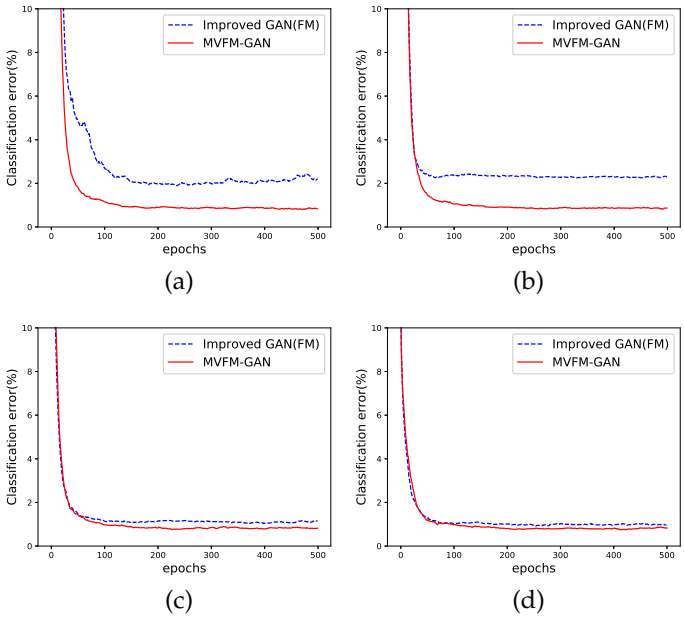


Fig. 3. MNIST classification error rate(%) evaluated after every epoch for Improved GAN(FM) and MVFM-GAN† in semi-supervised learning.(a)50 labelled samples,(b)100 labelled samples,(c)200 labelled samples,(d)1000 labelled samples.

SSL methods and the results are summarized in Table 2. Our method achieves the best semi-supervised classification performance also among these method expect the BadGAN. But our method is more stable than the BadGAN.

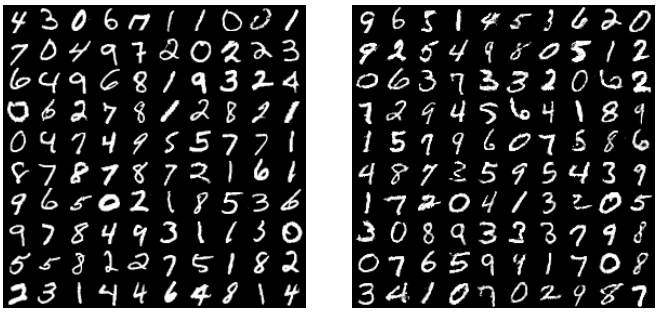
Some original samples and those generated during semi-supervised learning using our method are shown in Fig. 4. The generated samples look visually appealing and have good visual quality.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

TABLE 2
Percentage of incorrectly predicted test samples for a given number of labelled samples on the MNIST data set

Model	Error rates(%)	
	N_L	100
SVM		23.44
TSVM		16.81
Pseudolabel		10.49
VAE+SVM		11.82±0.25
VAE(M1+M2)		3.33±0.14
Ladder Network		1.06±0.037
Conv-Ladder Network		0.89±0.50
Skip DGM		1.32±0.07
Auxiliary DGM		0.96±0.02
VAT		1.36
MMCVA		1.24±0.54
Triple GAN		0.91
BadGAN		**0.795±0.098**
Cat-GAN		1.91±0.1
Improved GAN		0.93±0.065
Our model		0.81±0.04



(a) Original Images (b) Generated Images

Fig. 4. Comparision of original images(left) and generated images(right) on MNIST with MVFM-GAN.

4.2 CIFAR-10

CIFAR-10 is an established computer-vision dataset used for object recognition. It consists of 60,000 32×32 color images containing one of 10 object classes, with 6000 images per class. We use this data set to study semi-supervised learning, as well as to examine the visual quality of generated samples that can be achieved. The network architecture of MVFM-GAN, shown in Table 3, as same as that of the original Improved GAN.

We train the semi-supervised MVFM-GAN model on sets of labelled samples of size 50,100,200,400,800 and 1000 per class The remaining samples are left unlabelled. Fig.5 compares the classification error obtained using Improved GAN(FM) and MVFM-GAN. We then compare our method with other methods and the experimental results on this dataset are reported by averaging the classification error over ten runs. Table 4 summarizes our results on the semi-supervised learning task compared with other methods. From the results, we can easily see our method achieves better performance, with the error rate reduction of over 2% compared to Improved GAN. Finally, by training the model, some fake samples are generated as a by-product and these generated samples look visually appealing and are of good visual quality. Some original samples and samples generated during semi-supervised learning using MVFM-GAN are shown in Fig. 6.

TABLE 3
Network architecture used for CIFAR-10

Stage	Layer	Layer Type
Generator	0	Input Noise 100 units
	1	Dense layer 4*4*512 units+softPlus+BatchNorm
	2	Transposed Conv2D Layer Target Size(256,8,8)+Filter size(5,5) +strides(2,2)+BatchNorm
	3	Transposed Conv2D Layer Target Size(256,8,8)+Filter size(5,5) +strides(2,2)+BatchNorm
	4	Transposed Conv2D Layer Target Size(128,16,16)+Filter size(5,5) +strides(2,2)+BatchNorm
	5	Conv2D Layer Target Size(3,32,32)+Filter size(5,5) +strides(2,2)+BatchNorm
Discriminator	6	Input layer Size(3,32,32)
	7	Dropout Layer dropout probability=0.2
	8	Conv2D Layer Size(96,32,32), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	9	Conv2D Layer Size(96,32,32), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	10	Conv2D Layer Size(96,16,16), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	11	Dropout Layer dropout probability=0.5
	12	Conv2D Layer Size(192,16,16), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	13	Conv2D Layer Size(192,16,16), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	14	Conv2D Layer Size(192,8,8), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	15	Dropout Layer dropout probability=0.5
	16	Conv2D Layer Size(192,6,6), Filter size(3,3),Strides(2,2) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	17	NIN Layer Size(192,6,6) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	18	NIN Layer Size(192,6,6) +ReLU+Gaussian noise $\sigma = 0.05$ +weight Norm
	19	Global pool layer 192 units
	20	Dense Layer (K classes) units + weight norm+softmax

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

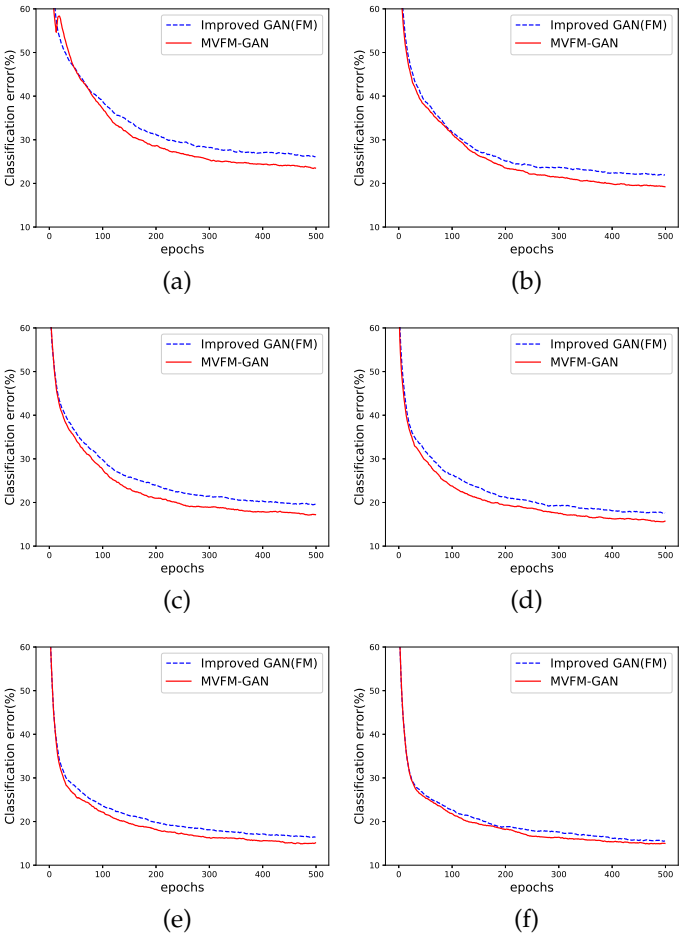


Fig. 5. CIFAR-10 classification error rates(%) evaluated after every epoch for Improved GAN(FM) and MVFM-GAN† in semi-supervised learning.(a)500 labelled samples,(b)1000 labelled samples,(c)2000 labelled samples,(d)4000 labelled samples,(e)8000 labelled samples,(f)10000 labelled samples.

TABLE 4
Percentage of incorrectly predicted test samples for a given number of labelled samples on the CIFAR-10 data set

Model	Error rate(%)	
	N_L	4000
Ladder network		20.40±0.47
ALI		17.99±1.62
CatGAN		19.58±0.46
Triple GAN		16.99±0.36
Improved GAN		18.63±2.32
Our model		**16.28±1.91**

4.3 SVHN

SVHN is a real-world dataset for an image recognition task. The SVHN dataset has 73,257 training samples and 26032 test points. We used the same architecture and experimental setup as for CIFAR-10 in Table 3. We train the semi-supervised MVFM-GAN model in several experiments, with labelled datasets of size 50,100,200 and 500 samples per class. The remaining samples are left unlabelled. Fig.7 compares the SVHN classification error obtained by training with Improved GAN(FM) and MVFM-GAN. The performance of the MVFM-GAN system is much better

(a) Original Images (b) Generated Images

Fig. 6. Comparision of original images(left) and generated images(right) on CIFAR-10 with MVFM-GAN.

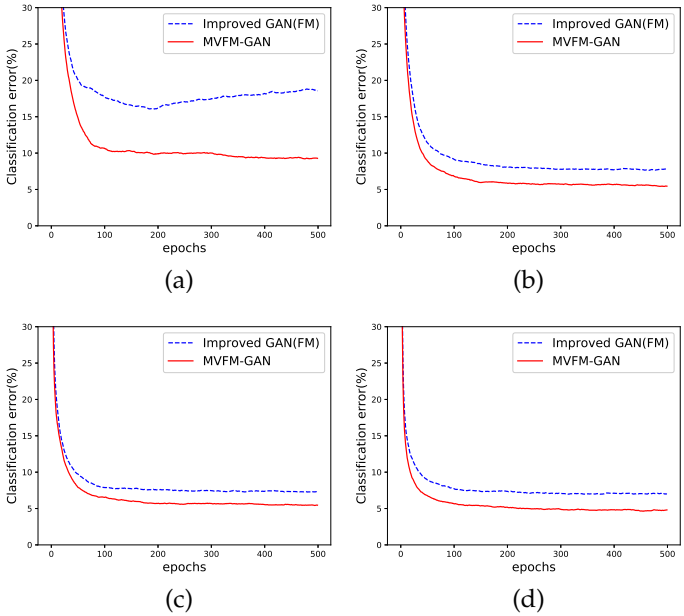


Fig. 7. SVHN dataset classification error rate(%) evaluated after every epoch for Improved GAN(FM) and MVFM-GAN† in semi-supervised learning.(a)500 labelled samples,(b)1000 labelled samples,(c)2000 labelled samples,(d)5000 labelled samples.

than that of the original Improved GAN, especially in the case where there are only 50 labelled samples per class. We then compared our method with other methods. The experimental results on this dataset are reported by averaging over ten runs. Table 5 summarizes our results in this semi-supervised learning task. From the results, we can see that our method exhibits superior performance. The fake samples generated as a by-product of the proposed semi-supervised learning method are of good visual quality. Fig. 8 shows some original samples and generated samples.

4.4 STL-10

STL-10 is a dataset of 96×96 color images with a 1:100 ratio of labelled to unlabelled examples, making it an ideal fit for our semi-supervised learning framework. The training set consists of 5000 labeled images, and 100,000 unlabelled images. The labeled images belong to 10 classes and were extracted from the Imagenet dataset and the unlabelled images come from a broader distribution of classes. Each class has 800 testing images. We extensively modified a

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

TABLE 5
Percentage of incorrectly predicted test samples for a given number of labeled samples on the SVHN data set

Model	Error rate(%)
	N_L 1000
TSVM	66.55
VAE(M1+M2)	36.02±0.10
ALI	7.41±0.65
Auxiliary DGM	22.86
Skip DGM	16.61±0.24
Improved GAN	8.11±1.3
Our model	**6.56±0.87**

(a) Original Images (b) Generated Images

Fig. 8. Comparision of original images(left) and generated images(right) on SVHN with MVFM-GAN.

publicly available implementation of DCGAN using TensorFlow to achieve high performance, using a multi-GPU implementation. MVFM-GAN learns some image statistics and generate contiguous shapes with natural color and texture but also learn some realistic objects. Some original samples and samples generated during semi-supervised learning using MVFM-GAN are shown in Fig. 9. We then evaluate classification accuracy of each model on the test set. The result is shown in table 6. Our method has comparable performance with the state-of-the-art method CC-GAN.

5 Conclusion

We develop mean and variance feature matching(MVFM) objective function for semi-supervised learning that incorporates the first and the second order moment of activations in the latent feature space. The aim is to capture the manifold of the data and improve the training stability of GAN. The motivation of our method is very intuitive

(a) Original Images (b) Generated Images

Fig. 9. Comparision of original images(left) and generated images(right) on STL-10 with MVFM-GAN.

TABLE 6
Percentage of incorrectly predicted test samples for a given number of labelled samples on the STL-10 data set

Model	Error rate(%)
	N_L 5000
SSL-GAN	26.19±0.5
CC-GAN	**22.21±0.8**
Improved GAN	25.63±0.7
Our model	23.19±0.5

and it is a natured development of improved GAN [43]. Empirically, it outperforms Improved GAN and all other baselines by a significant margin and establishes the state-of-the-art results in semi-supervised learning on several benchmarking datasets including MNIST, CIFAR-10, SVHN and STL-10. The research results indicate that the proposed method is a simple but effective method for semi-supervised learning. As a by-product, MVFM-GAN generates realistic images with good visual quality as the by-product can be generated after training MVFM-GAN.

In future, a theoretical analysis of our method will be carried out and the potential for its beneficial combination with other generative models will be explored.

References

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

[2] Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu, "Place classification with a graph regularized deep neural network," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 304–315, 2017.

[3] C. Hu, X.-J. Wu, and Z.-Q. Shu, "Discriminative feature learning via sparse autoencoders with label consistency constraints," *Neural Processing Letters*, pp. 1–13, 2018.

[4] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 356–367, 2017.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] J. S. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan, "Look, listen and learn-a multimodal lstm for speaker identification." in *AAAI*, 2016, pp. 3581–3587.

[8] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[9] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[10] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *International Conference on Machine Learning*, 2014, pp. 595–603.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

[13] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images," *IEEE transactions on medical imaging*, vol. 35, no. 1, pp. 119–130, 2016.

[14] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," *arXiv preprint arXiv:1806.08482*, 2018.

[15] H. Wang, N. Schor, R. Hu, H. Huang, D. Cohen-Or, and H. Huang, "Global-to-local generative model for 3d shapes," *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, vol. 37, no. 6, p. 214:1214:10, 2018.

[16] K. Li, Q. Dai, R. Wang, Y. Liu, F. Xu, and J. Wang, "A data-driven approach for facial expression retargeting in video," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 299–310, 2014.

[17] G. Guo, L. Liu, Z. Zhang, Y. Wang, and W. Gao, "An interactive method for curve extraction," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 1905–1908.

[18] T. Xia, B. Liao, and Y. Yu, "Patch-based image vectorization with automatic curvilinear feature alignment," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 115, 2009.

[19] C. Wang, J. Zhu, Y. Guo, and W. Wang, "Video vectorization via tetrahedral remeshing," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1833–1844, 2017.

[20] Z. Shu, X. Wu, H. Fan, P. Huang, D. Wu, C. Hu, and F. Ye, "Parameter-less auto-weighted multiple graph regularized nonnegative matrix factorization for data representation," *Knowledge-Based Systems*, vol. 131, pp. 105–112, 2017.

[21] X. Song, Z.-H. Feng, G. Hu, and X.-J. Wu, "Half-face dictionary integration for representation-based classification," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 142–152, 2017.

[22] Z.-H. Feng, G. Hu, J. Kittler, W. Christmas, and X.-J. Wu, "Cascaded collaborative regression for robust facial landmark detection trained using a mixture of synthetic and real images with dynamic weighting," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3425–3440, 2015.

[23] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu, "Random cascaded-regression copse for robust facial landmark detection," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 76–80, 2015.

[24] 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*, 2014, pp. 2672–2680.

[25] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[26] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, vol. 99, 1999, pp. 200–209.

[27] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," 2002.

[28] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[29] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, 2013, p. 2.

[30] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 3546–3554.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[32] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," *arXiv preprint arXiv:1602.05473*, 2016.

[33] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[34] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 271–279.

[35] C. Li, J. Zhu, T. Shi, and B. Zhang, "Max-margin deep generative models," in *Advances in neural information processing systems*, 2015, pp. 1837–1845.

[36] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[37] C. Li, K. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," *arXiv preprint arXiv:1703.02291*, 2017.

[38] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[39] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Advances in Neural Information Processing Systems*, 2017, pp. 6510–6520.

[40] E. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *arXiv preprint arXiv:1611.06430*, 2016.

[41] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727.

[42] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," *arXiv preprint arXiv:1505.03906*, 2015.

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[44] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2007, pp. 513–520.

[45] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[48] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[49] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2, 2011, p. 5.

[50] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.

Cong Hu received the B.Sc. degree from Jiangnan University, Wuxi, China, in 2009. He received the M.S. degree from Jilin University, Changchun, China in 2012. He is now a PhD student at the School of IoT Engineering, Jiangnan University. His research interests include image processing, pattern recognition, deep learning.

Xiao-Jun Wu received the B.Sc. degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991. He received the M.S. degree and the Ph.D. degree in pattern recognition and intelligent systems from Nanjing University of Science and Technology, Nanjing, China, in 1996 and 2002, respectively.

He is currently a Professor in artificial intelligent and pattern recognition at Jiangnan University, Wuxi, China. His current research interests include pattern recognition, computer vision, fuzzy systems, neural networks and intelligent systems.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2018.2875462, IEEE Transactions on Cognitive and Developmental Systems

Josef Kittler (M'74-LM'12) received the B.A., Ph.D., and D.Sc. degrees from the University of Cambridge, in 1971, 1974, and 1991, respectively. He is a distinguished Professor of Machine Intelligence at the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K. He conducts research in biometrics, video and image database retrieval, medical image analysis, and cognitive vision. He published the textbook Pattern Recognition: A Statistical Approach and over 700 scientific papers. His publications have been cited more than 57,800 times (Google Scholar).

He is series editor of Springer Lecture Notes on Computer Science. He currently serves on the Editorial Boards of Pattern Recognition Letters, Pattern Recognition and Artificial Intelligence, Pattern Analysis and Applications. He also served as a member of the Editorial Board of IEEE Transactions on Pattern Analysis and Machine Intelligence during 1982-1985. He served on the Governing Board of the International Association for Pattern Recognition (IAPR) as one of the two British representatives during the period 1982-2005, President of the IAPR during 1994-1996.

2379-8920 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.