SPECIAL SECTION ON ADVANCES IN MACHINE LEARNING
AND COGNITIVE COMPUTING FOR INDUSTRY APPLICATIONS

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2020, accepted March 30, 2020, date of publication April 16, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988284

Image Denoising With Generative Adversarial Networks and Its Application to Cell Image Enhancement

SONGKUI CHEN[1], DAMING SHI[1], (Senior Member, IEEE), MUHAMMAD SADIQ[1], AND XIAOCHUN CHENG[2], (Senior Member, IEEE)

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2]Department of Computer Science, Middlesex University, London NW4 4BT, U.K.

Corresponding author: Daming Shi (dshi@szu.edu.cn)

This work was supported in part by the Ministry of Science and Technology China (MOST) Major Program on New Generation of Artificial Intelligence 2030 under Grant 2018AAA0102200, in part by the Natural Science Foundation China (NSFC) Major Project under Grant 61827814, and in part by the Shenzhen Innovation Council of Science and Technology Projects under Grant JCYJ20170302153752613 and Grant JCY20190808153619413.

ABSTRACT This paper proposes an image denoising training framework based on Wasserstein Generative Adversarial Networks (WGAN) and applies it to cell image denoising. Cell image denoising is a challenging task which has high requirement on the recovery of feature details. Current popular convolutional neural network (CNN) based denoising methods encounter a blurriness issue that denoised images are blurry on texture details, which is fatal for the cell image denoising. In this paper, to solve the blurriness issue, we first theoretically analyze the cause of the blurriness issue. Subsequently, an image denoising training framework with WGAN based adversarial learning is proposed. This training framework solves the blurriness issue by guiding the denoising network to find the distribution space of real clean images rather than the distribution space of blurry images and introducing feature information. Experimental results show that this training framework can effectively solve the blurriness issue and achieve better denoising performance than the state-of-the-art denoising methods. Meanwhile, the application of this training framework on cell image denoising also achieves satisfactory performance. Recovered cell images of this training framework are clear on feature details.

INDEX TERMS Image denoising, cell image denoising, blurriness issue, adversarial learning, Wasserstein generative adversarial networks.

I. INTRODUCTION

Image denoising is a classic topic in low-level vision as well as an essential preprocessing step in many high-level vision tasks. In general, a given noisy image can be modeled as $y = x + v$, where x is the noise-free image, and v represents the noise. The task of image denoising is to recover the noise-free image x by removing the noise v from the given noisy image y. The existing denoising methods can be divided into two categories: image prior based methods [1]–[12], which obtain denoised image by processing the noisy image according to some prior knowledge about image; discriminative learning methods [13]–[24], which train a model to learn the mapping relationship from given noisy image to the denoised image using a large number of pairs of images (noisy and clean). Nowadays, the discriminative learning methods are the most popular one, because they can automatically exploit and utilize more statistical characteristic of image through training and thus can achieve better denoising performance. Especially, in the discriminative learning methods, the CNN based methods are the most popular one now, because the CNN has features such as sparse connection and weight sharing, these features make the CNN based models easy to train and able to avoid the overfitting problem.

In literature, CNN based methods can also be divided into four categories. The first one is to model the noise v as one model and use the trained model to process the given noisy images y. Usually, the noise is modeled as an additive white Gaussian noise. For example, DnCNN [13] and FFDNet [25] model the noise as additive white Gaussian noise and directly

The associate editor coordinating the review of this manuscript and approving it for publication was Min Xia.

denoise the noisy images using their models trained for removing Gaussian noise. CBDNet [26] models the noise on the raw data of imaging sensors in-camera as heteroscedastic Gaussian noise, and trains their denoising network according to this explicitly defined model. Zhou *et al.* [27] tries to find the relationship between Gaussian noise and real-world noise, and to find a way to apply the Gaussian-noise-trained network to remove the real-world noise. This kind of method can quickly generate many pairs of images for training, but the trained model can not well remove the sophisticated real-world noise.

The second one is the prior knowledge-based methods. Such methods train a denoising network according to some statistical laws. For instance, NOISE2NOISE [28] trains its denoising network using many different pairs of independently degraded images of the same scenes. This method is based on the statistical law, that L2 loss will guide a network to find the mean solution of all potential solutions (the mean solution will have weaker noise intensity). Furthermore, NOISE2VOID [29] offers a more simple solution in which a denoising network is trained only using many single noisy images of different scenes. According to the local similarity of the image, it takes the mean of the surrounding pixels of the objective noisy pixel as the corresponding clean pixel. This kind of method can overcome the need that many pairs of images are needed to train a denoising network. Still, their denoising performance is limited by the used prior knowledge.

The third category is the generative methods. This kind of method removes noise through two phases: noise modeling and supervised denoising. In the noise modeling phase, they are to model the real-world noise using the real-world noisy images and then synthesize many pairs of clean-noisy image pairs for supervised denoising. Like the above methods, the supervised denoising is to use the generated image pairs to train a denoising network to learn the mapping relationship. For example, GCBD [30] models the real-world noise using Generative Adversarial Networks [31], and synthesizes many pairs of clean-noisy images pairs by adding their generated noises to one clean image dataset. Comparing with the first kind of method, this kind of method can more accurately model the real-world noise, but the training process of this kind of method is also more complicated and so that it is not widely used.

The last category is to make a training dataset containing many pairs of clean-noisy images representative of real-world noise. This kind of method also can be divided into two sub-categories according to the way to get the clean images. The first one gets the clean images by taking pictures on low ISO value and taking careful post-processing on the images, while on the contrary, gets the noisy images by taking pictures on high ISO value. The representative dataset is RENOIR [32]. The other one gets the clean images differently. They firstly take multiple photos of a static scene and then get the clean image by computing the mean of the multi images and taking post-processing on the obtained mean image. Meanwhile, they take one of the multi images as the noisy image. The representative datasets are Nam [33], PolyU [34] and SIDD [35]. These made training datasets can be easily used to train a denoising network to remove the sophisticated real-world noise.

To make the denoising network to quickly and efficiently learn the mapping relationship from noisy image to the denoised image, the above four kinds of methods almost use the pixel-wise Mean Squared Error (MSE) as their loss function. However, the denoised images of these MSE based networks always are blurry on texture details, which is fatal for some noisy images with important feature details. In this paper, we will analyze the cause of this blurriness problem and propose a solution to solve this problem.

The major contributions of this paper can be summarized as follows:

- We theoretically analyze the cause of the blurriness problem;
- We propose an image denoising training framework with WGAN based adversarial learning;
- Experimental results prove that the framework can effectively solve the blurriness problem and achieve better denoising performance than the state-of-the-art denoising methods;
- The framework is successfully applied to the cell image denoising. The recovered cell images of this framework are clear on feature details.

The remainder of the paper is organized as follows. Section II and Section III introduce the problem this paper plan to solve and the relative works, respectively. Section IV presents our proposed method. In Section V, extensive conducted experiments are reported to validate the effectiveness of our method. In Section VI, our method is applied to cell image denoising. Section VII gives several concluding remarks.

II. PROBLEM STATEMENT

Taking the MSE between the output image and the corresponding clean image as the loss function of training, can quickly and efficiently guide a denoising network to find the mapping relationship from noisy image to the denoised image. But, because image denoising is a task with multiple solutions and the MSE will guide the denoising network to find the mean solution of the potential multiple solutions, the denoised images of the MSE-based denoising network always are blurry on texture details. Next, making a pixel as an example, more details about this problem will be introduced.

Because of the randomness of noise intensity, high possibility is that after being populated by noise, multiple different clean pixel intensities correspond to the same noisy pixel intensity. This means that during training, a noisy input pixel may correspond to many potential clean pixels. Here, we represent the noisy input pixel as y_0 and represent the corresponding clean pixels as $x_0^1, x_0^2, x_0^3, \ldots, x_0^n$. Besides, the denoising network with trainable parameters Θ is represented as $\mathcal{F}$, and the output denoised pixel is indicated as $\mathcal{F}(y_0; \Theta)$. The MSE between the denoised pixel and the corresponding multiple clean pixels and the expectation of the MSE are as follows

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{F}(y_0; \Theta) - x_0^i)^2 \tag{1}$$

$$\begin{aligned} E(MSE) &= \frac{1}{n}\sum_{i=1}^{n}[E(\mathcal{F}^2(y_0; \Theta)) \\ &\quad - 2 * x_0^i * E(\mathcal{F}(y_0; \Theta)) + (x_0^i)^2] \\ &= \frac{1}{n}\sum_{i=1}^{n}[Var(\mathcal{F}(y_0; \Theta)) + E^2(\mathcal{F}(y_0; \Theta)) \\ &\quad - 2 * x_0^i * E(\mathcal{F}(y_0; \Theta)) + (x_0^i)^2] \\ &= Var(\mathcal{F}(y_0; \Theta)) + \frac{1}{n}\sum_{i=1}^{n}(E(\mathcal{F}(y_0; \Theta)) - x_0^i)^2 \end{aligned} \tag{2}$$

Obviously, minimizing the $E(MSE)$ by upgrading the parameters Θ, the optimal $\mathcal{F}(y_0; \Theta)$ will be obtained at

$$\mathcal{F}(y_0; \Theta) = \frac{1}{n}\sum_{i=1}^{n} x_0^i \tag{3}$$

which means, that the MSE will statistically drive the denoising network to search for the mean solution of the all corresponding potential solutions. As a result, the pixel intensities in the final solution space tend to change slowly, which leads to the recovered images in this solution space are showed blurry on texture details. So, this is the reason why the denoised images of the MSE-based methods always are blurry on texture details.

III. RELATIVE WORK

A. SOLVING BLURRINESS PROBLEM

In the field of image denoising, there are little works to solve the blurriness problem. Most of the existed methods [15], [36], [37] solve the blurriness problem by introducing feature information from another trained high-level network, to guide the training of the denoising network. For example, [15] concatenates a trained network for image classification behind its denoising network during training. It feeds the denoised image and the corresponding real clean image to the trained network, and use the back propagated loss information to train its denoising network. Because the trained network has high capacity of extracting features, the loss information carries the feature information that indicates the distance between the input denoised image and real clean image on features. By making use of the feature information, the denoised network can perform better on recovering texture details.

However, the performance of these methods is still unsatisfactory since the established feature space (i.e., the trained network) can not provide sufficient feature information. Because the high-level network is trained for a particular task, the trained network is only sensitive to some features, relative to its solved task. For the features not relative to its solved task, the trained network can not extract the features from the input images, and so that it can not measure the distance between the denoised image and the real clean image on the features. As a result, the trained network can not provide useful feature information to help the denoising network to recover the features better.

In this paper, after analyzing the cause of the blurriness problem, we propose a method to radically solve the blurriness problem by guiding denoising network to find the distribution space of real clean images rather than the distribution space of blurry images. The proposed method is an image denoising training framework with adversarial learning based on WGAN [38]. The WGAN has been proved that it has high capacity to fit the distribution of generated data to the distribution of real clean data. The WGAN is a variant of Generative Adversarial Nets (GAN) [31], and it needs to be explained that the reason why we employ WGAN instead of GAN in this paper is that it is difficult to train GAN (more details will be introduced in IV-A) and the WGAN can well solve this training difficulty problem. Next, before introducing the WGAN, the GAN will be briefly introduced firstly in next subsection.

B. GENERATIVE ADVERSARIAL NETS

GAN [31] is a training framework that generally consists of a generative subnetwork and a discriminative subnetwork. The former takes random low-dimension noisy data as its input, and targets at generating high-dimension images which should follow the distribution of given real images so that they can cheat the discriminative subnetwork. Whereas, the latter targets at accurately distinguishing the generated image from the real image. Their role in training can be looked as a minimax two-player game.

$$\begin{aligned} \min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_r}[logD(x)] \\ + \mathbb{E}_{y \sim p_n}[log(1 - D(G(y)))] \end{aligned} \tag{4}$$

where p_r and p_n represent the distribution of real images and the distribution of input noisy data, respectively. For discriminative subnetwork, it is to maximize the above loss function and in contrary the generative subnetwork is to minimize the loss function. The two subnetworks will be alternately trained. After finishing training, the generative subnetwork can successfully learn the distribution p_r and generate ideal outputs following this distribution. This is the reason why we introduce adversarial learning to guide the training of our denoising network. The adversarial learning can help our denoising network to find the distribution space of real clean images rather than the distribution space of blurry images. However, training a GAN is tricky and unstable [39]. Fortunately, the WGAN can well solve this training problem, and thus we apply it to our image denoising work. In the next section, we will briefly introduce the WGAN.

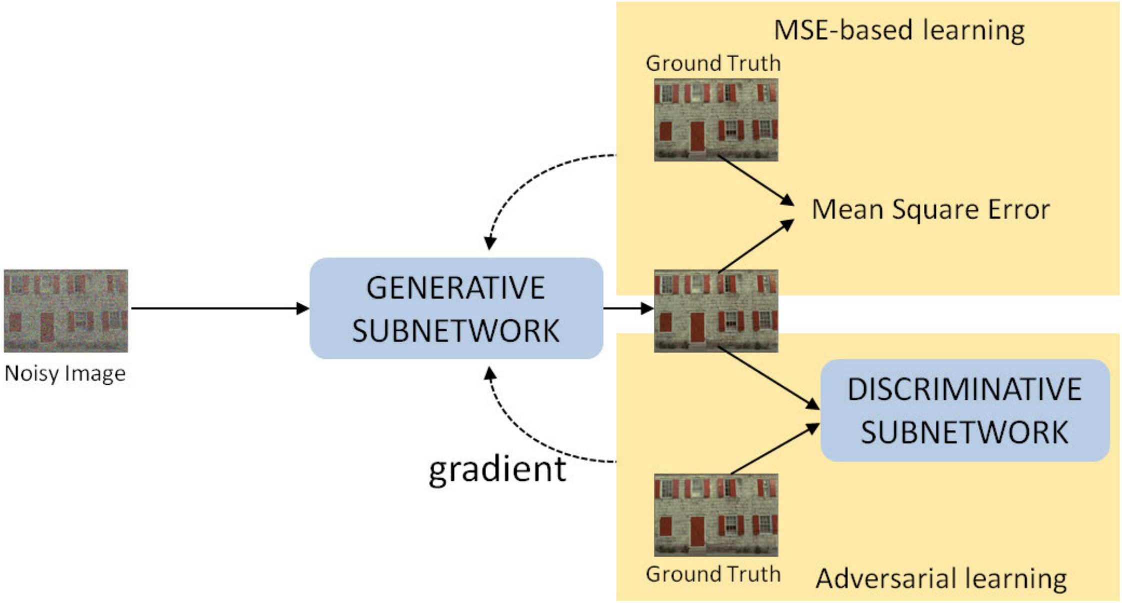


FIGURE 1. Overall structure of our proposed image denoising training framework with adversarial learning based on WGAN.

IV. PROPOSED IMAGE DENOISING TRAINING FRAMEWORK

To solve the blurriness problem with image denoising, we propose an image denoising training framework with adversarial learning based on WGAN. Also, we fine-tune the training details of WGAN to make it suitable for our denoising task. The overall structure of our framework is shown in Fig. 1. The framework can be divided into three parts: generative subnetwork, MSE-based learning, and adversarial learning. The generative subnetwork is to learn the mapping relationship from a noisy image to the denoised image. MSE-based learning is to guide the generative subnetwork to quickly and efficiently learn the mapping relationship. The adversarial learning is to solve the blurriness problem existed in MSE-based learning by guide the generative subnetwork to find the distribution space of real clean images, not the distribution space of blurry images. Next, we will firstly introduce the WGAN and the relative adversarial learning. Subsequently, the details about the generative subnetwork will be described. Finally, we will enumerate the training details of this image denoising training framework.

A. WASSERSTEIN GENERATIVE ADVERSARIAL NETWORKS

In GAN, the discriminative subnetwork and the generative subnetwork will be alternatively trained, and the task to train the generative subnetwork is equivalent to minimizing the Jensen-Shannon (JS) divergence between the distribution p_r of real data and the distribution p_g of generated data (from the generative subnetwork) [31]. Theoretically, by minimizing the JS divergence, the distribution p_g can be transformed to the distribution p_r. Actually, the training easily falls into gradient vanishing trap and thus the generative subnetwork can not be continuously updated. The reason is that there is no nonnegligible intersection between the two distributions at the beginning of the training, and so that the JS divergence is a constant, as a result, there is no gradient information can be backpropagated to upgrade the generative subnetwork [39]. Therefore, it is necessary but difficult to carefully balance the trainings of the two subnetworks to avoid the gradient vanish.

WGAN [38] uses Wasserstein distance rather than JS divergence to measure the distance between the two distributions. The Wasserstein distance has better property than the JS divergence. Even if there is no nonnegligible intersection between the two distributions, Wasserstein distance can still well measure the distance between the two distributions. Therefore, using Wasserstein distance as metric can well avoid the gradient vanishing problem. In WGAN, it is unnecessary to carefully balance the trainings of the two subnetworks, and the training becomes easy.

In WGAN [38], the Wasserstein distance is approximated as $\mathbb{E}_{x\sim p_r}D(x)+\mathbb{E}_{y\sim p_n}[1-D(G(y))]$ with regular discriminative subnetwork D. To satisfy the regularization, the weight of discriminative subnetwork D is clip to a certain range. Subsequently, in [40], a gradient penalty term is proposed to replace the weight clip, because the weight clip would lead to gradient vanishing or exploding. The final loss function can be formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_r}D(x) - \mathbb{E}_{y\sim p_n}D(G(y)) \\ - \lambda(\| \nabla_z D(z) \|_2 - 1)^2 \quad (5)$$

where the former two terms indicate the Wasserstein distance estimation and the third term is a gradient penalty term to regularize the discriminative subnetwork D. The z in the gradient penalty term is sampled uniformly along straight lines between the input real data x and generated data $G(y)$. For the discriminative subnetwork, it is to maximize the above loss function for accurately estimating the Wasserstein distance between the distribution p_g and the distribution p_r. In contrary, the generative subnetwork is to minimize the above loss function for minimizing the Wasserstein distance so as to make the distribution p_g fit the distribution p_r. Through training, the p_g will eventually fit the distribution p_r.

B. ADVERSARIAL LEARNING

To solve the blurriness problem, we introduce adversarial learning based on WGAN. In this adversarial learning, the discriminative subnetwork is to estimate the Wasserstein distance between the distribution of denoised images and the distribution of real clean images. And for the generative subnetwork, by upgrading itself to minimize the Wasserstein distance, it can make the distribution of its output denoised images to fit the distribution of real clean images.

1) FREEZING INPUT

In the original WGAN, in each alternative adversarial training iteration, the inputs to the generative subnetwork are different when respectively training the generative subnetwork and the discriminative subnetwork. Differently, in our framework, the inputs will remain same, which can bring feature information when training the generative subnetwork.

In each alternative adversarial training iteration, the discriminative subnetwork will be optimized firstly. As we all known, CNN has strong ability on extracting features. After being optimized, the discriminative subnetwork can well discriminate the input denoised image from the real clean image on features. When in turn optimizing the generative subnetwork, if we still take the same noisy image as the input of generative subnetwork, the out denoised image will also be keep same. Because the discriminative subnetwork has been trained to well discriminate the denoised image from the real clean image on features, the loss information from the discriminative subnetwork can guide the generative subnetwork to perform better on recovering the features. In contrary, if taking a new input to the generative subnetwork, the output denoised image is also a new input for the discriminative subnetwork. Because the discriminative subnetwork has not been trained to distinguish the new denoised image from the real clean image, it actually can not discriminate them on features. As a result, the loss function from the discriminative subnetwork would not be able to perform as it does in the above case.

2) ARCHITECTURE OF DISCRIMINATIVE SUBNETWORK

Our discriminative subnetwork is designed based on the discriminator in [37]. We make some changes to the original one. To reduce the complexity of the network, we remove some convolution layers. Besides, we use a max-pooling layer to replace the original two fully connected layers to reduce the dimension of input feature maps, which can significantly reduce the parameters of the network. It is worth noting that our experimental results show that the performance of using the max-pooling layer is better than the performance of using a average-pooling layer in this subnetwork. This is mainly due to the max-pooling layer that can efficiently extract the most important features to distinguish the denoised image from the real clean image correctly. According to GP-WGAN [40], the batch normalization layer would make the gradient penalty loss term invalid, so we remove all of the batch normalization layers. Besides, as the request of WGAN [38], we remove the sigmoid layer. The final architecture of our discriminative subnetwork is shown in Fig. 2. The functionality of this subnetwork can be divided into three parts. Firstly, the subnetwork extracts and learns the features of the input image through seven convolution layers with seven LeakyReLU activation functions. Then, one convolution layer with one 1 × 1 kernel is to fuse the learned features. Finally, one max-pooling layer selects the most important feature and output one value.

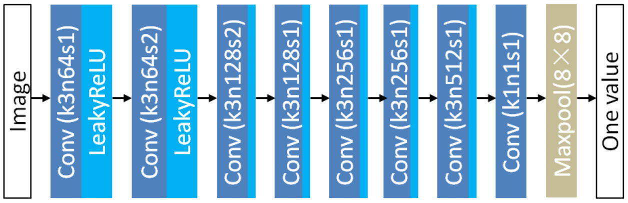


FIGURE 2. Architecture of our designed discriminative subnetwork. Each convolution layer is indicated with corresponding kernel size (k), number of feature maps (n), and stride (s).

The adversarial loss function to optimize this discriminative subnetwork can be formulated as

$$L_{Dis}(\Phi) = \frac{1}{N}\sum_{i=1}^{N}[D(x_i;\Phi) - D(\mathcal{F}(y_i;\Theta);\Phi) - \lambda(\parallel \nabla_z D(z) \parallel_2 - 1)^2] \tag{6}$$

where $\mathcal{F}(y_i;\Theta)$ represents the denoised image from generative subnetwork; the function $D(\cdot)$ represents the discriminative subnetwork with trainable parameters Φ; the first two terms indicate the Wasserstein distance estimation between the distribution of denoised images $\mathcal{F}(y_i;\Theta)$ and the distribution of real clean images x_i; the last term is a gradient penalty term for the regularization of discriminative subnetwork; the z in the gradient penalty term is sampled uniformly along straight lines between the input real clean image x_i and denoised image $\mathcal{F}(y_i;\Theta)$; the λ is the weight value of the gradient penalty term.

3) ADVERSARIAL LEARNING TO SOLVE BLURRINESS

The reason why our adversarial learning can solve the blurriness problem is in two-folds. Firstly, by making use of WGAN's great capacity on distribution fitting, this adversarial learning can help denoising network (i.e. the generative subnetwork) to find the distribution space of real clean images rather than the distribution space of blurry images. As we discussed in II, the MSE loss function will guide denoising network to find a solution space in which the pixel

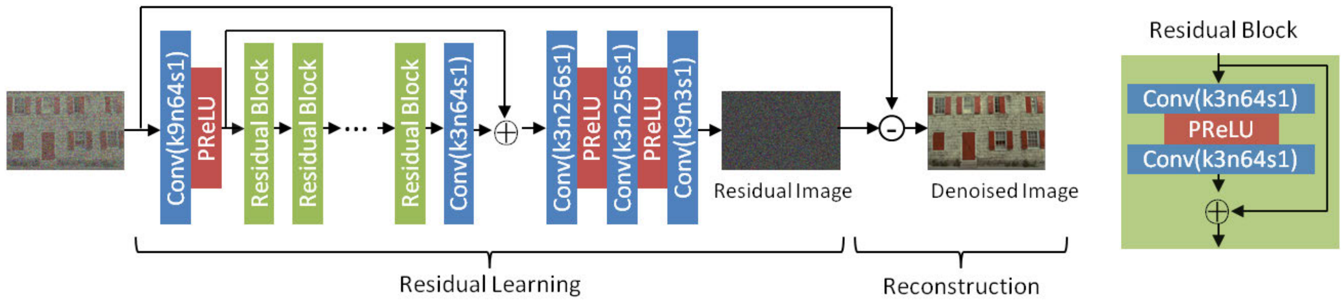


FIGURE 3. Architecture of our designed generative subnetwork. Each convolution layer is indicated with corresponding kernel size (k), number of feature maps (n), and stride (s). All of the convolution layers are set zero-padding to keep the dimensions of the output and the input as same. There are totally 16 residual blocks in this subnetwork.

intensities change slowly. And this is the root reason why the denoised images of MSE-based denoising network are blurry on texture details. In this adversarial learning, the generative subnetwork will be driven to learn the distribution of real clean images so that the distribution of its output denoised images fit the distribution of the real clean images. In other words, this adversarial learning can correct the over-smoothness solution space to the real clean images solution space.

Secondly, as the previous works [15], [36], [37] that are introduced in III, this adversarial learning also can provide feature information for the training of generative subnetwork. The methods [15], [36], [37] connect a trained network behind the denoising network during training, using the feature information from the trained network to guide their denoising network to better recovere the features. In this adversarial learning, it also connects a network (i.e. the discriminative subnetwork) behind the generative subnetwork. During this adversarial training, the discriminative subnetwork will be trained firstly, which means that for the generative subnetwork, the discriminative subnetwork is also a trained network. Therefore, like the previous works, the discriminative subnetwork can also offer useful feature information to guide the training of the generative subnetwork. Besides, the discriminative subnetwork is not trained for one specific task, so it can be sensitive to more features than the previous works.

C. GENERATIVE SUBNETWORK

We design our generative subnetwork as an end-to-end network, which takes a noisy image as an input and produces the corresponding denoised image as output. Basically, it is inspired by super-resolution deep learning [37], [41] and residual learning [13]. Our generative subnetwork firstly learns the residual image and then obtains the desired denoised image by removing the residual information from the noisy input image. This residual learning can avoid the optimization difficulty to learn identity mapping [42] because the denoising process is approximately close to an identity mapping, especially when the noise level is low. The architecture of our generative subnetwork is shown in Fig. 3. This architecture can be divided into two parts: residual learning and reconstruction. In the residual learning phase, one convolution layer with a PReLU function extracts the initial features of noisy input image. Then, a big residual block containing 16 small residual blocks (shown in green on Fig. 3) and a convolution layer removes the latent clean features. Later, three convolution layers with two activation functions reconstruct the remained residual features to get the final residual image. In the reconstruction phase, the denoised image is obtained by removing the residual image from the noisy input. The loss function to optimize this generative subnetwork can be formulated as follow:

$$L_{Den}(\Theta) = l_{MSE}(\Theta) + \alpha l_{Adv}(\Theta) \tag{7}$$

where l_{MSE} represents the MSE loss from the MSE-based learning and l_{Adv} represents the adversarial loss from the adversarial learning. α is the weight value to control the trade-off between the two loss terms. The MSE loss l_{MSE} can be formulated as:

$$l_{MSE}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \parallel \mathcal{F}(y_i;\Theta) - x_i \parallel^2 \tag{8}$$

where $\{(y_i, x_i)\}_{i=1}^{N}$ represents N noisy-clean training patch pairs and $\mathcal{F}(y_i;\Theta)$ represents the output denoised image from the generative subnetwork that takes the noisy image y_i as input.

The adversarial loss l_{Adv} can be formulated as

$$_{Adv}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} -D(\mathcal{F}(y_i;\Theta);\Phi) \tag{9}$$

D. TRAINING PROCEDURE

In this image denoising training framework, the discriminative subnetwork and the generative subnetwork will be alternately trained. We explain the training procedure in **ALGORITHM 1**. In the algorithm, the (X, Y) is the training dataset with many couples of clean images X and noisy images Y. t is the iteration number and T is the total number of iterations. Function $\mathcal{F}(\cdot)$ is the generative subnetwork and function $D(\cdot)$ is the discriminative subnetwork.

Algorithm 1 Training Procedure of Our Image Denoising Training Framework

```
procedure ADVERSARIAL TRAINING ($X, Y$)
    Initial generative subnetwork parameters $\Theta$ and
    discriminative subnetwork parameters $\Phi$. t = 0.
    while $t < T$ do
        $x, y = minibatch(X, Y)$
        $\hat{x} = \mathcal{F}(y; \Theta)$
        $O_{real} = D(x; \Phi)$
        $O_{denoised} = D(\hat{x}; \Phi)$
        $\{\epsilon^i\}_{i=1}^{B} \sim U[0, 1]$
        $z = x + \epsilon * (\hat{x} - x)$
        $O_{penalty} = D(z; \Phi)$
        Maximizing Eqn. 6 to update adversarial
        subnetwork.
        $O_{fake} = D(\hat{x}; \Phi)$
        Minimizing Eqn. 7 to update generative
        subnetwork.
        t = t + 1
```

O_{real} and $O_{denoised}$ indicate the output value of discriminative subnetwork that takes the real clean image and the denoised image as input respectively. $O_{penalty}$ indicates the output value of discriminative subnetwork in the gradient penalty term. B represents the batch size.

V. EXPERIMENTS

A. EXPERIMENTAL SETTING

1) DATASETS

We will validate the effectiveness of our proposed method through two aspects simultaneously: Gaussian denoisng and real-world noise removal. For the Gaussian denoising, follow [13], we use 400 gray images with size of 180 × 180 for training. By randomly cropping the 400 images, there are totally 128 × 1772 patches (40 × 40) as training dataset. We train our models on three noise levels ($\sigma = 25, 35, 50$), and each model will be trained by 50 epoches using the training dataset. For the real-world noise removal, we use the small SIDD [35] training dataset containing 160 color images to train our model. The model is trained by 640000 times with 64 patches (64 × 64).

We use two different test datasets to test the performance of our model for Gaussian denoising. One is BSD68 [43] that contains 68 gray images from Berkeley segmentation dataset and the other is Set12 that contains 12 gray images. The two datasets are widely used to test the performance of Gaussian denoising methods. We use the test dataset of SIDD to test the performance of our model trained for real-world noise removal. The test dataset of SIDD only contains noisy images, and the PSNR of denoised images is obtained through the online submission system.[1]

2) TRAINING DETAILS

For the model trained for Gaussian denoising, the learning rate of the first 30 epoches are set as 1×10^{-4} and the later epoches are set as 1×10^{-5}. For the model trained for real-world noise removing, the learning rates of the first 4×10^5 training times are set as 1×10^{-4} and the learning rates of the later training times are set as 1×10^{-5}. All models are trained using Adam optimizer in which the hyper-parameters β_1 and β_2 are set as 0 and 0.9 respectively. Specially, the weight value λ in Eqn. 6 is set as 10 and the weight value α in Eqn. 7 is set as 1.01.

B. PERFORMANCE OF PROPOSED IMAGE DENOISING TRAINING FRAMEWORK

In this part, we are to validate the effectiveness of our image denoising training framework. For convenience, we represent the denoising network trained by only using MSE-based learning as ID-MSE, and represent the denoising network trained by our training framework (i.e. the denoising network is trained by using MSE-based learning and adversarial learning (based on WGAN) simultaneously) as ID-MSE-WGAN. Specially, different from the ID-MSE-WGAN, the denoising network trained by using MSE-based learning and adversarial learning without freezing input is represented as ID-MSE-WGAN(WF).

The average PSNR of the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on test dataset Set12 at noise levels $\sigma = 25, 35, 50$ are respectively shown in Fig. 4, Fig. 5 and Fig. 6. It can be seen that at each noise level,

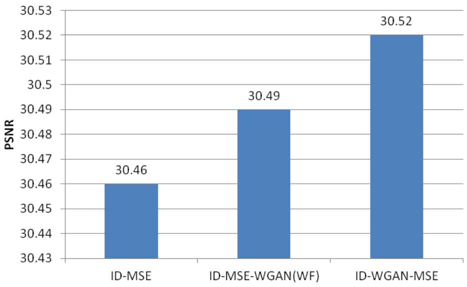


FIGURE 4. The average PSNR of the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on test dataset Set12 at noise level $\sigma = 25$.

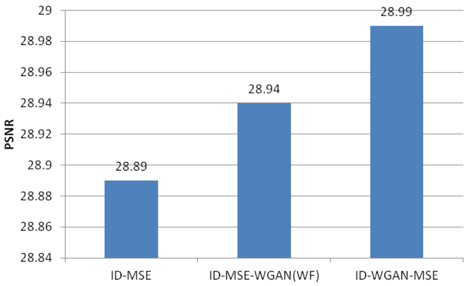


FIGURE 5. The average PSNR of the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on test dataset Set12 at noise level $\sigma = 35$.

[1] https://www.eecs.yorku.ca/ kamel/sidd/benchmark.php

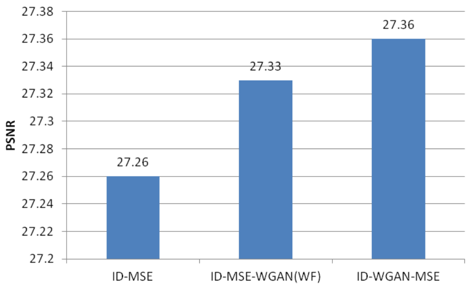


FIGURE 6. The average PSNR of the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on test dataset Set12 at noise level $\sigma = 50$.

the ID-MSE-WGAN(WF) surpasses the ID-MSE. This result proves that the adversarial learning, by guiding denoising network to find the distribution space of real clean images rather than the distribution space of blurry images, can effectively help the denoising network to improve its denoising performance. Comparing the ID-MSE-WGAN with the ID-MSE-WGAN(WF), it can also be easily seen that the ID-MSE-WGAN surpasses the ID-MSE-WGAN(WF) at each noise level. This improvement indicates that the freezing input in adversarial learning can help further improve denoising performance by bring feature information. The average PSNR of the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on test dataset BSD68 at noise levels $\sigma = 25, 35, 50$ are together shown in Table 1, and the best are shown in bold. The relationship among the ID-MSE, ID-MSE-WGAN(WF) and ID-MSE-WGAN on dataset BSD68 is same as the relationship on the dataset Set12. The above results prove that our proposed image denoising training framework can help a denoising network to improve its denoising performance.

TABLE 1. The average PSNR of our ID-MSEA and ID-MSE on test dataset BSD68 at noise level $\sigma = 25, 35, 50$, respectively.

	Gaussian Denoising		
Noise Level	25	35	50
ID-MSE	29.25	27.75	26.28
ID-MSEA(WF)	29.27	27.77	26.31
ID-MSEA	**29.28**	**27.79**	**26.32**

Furthermore, we have additionally trained two denoising networks (ID-MSE-GAN and ID-MSE-P). The ID-MSE-GAN is trained by using MSE-based learning and adversarial learning based on GAN. The ID-MSE-P is trained by using MSE-based learning and perceptual loss (i.e. the loss function from a trained image classification network VGG19) as the previous works [15], [36], [37]. The average PSNR of this two networks on the test dataset Set12 at noise level $\sigma = 25$ are 30.44db and 29.89db, respectively. They are both less than our ID-MSE-WGAN (30.52db). The reason why our ID-MSE-WGAN can perform better than the ID-MSE-GAN is that the GAN based method is difficult to be trained and it easily falls into gradient vanishing trap. And the result that our ID-MSE-WGAN outperforms the ID-MSE-P proves that our solution can perform better on removing noise and recovering features than the previous solutions.

We also show two examples, respectively, from our ID-MSE-WGAN and the ID-MSE, to visualize the effect of our method on recovering texture details. The two examples are shown in Fig. 7. In the first example, it can be easily found that our ID-MSE-WGAN can well recover the regular texture details of starfish from the severely noisy starfish image. And by contrast, the denoised image from the ID-MSE is blurry on these texture details and even loses them. In the second example, faced with the irregular texture details on the butterfly wing, our ID-MSE-WGAN can still recover them better than the ID-MSE. These two examples prove that our image denoising training framework can effectively help denoising network to improve its performance on recovering texture details.

C. COMPARISON WITH STATE-OF-THE-ART METHODS

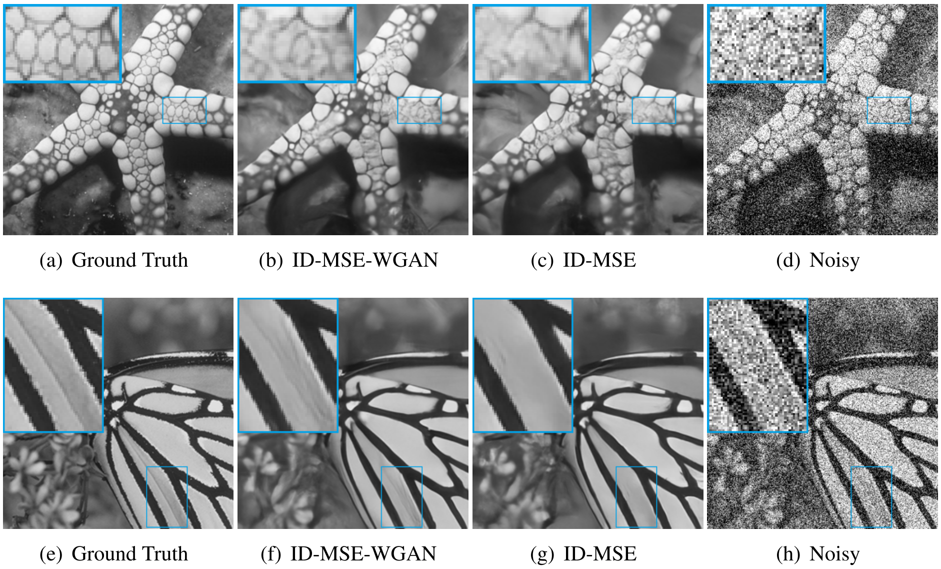
In this part, we are to compare our ID-MSE-WGAN with the state-of-the-art denoising methods to prove the superiority of our method. The compared methods include BM3D [3], EPLL [44], WNNM [4], MLP [45], TNRD [46], DnCNN [13], FFTNet [25] and CBDNet [26].

The results of all methods are shown in Table 2 and the best results are shown in bold. Specially, the CBDNet conducts experiments only on real-world noise removal not on Gaussian denoising. In the table, it can be easily observed that our proposed ID-MSE-WGAN significantly surpass the compared methods at each noise level. On Gaussian denoising, compared with the popular state-of-the-art denoising method DnCNN, our ID-MSE-WGAN has 0.08db, 0.17db, 0.18db gains on Set12 dataset at noise level $\sigma = 25, 35, 50$, respectively. On real world denoising, our ID-MSE-WGAN surpasses the DnCNN 13.48db. Compared with the second best one CBDNet, our ID-MSE-WGAN still exceeds it 3.86db. The above results show that our proposed image denoising method can achieve better performance than the state-of-the-art denoising methods.

VI. APPLICATION TO CELL IMAGE DENOISING

At present, identifying and classifying cell images remains a challenging task. Because the features of cells are very subtle, it is difficult to recognize and extract the features. Worse, when the cell images are noisy, recognizing and extracting the features will be more difficult, because the features are impaired or even obscured by noise. And in fact, the cell images are effortless to be polluted by noise (the noise may come from the imaging process, image transmission process, etc.). Therefore, it is necessary to remove the noise from the noisy cell images before identifying and classifying the cell images.

Because of the importance of features on noisy cell images, it is a requirement for the selected denoising method that recovering well the features when removing the noise. However, the most current denoising methods can not well meet this requirement as we discussed in II. Fortunately, our proposed ID-MSE-WGAN can well meet this requirement, as we discussed in IV and validated in V. And now, we are

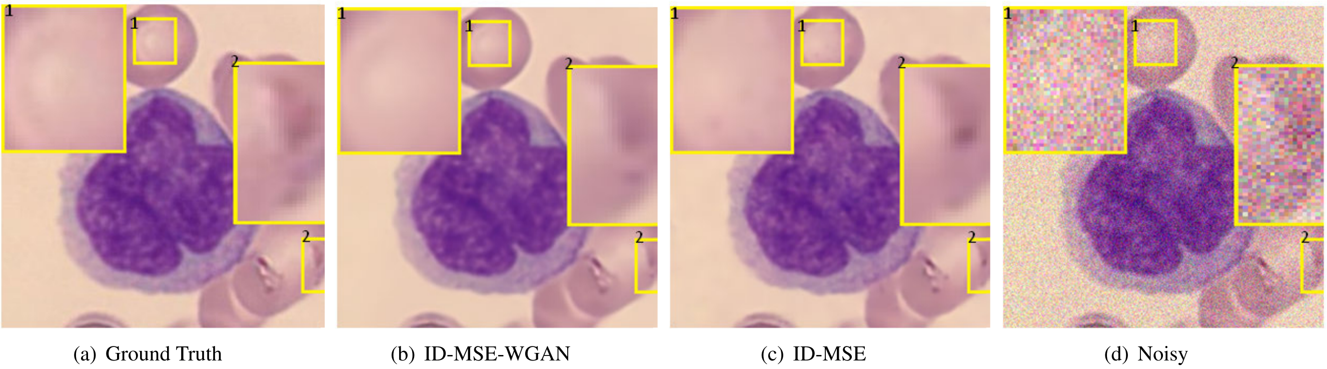

(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

(e) Ground Truth (f) ID-MSE-WGAN (g) ID-MSE (h) Noisy

FIGURE 7. **Two examples of denoised images ($\sigma = 50$) from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are shaper and more realistic on texture details.**

TABLE 2. **The average PSNR of our ID-MSE-WGAN and the compared methods on test datasets Set12, BSD68 and SIDD. The best results are shown in bold.**

	Gaussian Denoising						Real World Denoising
Dataset	Set12			BSD68			SIDD
Noise Level	25	35	50	25	35	50	-
BM3D	29.97	28.40	26.72	28.57	27.08	25.62	25.65
EPLL	29.69	-	26.47	28.68	-	25.67	27.11
WNNM	30.26	28.69	27.05	28.83	27.30	25.87	25.78
MLP	30.03	28.46	26.78	28.96	27.50	26.03	24.71
TNRD	30.06	-	26.81	28.92	-	25.97	24.73
DnCNN	30.44	28.82	27.18	29.23	27.69	26.23	23.66
FFDNet	30.43	28.92	27.32	29.19	27.73	26.29	-
CBDNet	-	-	-	-	-	-	33.28
ID-MSE-WGAN	**30.52**	**28.99**	**27.36**	**29.28**	**27.79**	**26.32**	**37.14**



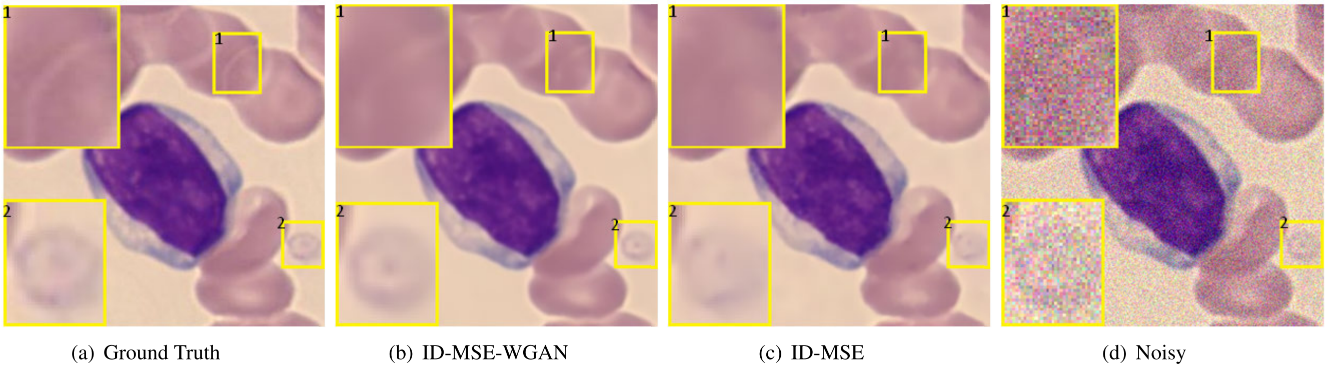
(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

FIGURE 8. **The first example of denoised images from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are more clear and more realistic on texture details.**

to apply our proposed ID-MSE-WGAN to the cell image denoising.

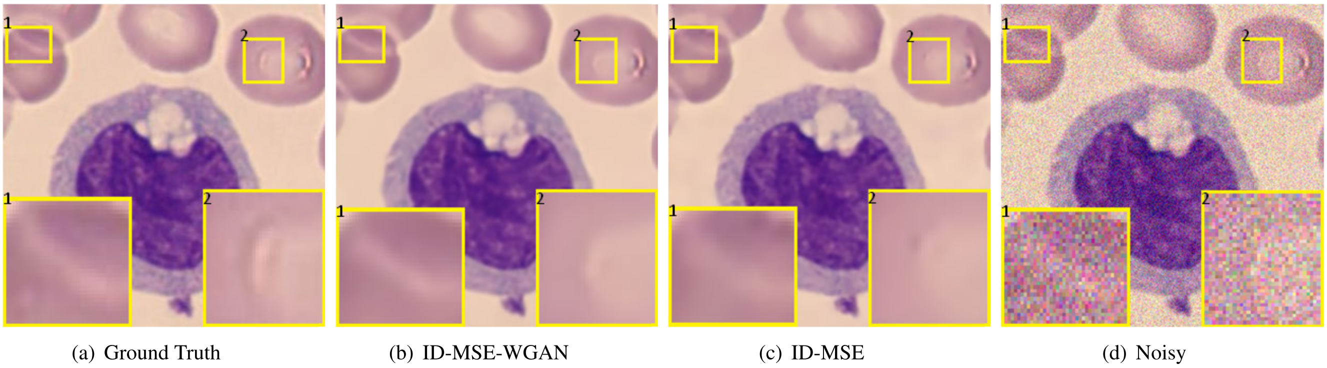
We apply our method to cell dataset collected by [47] from the CellaVision blog.[2] This dataset contains 100 cell images with size 300 × 300. We randomly select 88 images out of 100 and randomly clip 128 × 1303 patches as our training dataset. And the remaining 12 images are used as our test dataset. Therefore, the test dataset is absolutely different from the training dataset. We train three models to remove the light, medium and high level of noise ($\sigma = 25, 35, 50$) respectively.

[2] http://blog.cellavision.com/



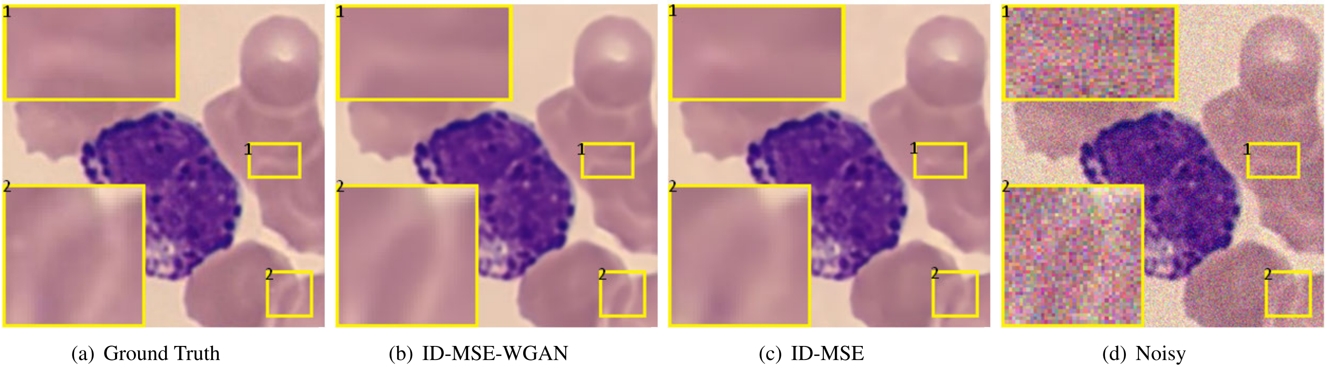
(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

FIGURE 9. **The second example of denoised images from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are more clear and more realistic on texture details.**



(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

FIGURE 10. **The third example of denoised images from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are more clear and more realistic on texture details.**



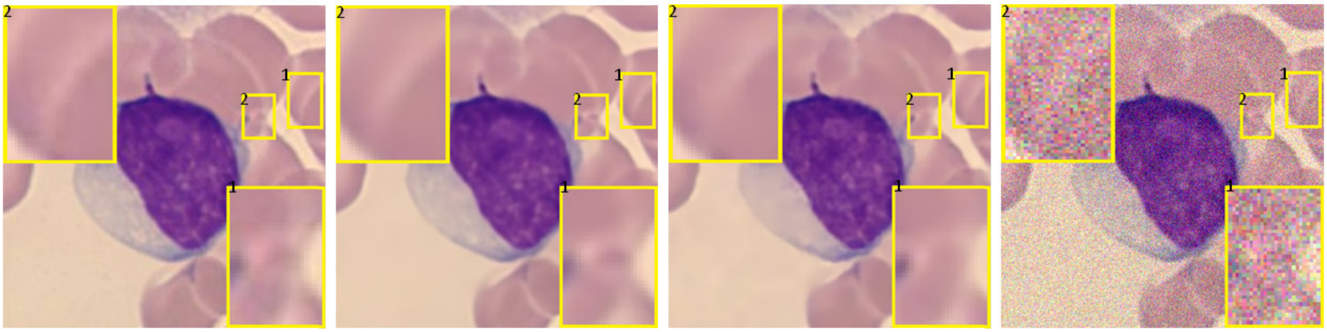
(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

FIGURE 11. **The fourth examples of denoised images from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are clearer and more realistic on texture details.**

All models are trained by 50 epoches. The learning rates of the first 30 epoches are set as 1×10^{-4} and the learning rates of the later epoches are set as 1×10^{-5}. Other setting details are keep same as the details shown in V-A.2.

Five examples on light level of noise ($\sigma = 25$) are illustrated to show the effectiveness of our method on recovering features. In each example, we zoom in two feature details in particular. The first example is shown in Fig. 8. In the first place, there is a halo, which can observed in the ground truth one and the denoised image from our ID-MSE-WGAN. However, in the denoised image from ID-MSE, the halo is barely visible. In the second place, there is a fault structure. In the denoised image from our ID-MSE-WGAN, the fault structure can still be seen. But in the denoised image from ID-MSE, the fault structure has been lost.

The second example is shown in Fig. 9. In the first place, there is a longitudinally curved crack feature. In the denoised image from our ID-MSE-WGAN, the curved crack feature



(a) Ground Truth (b) ID-MSE-WGAN (c) ID-MSE (d) Noisy

FIGURE 12. The fifth example of denoised images from our ID-MSE-WGAN and ID-MSE. Obviously, the denoised images from our ID-MSE-WGAN are more clear and more realistic on texture details.

still can be observed. But, in the denoised image from ID-MSE, the feature is lost. In the second place, there is a vesicle, which still can be clearly seen in the denoised image from our ID-MSE-WGAN. However, in the denoised image from ID-MSE, some features of the vesicle are blurry so that they can not be clearly seen.

The third example is shown in Fig. 10. In the first place, there is a transverse curved crack feature. In the denoised image from our ID-MSE-WGAN, the feature still can be clearly observed. But in the denoised image from ID-MSE, the feature is so blurry that it can not be clearly observed. In the second place, there is a moon-shaped feature, and the feature still can be extracted in the denoised image from our ID-MSE-WGAN. In the denoised image from ID-MSE, unfortunately, the feature can not be seen.

The fourth example is shown in Fig. 11. In the first place, there is a lateral fault structure. In the denoised image from our ID-MSE-WGAN, the feature is still clear. But in the denoised image from ID-MSE, the edge of the lateral fault is blurry and the feature becomes unclear. In the second place, there is a block. The shape of the block can be clearly and easily seen in the denoised image from our ID-MSE-WGAN. However, in the denoised image from ID-MSE, the edge of the block is blurry and thus its shape can not be easily distinguished.

The fifth example is shown in Fig. 12. In the first place, there is a small red block between the cracks. In the denoised image from our ID-MSE-WGAN, the small red block still can be clearly distinguished. But in the denoised image from ID-MSE, the edge of the block is so blurry that the block becomes a piece of the surrounding area, and it can not be distinguished. In the second place, there is also a longitudinally curved crack, and our ID-MSE-WGAN can well recover it but ID-MSE can not. From the above examples, it can be easily concluded that our ID-MSE-WGAN can well satisfy the requirement that removes the noise while recovering well the features in the cell images.

VII. CONCLUSIONS

In this paper, an image denoising training framework with WGAN-based adversarial learning is proposed. The main goal of WGAN-based adversarial learning is to address the blurriness problem caused by the pixel-wise loss function. We employed WGAN instead of primary GAN because it solves the training difficulty problem of GAN and improves GAN's performance. Moreover, to customize the WGAN to the image denoising application, the input to the generative subnetwork in each alternative adversarial training iteration is frozen. We validated our method based on three datasets, including Set12, BSD68, and SIDD, with Gaussian denoising and real-world denoising. The experimental results demonstrated that our proposed framework outperforms the state-of-the-art denoising methods and generates clearer and more realistic denoised images in terms of texture details. Keeping in view the importance of feature details restoration in cell image denoising, we applied our framework to cell image denoising. And results have proved that our method can remove noise and recover feature details very well simultaneously.

ACKNOWLEDGMENT

This research is carried out in National Engineering Laboratory for Big Data System Computing Technology, China.

REFERENCES

[1] J. Wang, F. Tian, H. Yu, C. H. Liu, K. Zhan, and X. Wang, "Diverse non-negative matrix factorization for multiview data representation," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2620–2632, Sep. 2018.

[2] O. Tarkhaneh and H. Shen, "An adaptive differential evolution algorithm to optimal multi-level thresholding for MRI brain image segmentation," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112820.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Proc. Image Process., Algorithms Syst., Neural Netw., Mach. Learn.*, Feb. 2006, Art. no. 606414.

[4] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[5] A. Buades, B. Coll, and J.-M. Morel, "A Non-Local Algorithm for Image Denoising," in *2005 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, 2005, pp. 60–65.

[6] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[7] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2272–2279.

[8] S. Chen, D. Shi, M. Sadiq, and M. Zhu, "Image denoising via generative adversarial networks with detail loss," in *Proc. 2nd Int. Conf. Inf. Sci. Syst. (ICISS)*, 2019, pp. 261–265.

[9] C. Knaus and M. Zwicker, "Progressive image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3114–3125, Jul. 2014.

[10] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 244–252.

[11] F. Chen, L. Zhang, and H. Yu, "External patch prior guided internal clustering for image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 603–611.

[12] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, Feb. 2013.

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[14] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2808–2817.

[15] D. Liu, B. Wen, X. Liu, Z. Wang, and T. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 842–848.

[16] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[17] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-Local Recurrent Network for Image Restoration," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1680–1689.

[18] M. Sadiq, D. Shi, M. Guo, and X. Cheng, "Facial landmark detection via attention-adaptive deep network," *IEEE Access*, vol. 7, pp. 181041–181050, 2019.

[19] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2774–2781.

[20] J. Batson and L. Royer, "Noise self: Blind Denoising by self-supervision," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 524–533.

[21] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11028–11037.

[22] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3155–3164.

[23] X. Jia, S. Liu, X. Feng, and L. Zhang, "FOCNet: A fractional optimal control network for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6047–6056.

[24] V. Sterzentsenko, L. Saroglou, A. Chatzitofis, S. Thermos, N. Zioulis, A. Doumanoglou, D. Zarpalas, and P. Daras, "Self-supervised deep depth denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1242–1251.

[25] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[26] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.

[27] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, and T. Huang, "When AWGN-based denoiser meets real noises," 2019, *arXiv:1904.03485*. [Online]. Available: http://arxiv.org/abs/1904.03485

[28] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning Image Restoration without Clean Data," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 10–15 Jul. 2018, pp. 2965–2974.

[29] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2 Void–Learning denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2124–2132.

[30] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2014, pp. 2672–2680.

[32] J. Anaya and A. Barbu, "RENOIR—A dataset for real low-light image noise reduction," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 144–154, Feb. 2018.

[33] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1683–1691.

[34] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark," 2018, *arXiv:1804.02603*. [Online]. Available: http://arxiv.org/abs/1804.02603

[35] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.

[36] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 694–711.

[37] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

[38] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 214–223.

[39] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–8.

[40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5767–5777.

[41] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[43] S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, Apr. 2009.

[44] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

[45] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.

[46] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[47] X. Zheng, Y. Wang, G. Wang, and J. Liu, "Fast and robust segmentation of white blood cell images by self-supervised learning," *Micron*, vol. 107, pp. 55–71, Apr. 2018.

SONGKUI CHEN received the B.S. degree from the School of Huizhou University, Huizhou, China, in 2017. He is currently pursuing the M.S. degree with the Department of Software Engineering, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include digital image processing and deep learning.

DAMING SHI (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from the Harbin Institute of Technology, China, and the Ph.D. degree in computer science from the University of Southampton, U.K. He worked at Middlesex University, U.K., Kyungpook National University, South Korea, and Nanyang Technological University, Singapore. He has been serving as a Distinguished Professor with Shenzhen University, since 2016. His current research interests include machine learning, image processing, pattern recognition, and neural networks.

MUHAMMAD SADIQ received the M.S. degree in computer science from Riphah International University, Pakistan, in 2015. He is currently pursuing the Ph.D. degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He has several publications in the last few years. His research interests are Artificial Intelligence, Cloud Computing, Cloud Security, and Computer Vision.

XIAOCHUN CHENG (Senior Member, IEEE) received the bachelor's degree in computer engineering and the Ph.D. degree in computer science from Jilin University, in 1992 and 1996, respectively. He has been Computer Science EU Project Coordinator in Middlesex University, since 2012. He contributed for peer reviewed 103 published journal articles, peer reviewed 134 conference papers, with five times best conference paper awards by end of 2018. Four articles are in the 2018 top 1% of the academic field based on a highly cited threshold for the field and publication year by Data from Essential Science Indicators. He is a member of the IEEE SMC Technical Committee on Enterprise Information Systems, the IEEE SMC Technical Committee on Computational Intelligence, the IEEE SMC Technical Committee on Cognitive Computing, the IEEE SMC Technical Committee on Intelligent Internet Systems, and BCS Artificial Intelligence Specialist Group.